

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод розпізнавання тексту на фотозображеннях
нейромережевими засобами

Галузь знань

12 – Інформаційні технології

Шифр і назва галузі знань

Спеціальність

122 – Комп'ютерні науки

Шифр і назва спеціальності

Освітня програма

Комп'ютерні науки

Назва освітньої програми

Виконала: студентка групи КН-20-1

Група виконавця




Підпис

Анастасія ШАПОВАЛОВА

Ім'я, ПРІЗВИЩЕ

Керівник: док.філ., ст. викл. каф. КН

Науковий ступінь, посада



Підпис

Павло РАДЮК

Ім'я, ПРІЗВИЩЕ

Нормоконтроль: к.т.н., доц. каф. КН

Науковий ступінь, посада



Підпис

Руслан БАГРІЙ

Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор



Підпис

Олександр БАРМАК

Ім'я, ПРІЗВИЩЕ

17 06 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук


(підпис)

д.т.н., професор Олександр БАРМАК

«16» 02 2024 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Метод розпізнавання тексту на фотозображеннях нейромережесивими засобами»

2. Завдання видано студентці Анастасії ШАПОВАЛОВІЙ

(ім'я, прізвище)

3. Керівник роботи старший викладач кафедри КН Павло РАДЮК

(посада, ім'я, прізвище)

5. Затверджено наказом університету від «15» 02 2024 р. № 8

5. Дата видачі завдання студенту: «16» 02 2024 р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – підвищення якості розпізнавання тексту на фотозображеннях нейромережесивими засобами. Досягнення мети роботи передбачало виконання таких завдань: проведено аналіз предметної області та обґрунтовано необхідність покращення процесу розпізнавання тексту на фотозображеннях товарних чеків; розроблено метод розпізнавання тексту на фотозображеннях нейромережесивими засобами; виконано програмну реалізацію методу розпізнавання тексту з інтуїтивно зрозумілим інтерфейсом; здійснено тестування розробленого методу та створеного на його основі застосунку; продемонстровано підвищення якості розпізнавання тексту на фотозображеннях через впровадження розробленого методу.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання роботи	січень 2024	виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2024	виконано
3	Робота над розділом 1 – Аналіз методів та підходів до розпізнавання тексту на фотозображеннях товарних чеків	лютий 2024	виконано
4	Робота над розділом 2 – Метод розпізнавання тексту на фотозображеннях нейромережевими засобами	березень 2024	виконано
4	Робота над розділом 3 – Програмна реалізація вебзастосунку для розпізнавання тексту на фотозображеннях товарних чеків	квітень 2024	виконано
5	Написання пояснювальної записки, урахування зауважень керівника, оформлення, відповідно до вимог	травень 2024	виконано
6	Розроблення презентаційних матеріалів та попередній захист кваліфікаційної роботи	травень 2024	виконано
8	Захист кваліфікаційної роботи бакалавра	червень 2024	виконано

Виконавець: студентка групи КН-20-1

Група виконавця



Підпис

Анастасія ШАПОВАЛОВА

Ім'я, ПРІЗВИЩЕ

Керівник:

док. філ., ст. викл. каф. КН

Науковий ступінь, посада



Підпис

Павло РАДЮК

Ім'я, ПРІЗВИЩЕ

Анотація

Тема кваліфікаційної роботи бакалавра: «Метод розпізнавання тексту на фотозображеннях нейромережевими засобами»

Виконавець кваліфікаційної роботи бакалавра: студентка групи КН-20-1 Анастасія Шаповалова

Керівник кваліфікаційної роботи бакалавра: доктор філософії, старший викладач кафедри КН Павло Радюк

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
71	29	14	35	4

Метою кваліфікаційної роботи бакалавра є підвищення якості розпізнавання тексту на фотозображеннях нейромережевими засобами. Мету роботи досягнуто у вигляді розробленого методу розпізнавання тексту на фотозображеннях нейромережевими засобами. Виконано програмну реалізацію методу розпізнавання у вигляді мобільного застосунку з використанням React Native та Firebase.

Розроблений застосунок призначений для спрощення процесу управління особистими фінансами користувачів. З його допомогою користувачі можуть швидко перетворювати фізичні товарні чеки покупок на цифровий формат, отримуючи деталізований список своїх витрат, включно з переліком продуктів, їхніми цінами та загальною сумою покупки. Список можливо редагувати, що дає змогу користувачам вести облік своїх витрат більш ефективно та організовано.

Ключові слова: управління фінансами, товарний чек, розпізнавання тексту, нейромережі, мобільний застосунок, React Native, Firebase.

Виконавець: студентка групи КН-20-1

Група виконавця



Підпис

Анастасія ШАПОВАЛОВА

Ім'я, ПРІЗВИЩЕ

Зміст

Перелік скорочень	4
Вступ.....	5
Розділ 1 Аналіз методів та підходів до розпізнавання тексту на фотозображеннях товарних чеків.....	7
1.1 Актуальність оброблення зображень паперових товарних чеків	7
1.2 Аналіз методів та підходів до розпізнавання тексту на зображеннях.....	9
1.3 Огляд наявних програмних рішень для оброблення товарних чеків	12
1.4 Мета та задачі кваліфікаційної роботи	16
Розділ 2 Метод розпізнавання тексту на фотозображеннях нейромережевими засобами	18
2.1 Схема методу розпізнавання тексту на фотозображеннях	18
2.2 Функціональна структура програмного забезпечення для реалізації методу розпізнавання тексту.....	23
2.3 Опис використаної моделі нейронної мережі	25
2.4 Проектна архітектура системи та взаємозв'язок компонентів.....	29
2.5 Проектування бази даних програмного забезпечення	34
2.6 Підготовка робочих вхідних даних для системи	40
2.7 Особливості використання спеціалізованих програмних компонентів	41
2.8 Висновки до розділу 2	43
Розділ 3 Програмна реалізація мобільного застосунку для розпізнавання тексту на фотозображеннях товарних чеків	44
3.1 Визначення шляхів дослідження та засобів створення програмного забезпечення	44
3.2 Вибір засобів розроблення програмного забезпечення	45
3.3 Структура та функціональне призначення програмних складових системи.....	48
3.4 Особливості реалізації програмних складових системи	53
3.5 Тестування програмної реалізації та вимоги до розгортання	59
3.6 Аналіз програмної реалізації розробленого методу	70
3.7 Результати досліджень	71
3.8 Висновки до розділу 3	74

Загальні висновки.....	75
Перелік посилань.....	76
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
БД	База даних
ПЗ	Програмне забезпечення
СКБД	Система керування базами даних
ШІ	Штучний інтелект
CNN	Convolutional Neural Networks
CRUD	Create Read Update Delete
ER	Entity-relationship
HTML	Hypertext Markup Language
ICDAR	International Conference on Document Analysis and Recognition
IDE	Integrated Development Environment
JS	JavaScript
JSX	JavaScript XML
LSTM	Long Short-Term Memory Networks
ML Kit	Machine Learning Kit
NoSQL	Not only Structured Query Language
npm	Node Package Manager
OCR	Optical Character Recognition
RNN	Recurrent Neural Networks
UI	User Interface
VS Code	Visual Studio Code

Вступ

Кваліфікаційна робота бакалавра присвячена підвищенню якості розпізнавання тексту на фотозображеннях через розроблення методу розпізнавання тексту на фотозображеннях нейромережевими засобами та створення відповідної програмної реалізації у вигляді мобільного застосунку, що дає змогу на фотозображеннях товарних чеків розпізнавати текст та подавати його у зручному для користувача форматі.

Актуальність. На сьогодні галузь роздрібної торгівлі спонукає продавців генерувати масу товарних чеків. Оброблення та аналіз товарних чеків є важливим для кінцевих користувачів, оскільки чеки часто містять багато цінної інформації, яку, проте, важко вилучати та обробляти вручну. Цей ручний процес займає багато часу та може призвести до помилок, що призводить до втрати даних і збільшення адміністративного навантаження. Для вирішення цієї проблеми важливо розробити метод розпізнавання тексту на фотозображеннях із використанням нейромережових технологій та розробити відповідний мобільний застосунок. Це рішення дасть можливість покращити процес розпізнавання, підвищивши її точність, що надалі спростить оброблення та використання товарних чеків.

Об'єкт дослідження – процес розпізнавання тексту на фотозображеннях.

Предмет дослідження – нейромережеві методи розпізнавання тексту на фотозображеннях.

Мета кваліфікаційної роботи бакалавра – підвищення якості розпізнавання тексту на фотозображеннях нейромережевими засобами.

Завдання кваліфікаційної роботи бакалавра – Провести аналіз предметної області та обґрунтувати необхідність покращення процесу розпізнавання тексту на фотозображеннях товарних чеків за допомогою нейромереж. Розробити метод розпізнавання тексту на фотозображеннях нейромережевими засобами. Спроекувати архітектуру застосунку на основі розробленого методу розпізнавання тексту на фотозображеннях. Розробити

модуль забезпечення зберігання та редагування сканованого тексту. Виконати програмну реалізацію застосунку за розробленим методом з інтуїтивно зрозумілим інтерфейсом. Провести тестування розробленого методу розпізнавання тексту на фотозображеннях та створеного на його основі застосунку. Дослідити якість виконання методу OCR, а саме, провести низку тестів для оцінювання точності та швидкості розпізнавання тексту. Продемонструвати підвищення якості розпізнавання тексту на фотозображеннях через впровадження розробленого методу розпізнавання тексту.

Розділ 1 Аналіз методів та підходів до розпізнавання тексту на фотозображеннях товарних чеків

1.1 Актуальність оброблення зображень паперових товарних чеків

В сучасному світі ми постійно здійснюємо покупки. Ми купуємо продукти в супермаркетах, замовляємо одяг та здійснюємо імпульсивні покупки під впливом реклами. Вміння слідкувати за своїми витратами стає дуже важливим. Раніше для цього люди повинні були вести паперові записи та зберігати паперові чеки, що забирало багато часу і займало місце.

З появою інтернет-шопінгу стежити за грошима стало простіше, адже люди можуть сплачувати онлайн та отримувати чеки як на сайті магазину, так і в застосунку банку або на вказану електронну пошту. Проте, інтернет-покупки є лише частиною наших витрат, і звичайні магазини досі мають свою актуальність [1]. Під час фізичних покупок накопичується багато товарних чеків, які засвідчують покупки, але зазвичай ці чеки викидаються в смітник як щось непотрібне та зайве. Проте інформація, що в них міститься, могла б допомогти слідкувати за витратами для ефективного використання бюджету [2].

Ось чому важливо автоматизувати цей процес. Адже автоматизувавши його, ми зможемо не тільки тримати всю інформацію в електронному вигляді, але й переглядати, скільки грошей витратили, та планувати бюджет на основі цієї інформації. Так з'являються ідеї, такі як сканування товарних чеків із подальшим витягуванням інформації для користувача [3]. Подібні ідеї є рішенням для, здавалося б, незначної проблеми, але якщо задуматися, такий спосіб відстеження коштів зробив би нас більш фінансово грамотними.

В еру технологічних досягнень користувачі мають доступ до багатьох рішень автоматизації процесу людської роботи. Застосування нейромереж є популярним засобом у цій сфері. Зараз розпізнавання фотозображень є як ніколи актуальним, адже цьому сприяє поява штучного інтелекту (ШІ) та непокрита автоматизацією сфери життя [4].

Поставлена задача має низку переваг, що підтверджують її актуальність та ефективність у порівнянні з традиційними методами:

а) швидкість – швидкість роботи ШІ значно випереджає стандартні методи ручного розпізнавання тексту, що також вказує на його ефективність;

б) автоматизація – нейромережі можуть працювати автоматично і безперервно, завдяки чому можна обробляти великі обсяги фотозображень без участі людини;

в) точність – нейромережі вміють самовдосконалюватися та тренуватися на своїх результатах роботи та помилках;

г) масштабованість – системи ШІ дають змогу масштабувати результати для використання в різноманітних сферах, починаючи від медицини та закінчуючи власними потребами.

Ще однією причиною необхідності автоматизації є фактор людської помилки, що невпинно впливає на рішення та життя людей у різних сферах. Також із розвитком технологій росте попит на все, що не потрібно робити руками, а можна отримати іншими методами. Більше того, кількість цифрової інформації постійно збільшується, і нам так чи інакше потрібні засоби роботи з нею [5]. Часто з'являються новини про інновації у сфері ШІ, що засвідчує невпинний розвиток та розширення даної технології [6].

Розглянемо варіант використання автоматизації зчитування тексту з фотозображень у суспільному житті. В умовах нашого сучасного життя є багато людей, які перебувають в Україні або за кордоном і регулярно здійснюють покупки. Найбільшими проблемами, з якими вони стикаються, є такі:

– орієнтація в асортименті товарів у чужих країнах, де вся інформація подається чужою мовою;

– люди, що мають вади зору, мають труднощі з читанням інформації на невеликому чеку;

– деякі люди не хочуть витратити значну частину свого часу на процес відстежування фінансів, адже без автоматизації це займає значно більше часу.

Використання нейромереж допоможе додати автоматизацію та зменшити проблеми, з якими стикаються люди. Використовуючи сучасні методи роботи зі ШІ та нейромережами, ми зможемо автоматизувати зчитування інформації з товарних чеків та відображати список цін і куплених продуктів в електронному форматі. Цей підхід значно полегшить організацію фінансів та дасть змогу зібрати статистику покупок за різні періоди часу.

Такі підходи до розпізнавання тексту за допомогою нейромереж дадуть можливість підняти ефективність та точність процесу, зменшуючи витрати людського часу та ресурсів. Основна ідея полягає у використанні новітніх технологій для автоматизації та вдосконалення завдань, які раніше виконувалися переважно вручну.

1.2 Аналіз методів та підходів до розпізнавання тексту на зображеннях

На сьогодні значний обсяг інформації передається у вигляді картинок, важливо мати спосіб розпізнавати текст на фотографіях. Це особливо актуально для тих, хто робить покупки та слідкує за своїми фінансами.

Огляд наявних теоретичних підходів до розв'язання цієї задачі дасть змогу визначити ключові методи та техніки, що вже використовуються у даній області.

Розробники, що почали активно використовувати нейромережі [7], пропонують наступні підходи, що коротко описані нижче.

1. Використання CNN (Convolutional Neural Networks) – згорткові нейронні мережі, що вирізняються продуктивністю роботи з зображеннями. Даний тип архітектури нейронної мережі допомагає виявляти особливості та закономірності зображень, стійкий до масштабування, може обробляти великі обсяги за раз [8].

2. Attention Mechanism [9] – тип архітектури нейронних мереж, що дає змогу зосереджуватися на конкретних розділах вхідних даних під час виконання

завдання. Повторювані моделі використовують тип навчання з підкріпленням, суть якого – зосередити увагу на ключових елементах зображення.

3. RNN (Recurrent Neural Networks) та Long Short-Term Memory Networks – Recurrent Neural Networks [10] та їх розширення, такі як LSTM(Long Short-Term Memory Networks), використовуються для роботи з послідовними даними. Вони враховують зв'язок між текстом і символами, що значно покращує роботу з текстовими зображеннями.

4. Deep Reinforcement Learning [11] – метод, оптимізація процесу розпізнавання з використанням принципів посиленого навчання. Таким чином, модель може самостійно визначати найбільш ефективні стратегії при знаходженні тексту на зображенні.

5. Transformers [12] – трансформери, що використовують механізми самоуваги для оброблення послідовностей даних. Результат використання даних механізмів у розв'язанні задач обробки природної мови мав значний результат.

6. OCR (Optical Character Recognition) [13], або оптичне розпізнавання символів, – це процес трансформування зображень тексту, наприклад, сканованих фотозображень, у машинно-читані дані. Сучасні технології оптичного розпізнавання символів широко використовують ШІ і нейромережі для покращення точності та ефективності процесу.

Кожен з описаних вище методів відіграє важливу роль у розвитку галузі. Кожен з підходів вирізняється певною особливістю, що надає методу перевагу перед іншими при використанні у певного роду завданнях.

RNN та LSTM є найбільш прийнятними методами аналізу даних, зокрема тексту, тому що вони здатні зберігати інформацію про попередні елементи в послідовності. В контексті розпізнавання тексту із зображень це дає змогу ідентифікувати зв'язки між словами та символами, навіть у випадках з нерегулярним розміщенням тексту.

Глибоке посилене навчання є корисним у перспективі адаптації до різних форматів тексту, не включаючи методи ручного налаштування. Трансформери мають значний потенціал у сфері розпізнавання тексту на зображеннях, де вони

можуть бути корисними для ідентифікації та інтерпретації текстових структур. Для кращого розуміння розглянемо порівняльну характеристику в таблиці 1.1.

Таблиця 1.1 – Порівняння нейромережових підходів до задачі розпізнавання тексту на зображеннях

Назва підходу	Переваги	Недоліки	Технічні деталі
CNN	Точність в розпізнаванні образів; стійкість до масштабування.	Може потребувати значних обчислювальних ресурсів.	Використання глибоких нейронних мереж з багатьма шарами.
Attention Mechanism	Підвищує точність моделі.	Складність імплементації; збільшення часу навчання.	Інтеграція з іншими моделями нейромереж для підвищення ефективності.
RNN та LSTM	Ефективність при роботі з послідовними даними; здатність враховувати контекст.	Висока складність моделей; проблеми з тренуванням при довгих послідовностях.	Використання спеціалізованих архітектур для оброблення послідовних даних.
Глибоке посилене навчання	Адаптивність; самовдосконалення.	Складності у реалізації.	Використання нагород для оптимізації.
Transformers	Висока точність ідентифікації складних структур.	Великі вимоги до обчислювальних потужностей.	Механізми самоуваги; обробка послідовностей.
OCR (Optical	Висока швидкість	Складності в	Є комбінацією

Character Recognition)	і точність розпізнавання тексту; обробка різних шрифтів та різних мов.	розпізнаванні рукописного тексту.	алгоритмів штучного інтелекту, разом з традиційними алгоритмами.
------------------------	------------------------------------------------------------------------	-----------------------------------	------------------------------------------------------------------

Попри стрімкий прогрес у розробленні методів розпізнавання тексту, подібні технології все ще потребують певних вдосконалень. Значні покращення, що очікуються від подібних підходів:

- покращення точності розпізнавання;
- швидка обробка великих обсягів даних;
- розпізнавання зображень підвищеної складності;
- збільшення швидкості обробки даних.

Наведені підходи активно використовуються для вирішення певних задач. З огляду на характер та особливості завдань, у роботі вибрано комбінацію підходів CNN та OCR.

1.3 Огляд наявних програмних рішень для оброблення товарних чеків

Розглянемо різноманіття програм, що доступні для використання вже сьогодні. Ці програми спеціалізуються на витягуванні тексту із зображень, використовуючи можливості штучного інтелекту, зокрема механізми нейромереж. Завдяки сучасним технологічним досягненням існує можливість ефективно сканувати візуальний контент із зображень для подальшого перетворення у текстовий формат для оброблення в межах певної задачі.

З метою оцінки поточного спектра вже створених програмних рішень розглянемо сучасні застосунки, які націлені на вирішення подібних задач, їхні характеристики та методи. Використання ШІ для витягування даних із

зображення є досить поширеною практикою. Далі наведено декілька популярних прикладів.

CamScanner – це застосунок, який дає змогу використовувати смартфон у якості сканера документів [14] (рисунок 1.1). З його допомогою можна легко сканувати будь-які документи, статті або записи. Застосунок допомагає зберегти якісну копію паперового варіанту зображення.

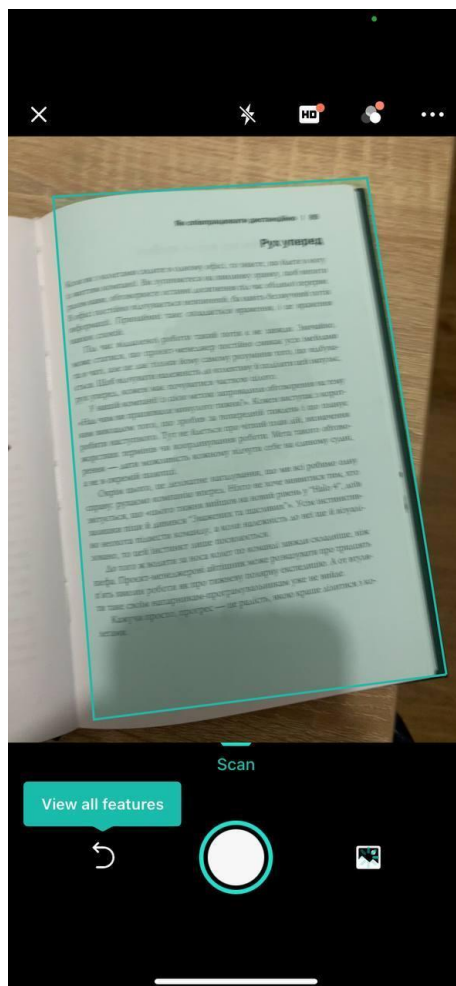


Рисунок 1.1 – Сканування документа з CamScanner [14]

Перевагою CamScanner є застосування технології оптичного розпізнавання символів. Це чи не найважливіша функція застосунку, яка завдячує своїй ефективності ШІ. Завдяки розпізнаванню символів користувачі мають змогу зберегти документ як текст, доступний для редагування та копіювання [15].

CamScanner є комплексним рішенням цифрової обробки даних, але попри свою популярність має недоліки, як от, реклама, конфіденційність, якість тощо.

Інше відоме рішення, що застосовується для роботи з паперовим текстом, – Adobe Scan [16]. Цей застосунок надає доступ до функціоналу для сканування документів, квитанцій, візиток і навіть рукописних нотаток. З його допомогою можна сканувати паперові документи і зберігати їх у форматі PDF-файлів (рисунок 1.2).

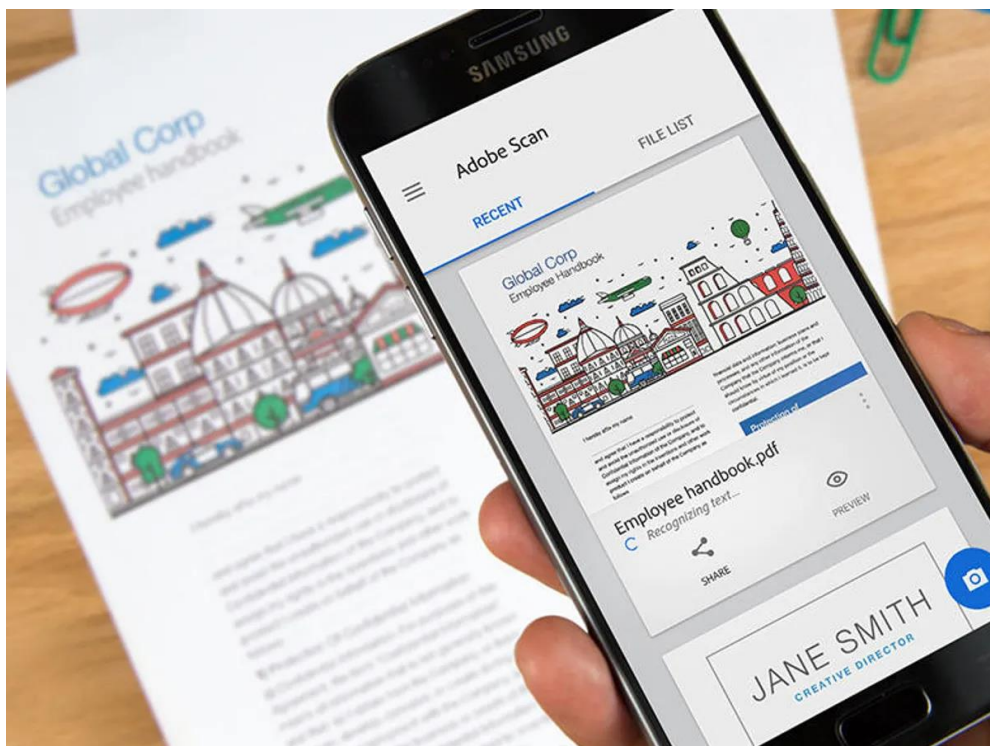


Рисунок 1.2 – Сканування з допомогою Adobe Scan [17]

Даний застосунок використовує ШІ для ключового функціонала, такого як виявлення документів, покращення якості картинки, розпізнавання символів. До його недоліків можна віднести наступне:

- залежність від облікового запису Adobe;
- обмеженість – без підписки користувач отримує дуже обмежений доступ до функціоналу;
- інтерфейс – деякі користувачі виділяють інтерфейс Adobe Scan як не цілком інтуїтивно зрозумілий [18].

Microsoft Office Lens – застосунок, що дає змогу сканувати документи, візитки, нотатки та інші матеріали. Результат сканування може бути збережений в різних форматах, наприклад, у PDF або JPEG (рисунок 1.3). Використання методів штучного інтелекту в Office Lens дає змогу автоматично здійснювати комплексні завдання оброблення зображень, що забезпечує високу якість сканування та зручність використання для користувачів [19].

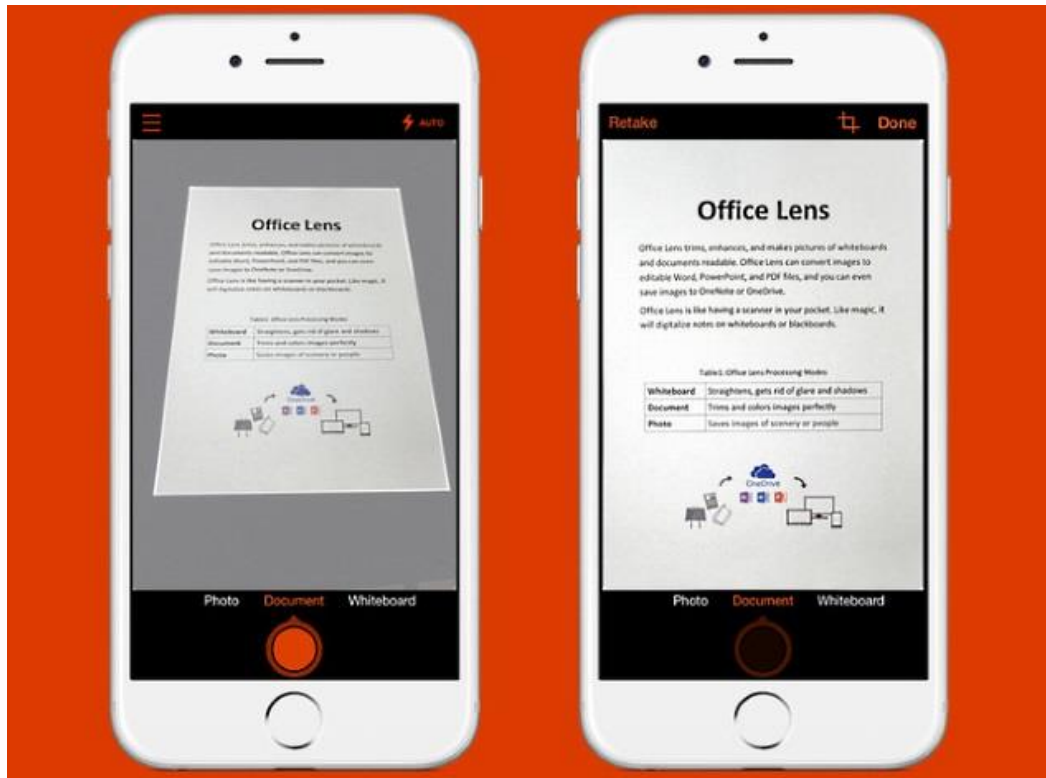


Рисунок 1.3 – Microsoft Office Lens [19]

Як і будь-який інший застосунок, Microsoft Office Lens має свої недоліки:

- залежність якості від камери користувача;
- обмеженість форматів файлів;
- недостатня точність;
- проблеми зі стабільністю.

Дослідження існуючих програмних рішень для розпізнавання тексту з фотозображень вказує на широкі перспективи для розуміння поточного стану використання ШІ для розроблення та застосування даної технології. Програмні рішення, продемонстровані вище, засвідчують ефективність сучасних алгоритмів

штучного інтелекту та доцільність їх використання. Отже, розглянута задача розпізнавання тексту з фотозображень є актуальною, а використання нейронних мереж є необхідним етапом розв'язання даної задачі, що сприятиме подальшому розвитку цифрової обробки даних і доступності інформації.

Насамкінець, у сучасному світі швидка та точна обробка інформації має велике значення, і автоматизація розпізнавання тексту з фотографій відіграє важливу роль у цьому процесі. Використання методів оптичного розпізнавання символів (OCR), що базуються на нейронних мережах, значно допомагає збільшити точність і швидкість виявлення тексту із зображень. У рамках виконання роботи було використано методи OCR [20] із застосуванням можливостей ML Kit (Machine Learning Kit) від Firebase [21] та Vision Framework від Apple [22].

OCR, або оптичне розпізнавання символів (Optical Character Recognition), – це нейромережева технологія, що дає змогу конвертувати різні типи документів, наприклад, такі як зображення друкованого тексту, рукописний текст чи файли, у вихідні редаговані дані без потреби в ручному введенні. Vision Framework від Apple надає потужні інструменти для реалізації OCR на iOS-пристроях та забезпечує високу точність і швидкість роботи.

1.4 Мета та задачі кваліфікаційної роботи

Проведено аналіз та обґрунтовано необхідність покращення процесу розпізнавання тексту на фотозображеннях товарних чеків за допомогою нейромереж. У результаті проведеного аналізу визначено проблеми користувачів, що виконують ручний запис даних, та запропоновано спосіб покращення цієї проблеми. Основна увага приділяється підходам, що призведуть до швидкості оброблення даних, уникнення чинників людської помилки та більшої доступності інформації для подальшого аналізу. Проведений аналіз також підкреслив важливість використання ШІ для автоматизації оброблення візуальних даних і вказав на поточні тенденції в цій області.

Отже, в результаті проведеного аналізу предметної області формалізовано мету кваліфікаційної роботи бакалавра як підвищення якості розпізнавання тексту на фотозображеннях нейромережевими засобами. Досягнення мети роботи передбачає розроблення методу розпізнавання тексту на фотозображеннях нейромережевими засобами та реалізацію на його основі застосунку, що дасть змогу користувачам переводити текст із фото товарних чеків у редагований електронний формат. Створений застосунок мусить перетворювати фото тексту на цифровий текст, який можна редагувати, що дасть можливість користувачам швидше працювати з текстовою інформацією, не витрачаючи час на її ручне переписування.

Досягнення мети кваліфікаційної роботи передбачає розв'язання таких завдань.

1. Розробити метод розпізнавання тексту на фотозображеннях нейромережевими засобами.
2. Спроекувати архітектуру застосунку на основі розробленого методу розпізнавання тексту на фотозображеннях.
3. Розробити модуль забезпечення зберігання та редагування тексту, що розпізнаний на зображенні товарного чеку.
4. Виконати програмну реалізацію застосунку за розробленим методом з інтуїтивно зрозумілим інтерфейсом.
5. Провести тестування розробленого методу розпізнавання тексту на фотозображеннях та створеного на його основі застосунку.
6. Продемонструвати підвищення якості розпізнавання тексту на фотозображеннях через впровадження розробленого методу розпізнавання.

Розділ 2 Метод розпізнавання тексту на фотозображеннях неймережевими засобами

2.1 Схема методу розпізнавання тексту на фотозображеннях

Розглянемо схему відображення методу на рисунку 2.1.

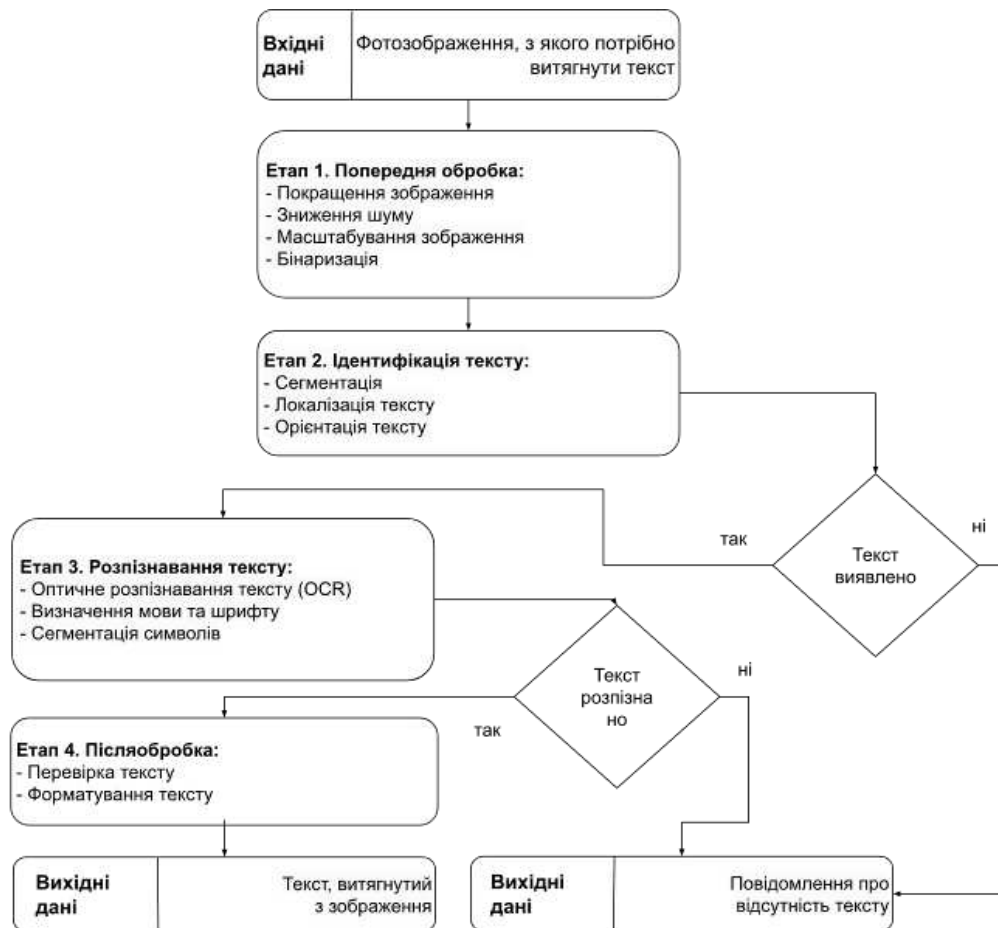


Рисунок 2.1 – Схема методу розпізнавання тексту на фотозображеннях

Вхідні дані отримано з фотозображення з камери або завантаження з галереї, після чого виконуються етапи розпізнавання.

Етап 1. Попереднє оброблення:

- покращення зображення за рахунок зміни яскравості, різкості та контрастності;

- зниження шуму;

- масштабування зображення: зміна розміру завантаженого зображення за потреби;

- бінаризація: перетворення зображення у двоколіоровий формат для покращення процесу розпізнавання.

Етап 2. Ідентифікація тексту за допомогою нейромереж:

- сегментація: розбиття фотозображення на текстові сегменти;

- локалізація тексту: визначення потенційних зон тексту;

- текст виявлено: перехід до наступного етапу розпізнавання тексту;

- текст не виявлено: повідомлення про відсутність тексту та завершення процесу.

Етап 3. Розпізнавання тексту:

- оптичне розпізнавання тексту (OCR): розпізнавання тексту у виявлених текстових зонах;

- помилка розпізнавання: повідомлення про відсутність тексту та завершення процесу;

- текст виявлено: перехід до наступного етапу післяоброблення.

Етап 4. Післяоброблення:

- перевірка тексту: корекція розпізнаного тексту з допомогою мовних моделей;

- форматування тексту: форматування тексту читабельними реченнями або словосполученнями.

На етапі форматування тексту відбувається значуща частина обробки даних. Після того, як нейромережева модель повертає дані, вони повинні бути оброблені. Вихідні дані являють собою масив тексту, що було розпізнано на фотографії (рисунок 2.2). Ціль форматування наступна – знайти в тексті частини що відповідають за назву та ціну продуктів, та скласти їх у відповідний масив об'єктів (рисунок 2.3), який можна буде використати для автоматичного перенесення в форму витрат.

```

> inputData
< (51) ['BULEVARDUL MUNGILY', 'JUD. CLUJ, CLUJ-NAPOCA', 'C.I.F.: R017233051', 'BON FISCAL', 'ZILNIC 0
7.00-22.00', 'MOZZARELLA DE VACA C', '1 BUC x 12,00', 'CIUPERCI CHAMPIGNON 1 BUC X 5,99', 'BAGHETA 3
00G**', '# BUC X 2,40', 'STRUGURI ALBI 500G S', '1 BUC X 7,89', 'NECTARINE CASEROLA 5', '1 BUC X 8,9
9', 'TAURT CU STRACIATELE', '1 BUC X 10 20', 'ROSII PRUNISOARA RO', '1 BUC X 11,99', 'BROCCOLI KG',
'0,542 KG X 14,49', 'PUNGA BIO FLEG 34 X 1 BUC X 0,10', 'CASTRAVETI CORNICHON', '1 BUC X 6,99', 'LAP
TE SEMIDEGRESAT 1 BUC X 12,20', 'PAINE DIN GRAU DUR 4', '1 BUC X 6,70', 'PARIZER CU CARNE DEA', 'FIL
E HAIDUCESC FOX', '0,432 KG X 31,90', 'ALMETTE SMANTANA 150', '0,496 KG X 68,90', 'BANANE KG', '1 BU
C X 6,85', 'AVOCADO PLASA 500€', '1,228 KG X 6,99', '12,00 B', '5,99 B', '2,40 B', '7,89 B', '8,99
B', '10,20 D', '11,99 B', '7,85 B', '0,10 D', '6,99 B', '12,20 B', '6,70 B', '13,78 B', '34,17 B',
'6.85 B', 'ManDank Air']

```

Рисунок 2.2 – Масив розпізнаного тексту до парсингу

```

> outputData
< (11) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}] ⓘ
  ▶ 0: {name: 'STRUGURI ALBI 500G S', price: '7.89'}
  ▶ 1: {name: 'NECTARINE CASEROLA 5', price: '8.99'}
  ▶ 2: {name: 'ROSII PRUNISOARA RO', price: '11.99'}
  ▶ 3: {name: 'BROCCOLI KG', price: '7.85'}
  ▶ 4: {name: 'CASTRAVETI CORNICHON', price: '6.99'}
  ▶ 5: {name: 'PAINE DIN GRAU DUR 4', price: '6.70'}
  ▶ 6: {name: 'FILE HAIDUCESC FOX', price: '13.78'}
  ▶ 7: {name: 'ALMETTE SMANTANA 150', price: '34.17'}
  ▶ 8: {name: 'BANANE KG', price: '6.85'}
  ▶ 9: {name: 'AVOCADO PLASA 500€', price: '8.58'}
  ▶ 10: {name: 'ManDank Air', price: 5.99}

```

Рисунок 2.3 – Масив розпізнаного тексту після парсингу

Спочатку здійснюється перевірка чи вхідний масив є дійсно масивом та чи містить він потрібні дані. Якщо масив відповідає очікуваному формату, відбувається визначення рядків, що містять назви продуктів та їх ціни. Для визначення використовуються регулярні вирази: один для розпізнавання назв, що дає змогу ідентифікувати рядки з текстом, що містить саме назви товарів, та інший, що виявляє рядки, що містять цінові значення. Застосування даних виразів допомагають виділити необхідну інформацію з загального масиву тексту, відкинувши зайву інформацію.

Алгоритм проходиться по кожному рядку вхідного масиву, перевіряючи чи він відповідає регулярному виразу для назви продуктів. Якщо відповідність існує – перевіряється чи наступний рядок містить ціну. У разі відповідності назва та ціна продукту виділяються та додаються до масиву результатів, у вигляді об'єкта з ключами `name` та `price`. Після чого алгоритм переходить до обробки наступних рядків.

Даний процес дає змогу перетворити неструктурований масив рядків на структурований масив об'єктів, що містить тільки потрібну користувачу

інформацію. Це значно спрощує подальшу роботу з даними та забезпечує зручність переносу списку з фото зображення в форму створення витрати. Адже після того як користувач підтверджує введення даних, форма автоматично заповнюється наявними даними (рисунок 2.4).



Рисунок 2.4 – Заповнена форма розпізнаним текстом з фото зображення

Для визначення оцінки ефективності методу розпізнавання тексту з фотозображень використовується показник точності. Показник дає змогу зрозуміти, наскільки точно система може ідентифікувати та витягнути текст з різних зображень.

Нижче подамо формулу обрахування точності:

$$A = \frac{T_c}{T_i} \cdot 100\%, \quad (2.1)$$

де A – показує відсоток тексту на зображенні, що був ідентифікований без помилок; T_c – кількість символів, які система успішно ідентифікувала на

зображенні; T_i – загальна кількість символів, які є на зображенні і які потрібно розпізнати.

Базова діаграма активностей (рисунок 2.5) відображає основні етапи методу та послідовність кроків у процесі розпізнавання інформації.

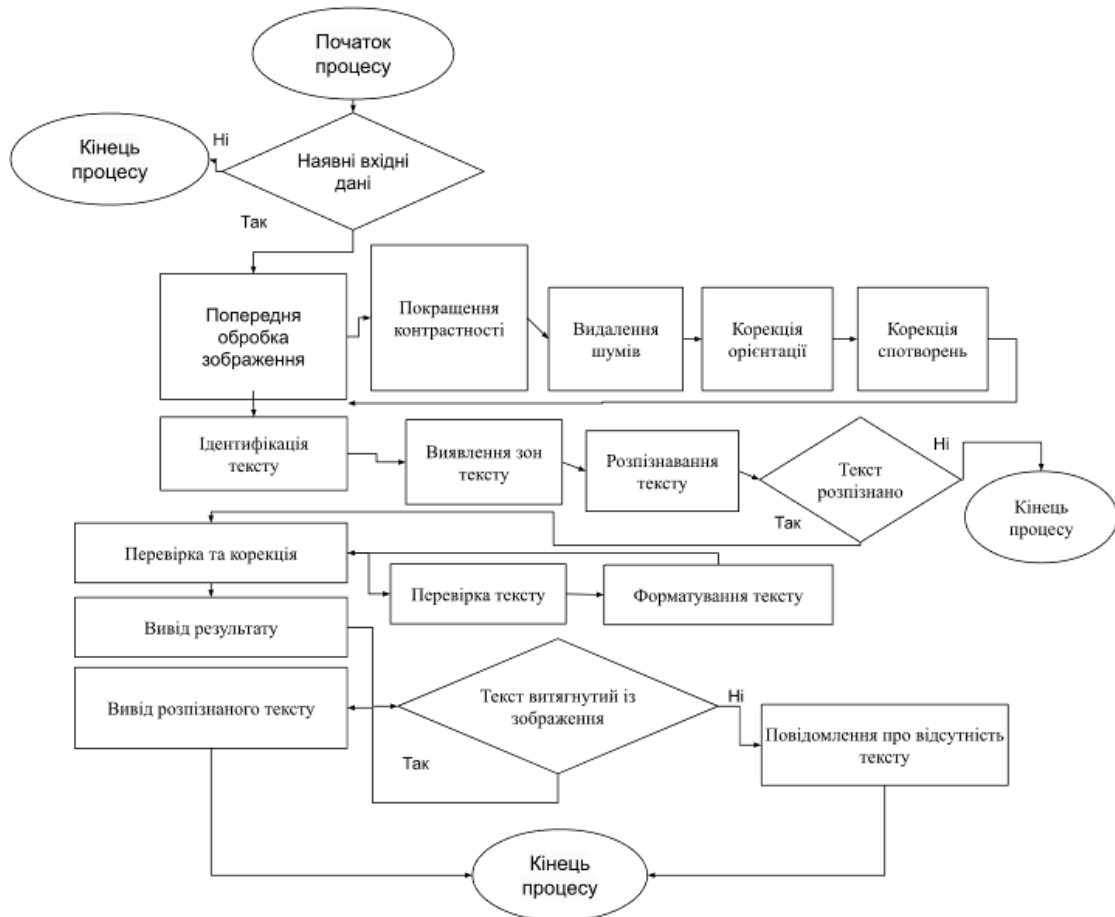


Рисунок 2.5 – Діаграма активностей процесу розпізнавання тексту запропонованого методу

1. Початок процесу.
2. Перевірка вхідних даних:
 - відсутність вхідних даних – кінець процесу;
 - наявні вхідні дані – перехід до наступного етапу.
3. Попереднє оброблення зображення:
 - покращення контрастності;
 - виявлення шумів;
 - корекція орієнтації;

- корекція спотворень.

4. Ідентифікація тексту:

- виявлення зон з текстом;
- текст не розпізнано – кінець процесу;
- текст розпізнано – перехід до наступного етапу.

5. Перевірка та кореляція:

- перевірка тексту;
- форматування тексту.

6. Виведення результату:

- текст не виявлено – повідомлення про відсутність тексту на зображенні;
- текст виявлено – виведення виявленого тексту.

7. Кінець процесу.

Vision Framework від Apple забезпечує потужні інструменти для оброблення та розпізнавання тексту з фотозображень при використанні сучасних методів машинного навчання.

Описаний метод розпізнавання тексту використовує технології машинного навчання для оброблення фотозображень, розпізнавання та витягування тексту з них. Це дає змогу автоматизувати та у такий спосіб процес розпізнавання даних та забезпечити ефективність і точність розпізнавання.

2.2 Функціональна структура програмного забезпечення для реалізації методу розпізнавання тексту

Функціональна структура програмної реалізації призначена для автоматизації обробки фінансових даних користувачів. Основні можливості системи включають авторизацію, управління підписками та витратами користувача, збирання статистики, а також використання нейромереж для аналізу фотозображень для зчитування тексту. Далі описано ключові функціональні блоки та їх взаємодії, що сприяють плануванню та виконанню різних завдань системи.

Основні функції системи подано нижче.

а) Авторизація користувача – кожен користувач може без проблем створити акаунт або залогінитися в уже існуючий за допомогою Google авторизації.

б) Управління профілем користувача – після авторизації доступно переглянути інформацію користувача та змінити її. Інформація, доступна користувачеві: firstName, lastName, avatarUrl. Email неможливо відредагувати, оскільки він використовується для авторизації, проте користувач завжди може побачити його в профілі.

в) Управління підписками – користувач може додавати та видаляти інформацію про свої підписки. Для цього користувач обирає відповідну іконку, вказує назву сервісу, на який оформлено підписку, вказує ціну, валюту та період, за який було сплачено.

г) Управління витратами – тут користувач може вносити інформацію про свої витрати, додавши відповідні дані. Ключовими є дані, які відображаються як список покупок з ціною. Для заповнення цих даних користувач може як сам ввести дані в поля, так і завантажити фото чека, після чого засобами нейромережі буде зчитано та відображено інформацію про назву та ціну продуктів.

г) Статистика – усі дії користувача зберігаються задля відображення статистики: графік та відображення усіх витрат за обраний місяць.

Навігаційні схеми відіграють ключову роль у визначенні навігації в системі. Вони візуалізують можливі шляхи для пересування між різними частинами застосунку та вказують на зв'язки між різними екранами та модулями.

Переходи між екранами відбуваються через меню навігації та є інтуїтивно зрозумілими для користувача. Основні компоненти навігаційної системи відображені на рисунку 2.6.

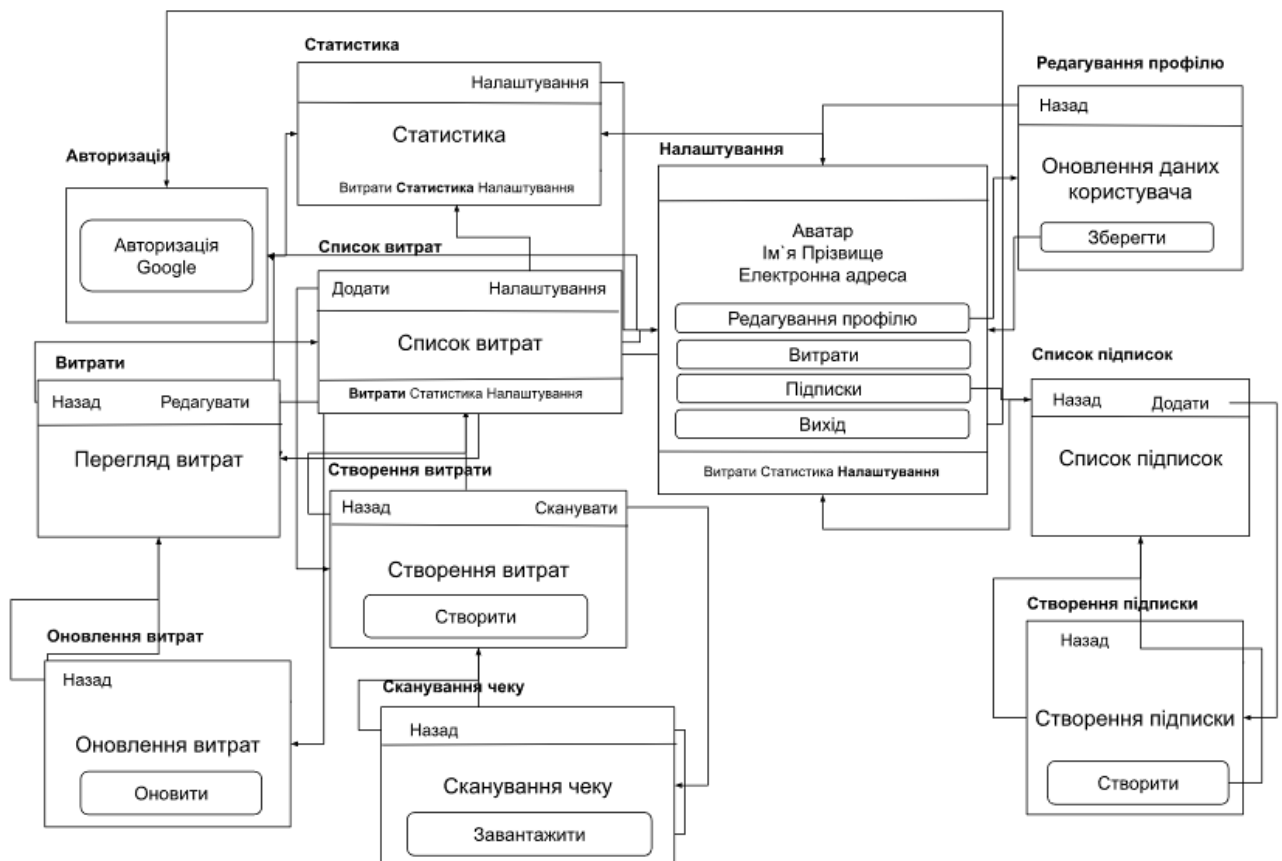


Рисунок 2.6 – Компоненти навігаційної системи

Відповідно до навігаційної схеми, ми можемо бачити, що після авторизації користувач потрапляє на головну сторінку програми.

2.3 Опис використаної моделі нейронної мережі

Попри те, що Vision Framework не надає вичерпної інформації про внутрішню архітектуру використовуваних мереж, розглянемо деталі та пов'язану інформацію, що так чи інакше була представлена Apple.

Наразі Vision Framework використовує два напрями для розпізнавання тексту на зображеннях:

- Fast Path використовує модель машинного навчання, щоб виявити окремі слова та символи. Даний підхід наслідує традиційне використання OCR;
- Accurate Path розпізнає текст за допомогою складнішої моделі, що може враховувати контекст [23].

На базі документації Vision Framework можемо зробити наступні висновки на основі загальних принципів, використаних у продуктах Apple [24].

Загальні параметри та гіперпараметри вбудованої в Vision Framework нейромережі наведено нижче.

1. Розмір вхідного зображення: зображення масштабується для забезпечення стабільної обробки.

2. Використання шарів для оброблення даних: структура включає наступні типи шарів:

– CNN: модель має 10 подібних шарів, що використовуються для виділення ознак; розмір ядра – 3×3 та 5×5 ;

– Pooling Layers використовується задля зменшення розміру просторових ознак, ядро – 2×2 ;

– Рекурентні шари – для оброблення послідовностей;

– Повнозв'язні шари: – 2 шари даного типу.

3. Функції активації використовуються для прихованих шарів та класифікації символів.

4. Функції втрат вирішують проблеми невідповідності вхідних та вихідних даних.

5. Оптимізатор використовується для пришвидшення та покращення навчання моделі.

6. Кількість епох навчання – 100 епох.

7. Пакетний розмір – 128 зразків.

Vision Framework використовує низку різних наборів даних для навчання. Навчання відбувається у різних умовах та на різних мовах. Основні типи даних для навчання, що запропоновані компанією Apple, є такими:

а) ICDAR (International Conference on Document Analysis and Recognition) містить різноманітні види текстової інформації та є найчастіше використовуваним набором даних при тренуванні OCR;

б) SynthText використовується для тренування моделей та пропонує широкий спектр зображень тексту в різних умовах;

в) Google's proprietary dataset набір даних, представлений Google, що пропонує набір текстів, документів, зображень тощо;

г) COCO-Text набір даних, що базуються на реальних джерелах.

Для навчання моделей використовуються набори даних з анотаціями, які точно вказують де знаходяться текстові блоки на зображеннях. Це допомагає моделям вчитися розпізнавати текст з високою точністю. Завдяки різноманіттю наборів даних моделі стають здатними розпізнавати текст у різних умовах. Аугментація даних, яка включає зміни освітлення, повороти, масштабування та інші трансформації, покращує стійкість моделей до різних умов зйомки.

Одним з найважливіших етапів при розпізнаванні тексту, є попереднє оброблення зображення. Цей етап має на мені підвищити точність та якість результатів. У Vision Framework для попереднього оброблення використовуються такі етапи (рисунок 2.7):

а) підвищення контрастності: алгоритми для збільшення різниці між темними та світлими областями зображення;

б) видалення шуму: зменшення небажаних елементів, що заважають розпізнаванню зображення;

в) корекція орієнтації: виправлення можливих перекосів та нахилу вхідного зображення;

г) усунення перекосів: корекція візуальних спотворень, що заважають розпізнаванню зображення;

д) виділення регіону інтересу: акцентування на частині фотозображення з текстом, щоб покращити розпізнавання.

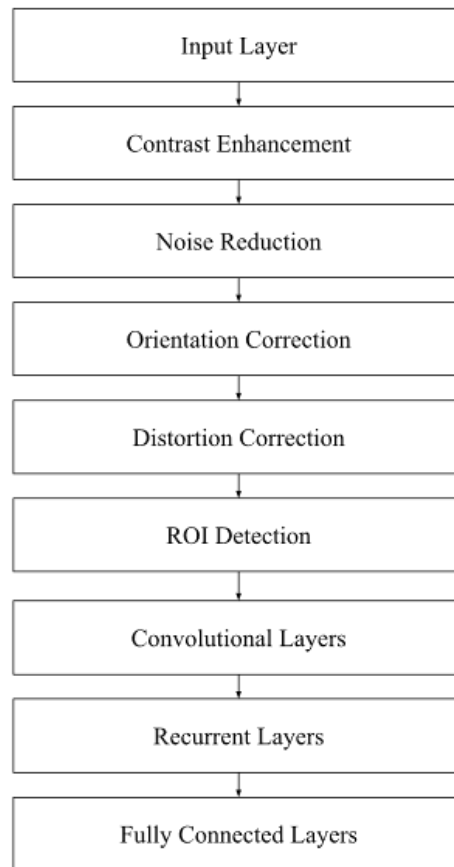


Рисунок 2.7 – Архітектура Vision Framework

Послідовність дій оброблення є такою наступна:

- 1) зображення завантажується в систему через завантаження фото з галереї або камери;
- 2) відбувається покращення контрастності між фоном і текстом, для спрощення розпізнавання;
- 3) застосовуються фільтри, що націлені на видалення шуму задля очищення зображення від небажаних деталей;
- 4) виправляються перекося та нахил зображення;
- 5) зображення піддається обробці для корекції спотворень та викривлень;
- 6) розпізнаються зони зображень, які містять текст;
- 7) за допомогою використання CNN виділяються ознаки тексту на фотозображенні;
- 8) за допомогою використання RNN аналізуються послідовності виділеного тексту;

9) застосовуються повнозв'язані шари для фінального розпізнавання тексту.

Наведені етапи дозволяють покращити якість вхідних даних, що забезпечує більш точне та надійне розпізнавання тексту.

Вище було детально розглянуто загальні принципи та деталі архітектури що використовується в Vision Framework. Було описано основні методи що використовуються для попереднього оброблення зображень. Кожен з цих методів спрямований на покращення якості зображень, що забезпечує більш точне розпізнавання тексту. Послідовність дій оброблення зображень також була описана: від етапу завантаження зображення до його обрізки до областей з концентрацією тексту. Ці кроки є важливими для високої точності та надійності розпізнавання тексту за допомогою Vision Framework.

2.4 Проектна архітектура системи та взаємозв'язок компонентів

Розглянемо нижче ключові архітектурні принципи, що є визначальними для побудови даної системи:

– модульність: система побудована на основі створення модулів, що дає змогу розділити їх на компоненти, мало залежні один від одного, але доступні для перевикористання;

– інтеграція сторонніх сервісів для спрощення розроблення: використання сервісів, як от, Google, Firebase ML Kit, Vision Framework, дасть змогу спростити частину роботи для інтеграції;

– легкість внесення змін: архітектура розроблена так, що внесення змін, оновлення старого функціоналу та внесення нового функціоналу відбувається без значних перебоїв у роботі програми.

Відповідно до функціональних та структурних вимог системи, було спроектовано структуру програмної реалізації, що зображена на рисунку 2.8.



Рисунок 2.8 – Інформаційна структура програмного забезпечення

Інформаційна система включає:

- підсистему роботи з базою даних;
- підсистему користувацького створення;
- підсистему авторизації;
- підсистему зчитування тексту з фотозображень;
- підсистему статистики.

Підсистема роботи з відомостями бази даних включає роботу з сутностями. В базі даних зберігатимуться дані про користувачів, підписки, витрати та статистику. Завдяки методам взаємодії з БД (База даних), буде виведено інформацію для користувача в застосунку. Основні функції управління даними наступні: створення, читання, редагування, видалення – CRUD (Create Read Update Delete).

Підсистема користувацького створення включає методи створення сутностей доступні користувачу через UI (User Interface). Користувач має змогу

створити акаунт, після чого йому буде доступно створення підписок та витрат, та редагування власної інформації.

Підсистема авторизації – авторизація через Google використовується для аутентифікації користувача та забезпечення доступу до особистих даних.

Підсистема зчитування тексту з фотозображень – включає методи роботи з фотозображеннями, з використанням інструменту ML Kit та Vision Framework, що дасть змогу ефективно зчитувати та аналізувати текст на фото, повертаючи вхідні дані у вигляді стрічки тексту.

Підсистема статистики – статистика показуватиме користувачу його поточні витрати за певний місяць. Завдання статистики – показати скільки та на що витратив користувач, в стиснутому вигляді. Для цього використовуються додаткові функції, що відповідають за збір даних, калькуляцію та конвертацію.

Для розроблення архітектури схеми інтерфейсів користувачів, необхідно визначити наступні моменти:

- цільова аудиторія;
- дизайн та елементи навігації – розроблення інтерфейсів та навігації;
- реакція програми на дії користувача;
- технічні обмеження та інтеграція сторонніх сервісів.

Нараз у системі передбачено два типи користувачів:

а) гості, що мають обмежений доступ до застосунку; вони можуть потрапити на вікно реєстрації, але інша частина програми не буде доступна поки користувач не авторизується;

б) зареєстровані користувачі, що мають доступ до всіх базових функцій системи, як от, перегляд та редагування профілю; створення, перегляд та видалення підписок; створення, видалення, редагування та перегляд витрат; перегляд статистики; ці користувачі мають доступ до даних, але не можуть змінювати критичні системи налаштувань;

Можливості користувачів регулюються правилами колекції БД, що подані в таблиці 2.1.

Таблиця 2.1 – Правила колекції бази даних

Колекція	Дія	Умова
users	read	if signedIn()
	create	if true
	delete	if signedIn()
	update	Not allowed
subscriptions	read	if signedIn()
	create	if signedIn()
	delete	if signedIn()
	update	if signedIn()
expenses	read	if signedIn()
	create	if signedIn()
	delete	if signedIn()
	update	if signedIn()
statistics	read	if signedIn()
	create	if signedIn()
	delete	if signedIn()
	update	if signedIn()

Правила поширюються на кожну колекцію бази даних та регулюються булевими значеннями та наявністю `uid` в користувача, що вказує на наявність даних користувача в системі, тобто чи залогінений він чи ні.

Відповідно до виконуваних груп функцій, виділено такі підсистеми (рисунок 2.9).

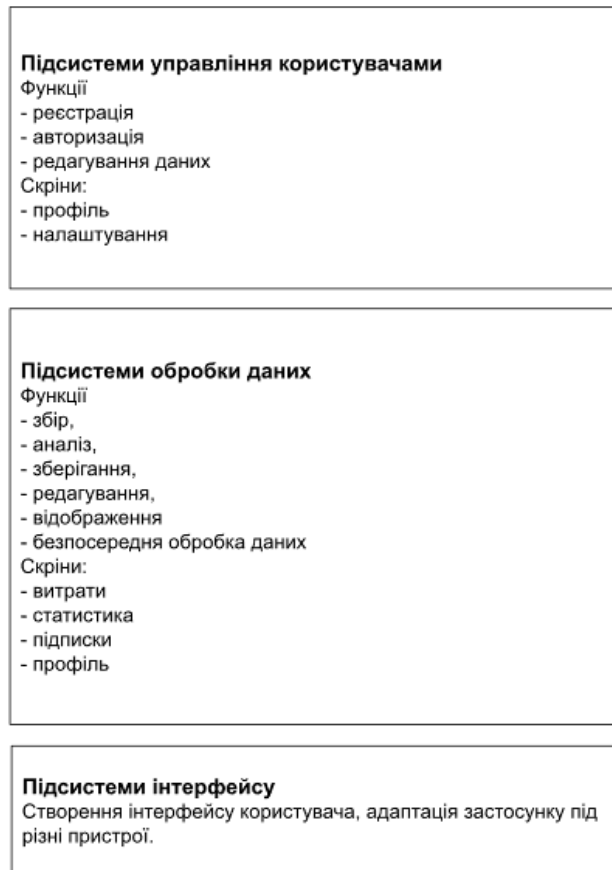


Рисунок 2.9 – Правила бази даних для користувачів

1. Підсистеми управління користувачами:
 - функції: реєстрації, авторизація, редагування профіля;
 - світлини з екрана: профіль, налаштування.
2. Підсистема обробки даних:
 - функції: збір, аналіз, зберігання, редагування, відображення та обробка даних;
 - світлини з екрана: витрати, статистика, підписки, тощо.
3. Підсистема інтерфейсу:
 - створення інтерфейсу користувача, адаптація застосунку під різні пристрої.

Структура системи робить її гнучкою та здатною масштабуватися в залежності від потреб користувача. Було описано різні підсистеми та групи функцій, які взаємодіють між собою. Одним з головних аспектів системи є

можливість адаптуватися до змін. Завдяки модульній структурі можна легко додавати нові функції.

Завдяки подібному плануванню та проектуванню було створено продукт, що не тільки відповідає потребам, але й готовий до змін та покращень.

2.5 Проектування бази даних програмного забезпечення

Зосередимося на даталогічній моделі БД програмного застосування, що є ключовим компонентом у забезпеченні функціональності системи. Будь-який сучасний програмний продукт вимагає використання БД для зберігання та обробки даних.

На рисунку 2.10 представлено ER-діаграму (Entity-relationship).

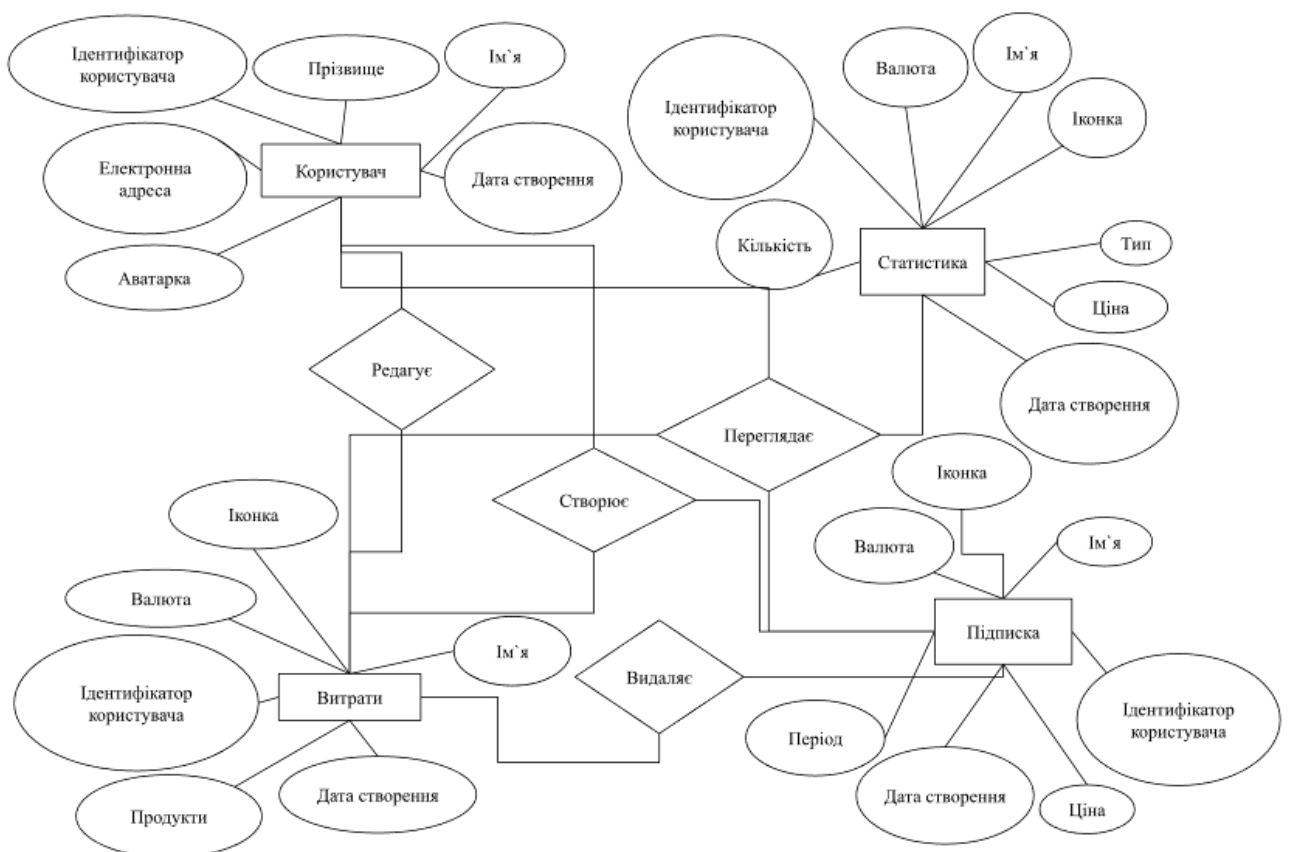


Рисунок 2.10 – ER-діаграма системи

Діаграма демонструє структуру та зв'язки між об'єктами. Ця діаграма ілюструє структуру даних та основні взаємозв'язки між сутностями системи.

1. Користувач: представляє користувачів застосунку, які авторизуються через систему Google автентифікації. Ця сутність містить атрибути, що містять інформацію про користувача. Ця інформація доступна для перегляду та оновлень.

2. Підписка: дана сутність дає змогу користувачам додавати свої підписки. Кожен документ про підписку містить такі дані як: назва підписки, валюта та ціна. Користувачі можуть додавати, переглядати та видаляти підписки.

3. Витрати: відображає фінансові витрати користувача. Кожен документ витрат пов'язаний зі списком продуктів, що відображаються, назву та ціну кожного продукту та містить такі атрибути, як сам список, тип. Користувачі можуть виконувати операції CRUD із цією сутністю.

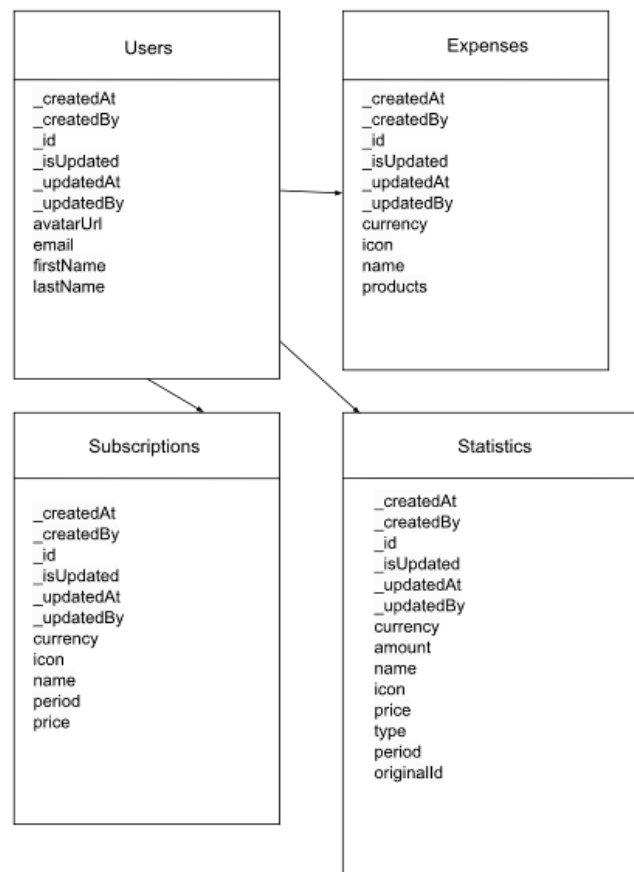


Рисунок 2.11 – Даталогічна модель бази даних

4. Статистика: сутність пов'язана із сутністю витрати та підписки. Ця сутність зберігає інформацію необхідну для відображення статистики. Атрибути включають назву, ціну та інші дані статистики.

Розглянемо структуру бази даних. На рисунку 2.11 зображено даталогічну модель БД програмної реалізації. В продемонстрований БД є такі таблиці: Users, Expenses, Subscriptions, Statistics. Кожна таблиця призначена для зберігання даних про окрему сутність. Таблиця «Users» призначена для збереження даних про користувачів (таблиця 2.2).

Таблиця 2.2 – Атрибути таблиці «Users»

№ п/п	Назва	Тип даних	Опис
1.	_id	string	Первинний ключ. Унікальний ідентифікатор користувача
2.	firstName	string	Ім'я користувача.
3.	lastName	string	Прізвище користувача.
4.	avatarUrl	string	Фото користувача.
5.	email	string	Електронна адреса користувача.
6.	_createdAt	timestamp	Час створення документа.
7.	_createdBy	string	Унікальний ідентифікатор користувача: ким було створено документ.
8.	_isUpdated	boolean	Булеве значення, вказує на те чи було документ оновлено чи ні.
9.	_updatedAt	timestamp	Час оновлення документа.
10.	_updatedBy	string	Унікальний ідентифікатор користувача: ким було оновлено документ.

Таблиця «Expenses» призначена для збереження даних про витрати (таблиця 2.3).

Таблиця 2.3 – Атрибути таблиці «Expenses»

№ п/п	Назва	Тип даних	Опис
1.	_id	string	Первинний ключ. Унікальний ідентифікатор документа.
2.	currency	string	Обрана валюта. Валюта константне значення, на вибір з доступного масиву – ['UAH', 'EUR', 'USD'].
3.	name	string	Назва списку.
4.	products	object	Список продуктів: {ключ: значення}, де ключ – назва продукту, значення – ціна продукту.
5.	icon	string	Ідентифікатор іконки, вибраної користувачем.
6.	_createdAt	timestamp	Час створення документа.
7.	_createdBy	string	Унікальний ідентифікатор користувача: ким було створено документ.
8.	_isUpdated	boolean	Булеве значення, вказує на те чи було документ оновлено чи ні.
9.	_updatedAt	timestamp	Час оновлення документа.
10.	_updatedBy	string	Унікальний ідентифікатор користувача: ким було оновлено документ.

Таблиця «Subscriptions» призначена для збереження даних про витрати (таблиця 2.4).

Таблиця 2.4 – Атрибути таблиці «Subscriptions»

№ п/п	Назва	Тип даних	Опис
1.	_id	string	Первинний ключ. Унікальний ідентифікатор підписки.
2.	currency	string	Обрана валюта. Валюта константне значення, на вибір з доступного масиву – ['UAH', 'EUR', 'USD'].
3.	name	string	Назва списку.
4.	period	string	Обраний період. Період – константне значення, на вибір з доступного масиву – ['week', 'month', 'year'].
5.	icon	string	Ідентифікатор іконки, вибраної користувачем.
6.	price	number	Ціна, вказана користувачем.
7.	_createdAt	timestamp	Час створення документа.
8.	_createdBy	string	Унікальний ідентифікатор користувача, ким було створено документ.
9.	_isUpdated	boolean	Булеве значення, що вказує на те чи було документ оновлено чи ні.
10.	_updatedAt	timestamp	Час оновлення документа.
11.	_updatedBy	string	Унікальний ідентифікатор користувача, ким було оновлено документ.

Таблиця «Statistics» призначена для збереження даних про витрати (таблиця 2.5).

Таблиця 2.5 – Атрибути таблиці «Statistics»

№ п/п	Назва	Тип даних	Опис
1.	_id	string	Первинний ключ. Унікальний ідентифікатор користувача.
2.	currency	string	Обрана валюта. Валюта константне значення, на вибір з доступного масиву – ['UAH', 'EUR', 'USD'].
3.	name	string	Назва списку.
4.	amount	number	Кількість елементів в списку.
5.	icon	string	Ідентифікатор іконки, вибраної користувачем.
6.	price	number	Ціна, вказана користувачем.
7.	type	string	Тип документа статистики. Тип – константне значення, на вибір з доступного масиву – ['EXPENSES', 'SUBSCRIPTIONS'].
8.	period	string/null	Обраний період. Для типу підписки період константне значення, на вибір з доступного масиву – ['week', 'month', 'year']. В іншому випадку = null.
9.	originalId	string	Первинний ключ. Унікальний ідентифікатор оригінального документа.

10.	_createdAt	timestamp	Час створення документа.
11.	_createdBy	string	Унікальний ідентифікатор користувача, ким було створено документ.
12.	_isUpdated	boolean	Булеве значення, що вказує на те чи було документ оновлено чи ні.
13.	_updatedAt	timestamp	Час оновлення документа.
14.	_updatedBy	string	Унікальний ідентифікатор користувача, ким було оновлено документ.

Було детально розглянуто структуру бази даних для нашого застосунку. Вона містить кілька основних таблиць: Користувачі, Підписки, Витрати, та Статистику. Кожна таблиця має свої атрибути та зв'язки, які відповідають вимогам застосунку, таким як управління фінансовою інформацією користувачів, автентифікація через Google, та аналіз витрат за допомогою OCR.

Завдяки такому підходу БД забезпечує зручний доступ до даних. Створення цієї бази даних є важливим етапом у розробленні надійного та ефективного застосунку.

2.6 Підготовка робочих вхідних даних для системи

Для даного ПЗ важлива робота з даними. Проте не всі дані приходять від користувачів – є різні типи даних які мають різне джерело. Розглянемо їх нижче.

1. Формування даних для системи розпізнавання фото. Дані для системи розпізнавання фото беруться з фотографій які завантажує користувач. Цілий процес включає кілька етапів: завантаження фото, його додаткова обробка для покращення зображення, безпосереднє розпізнавання тексту та його зчитування. Вихідні дані з цього процесу використовуються для автоматизованого заповнення деталей витрат у системі.

2. Використання констант. Для структурованості даних, система використовує низку констант, таких як типи валют, типи даних (“expenses” або “subscriptions”), а також періодичність підписок (“week”, “month”, “year”). Ці константи не тільки спрощують керування даними в системі, але й допомагають користувачам краще організувати свої фінанси, адже використовуються при створенні та відображенні даних.

Типи списків (наприклад, покупки, медицина тощо) та підписок (такі як Apple Music, Spotify тощо) визначаються на основі найпопулярніших варіантів використання. Кожен тип має унікальний ідентифікатор, назву та колір, що дає змогу користувачам легко орієнтуватися.

3. Іконки та категоризація. Кожен тип списку та підписки має визначену іконку. Іконки допомагають користувачам швидко орієнтуватися між категоріями та вибирати потрібні опції.

4. Зовнішні дані. Всі інші дані завантажує або змінює користувач і вони зберігаються в базі даних.

Використання структурованих вхідних даних та констант дає змогу даній системі ефективно функціонувати та надає зручний та інтуїтивно зрозумілий інтерфейс. Завдяки визначеним джерелам даних та методам їх обробки, користувачі можуть ефективніше планувати свої витрати та оптимізувати фінансове планування.

2.7 Особливості використання спеціалізованих програмних компонентів

У даному проєкті використано низку спеціалізованих бібліотек, які забезпечують необхідні функції для роботи застосунку та полегшують розроблення. Нижче наведено детальний опис кожної з них.

1. @react-native-firebase [25]:

– ця бібліотека є обгорткою для Firebase, яка дає змогу інтегрувати різні сервіси Firebase в застосунки на React Native; вона підтримує аутентифікацію, аналітику, бази даних і багато іншого;

– використовується для аутентифікації користувачів, зберігання даних та управління сесіями користувачів.

2. @moment [26]:

– бібліотека для роботи з датами та часом у JavaScript, яка дає змогу легко маніпулювати, перевіряти та формувати дати;

– використовується для оброблення та відображення дат в системі.

3. @react-native [27]:

– бібліотека дає змогу створювати нативні мобільні застосунки, використовуючи React;

– дає змогу створювати інтерфейс користувача та управляти взаємодією.

4. @react-native-text-recognition [28]:

– бібліотека інтегрує можливості ML Kit від Firebase та Vision Framework у застосунки React Native, даючи змогу використовувати машинне навчання для розпізнавання тексту, зображень та інших об'єктів;

– використовується для розпізнавання тексту на чеках через OCR, щоб автоматизувати введення даних про витрати.

5. @react-native-google-signin/google-signin [29]:

– дає змогу легко інтегрувати функціонал Google Sign-In в мобільні застосунки на React Native.

– використовується для надання користувачам можливості входити в систему за допомогою своїх Google облікових записів, забезпечуючи швидкий та безпечний доступ до застосунку.

Використання цих бібліотек у проєкті значно спрощує розроблення та покращує функціональність застосунку. Ці бібліотеки дозволяють використовувати сучасні технології для ефективного управління даними та користувацького інтерфейсу, роблячи застосунок більш надійним і доступним.

2.8 Висновки до розділу 2

У цьому розділі розроблено метод розпізнавання тексту на фотозображеннях нейромережевими засобами, який використовується для створення застосунку. Розроблений метод розпізнавання тексту використовує технології машинного навчання для оброблення фотозображень, розпізнавання та витягування тексту з них. Це дає змогу автоматизувати процес розпізнавання даних та забезпечити точність розпізнавання.

У розділі також описано функціональну структуру програмної реалізації, що включає такі функції як авторизація, управління підписками та витратами, доступ до статистики. Розглянуто навігаційну систему, що є важливим кроком при плануванні застосунку, проєктну архітектуру та продемонстровано взаємозв'язок компонентів та БД. Даталогічна модель бази даних ілюструє наявні сутності та детальний опис даних.

Варте уваги те, що використання сторонніх сервісів та бібліотек має сприяти ефективному розробленню застосунку, підвищуючи його функціональність та покращуючи користувацький досвід, забезпечуючи інтеграцію сучасних технологій та рішень для управління даними, безпеки, та користувацького інтерфейсу.

Завдання на подальшу роботу є такими.

1. Виконати програмну реалізацію методу у вигляді застосунку на основі описаних методів та технологій.
2. Дослідити якість виконання методу OCR, а саме, провести низку тестів для оцінювання точності та швидкості розпізнавання тексту.
3. Виконати тестування всіх компонентів системи для оцінювання коректності та надійності роботи застосунку.
4. Оцінити зручність та інтуїтивність інтерфейсу.

Поставлені задачі дадуть можливість розробити продукт, перевірити його технічну відповідність та ефективність та підготувати застосунок до фінального кроку.

Розділ 3 Програмна реалізація мобільного застосунку для розпізнавання тексту на фотозображеннях товарних чеків

3.1 Визначення шляхів дослідження та засобів створення програмного забезпечення

Для реалізації поставленої мети було створено застосунок з поданим нижче функціоналом:

- робота зі списками витрат;
- робота з підписками;
- робота зі статистикою;
- робота з зчитуванням тексту з зображень.

Зчитування тексту відбувається наступним чином:

- завантаження фотозображення чеку;
- оброблення зображення;
- розпізнавання тексту;
- парсинг та інтерпретація розпізнаного тексту;
- збереження та візуалізація даних у список.

Для оцінювання ефективності роботи застосунку будуть використовуватися наступні параметри:

- точність розпізнавання символів: наскільки точний результат розпізнавання тексту з фотозображень;
- час обробки зображення: час, що необхідний для оброблення зображення, включно з розпізнавання тексту та парсингом;
- зручність та зрозумілість застосунку: оцінка зручності, легкості взаємодії з інтерфейсом та простоти в використанні;
- кількість збоїв в застосунку: кількість помилок чи збоїв, що виникають під час користування додатком;
- функціональність застосунку, що включає CRUD операції зі списками, підписками та інші функції.

Далі буде детально розглянуто структуру, функціональність та реалізацію даного застосунку. Буде проведено аналіз архітектури програмного забезпечення та описані основні компоненти і їх взаємодія. Після цього будуть продемонстровані результати тестування.

3.2 Вибір засобів розроблення програмного забезпечення

Для вирішення поставленої задачі потрібно використати платформу Node.js, мову програмування JavaScript, середовище програмування – редактор коду VS Code (Visual Studio Code), систему керування базами даних СКБД (Система керування базами даних) Firebase, фреймворк користувацького інтерфейсу React Native.

Node.js – це серверне середовище виконання JavaScript [30]. Дана платформа пропонує асинхронну модель, що є ефективною в обробці великої кількості запитів. Її використання також забезпечує високу продуктивність та доступ до багатьох модулів та бібліотек через npm (Node Package Manager).

До переваги даної платформи можна віднести наступне:

- швидкість та ефективність у роботі із запитами;
- уніфікація, що забезпечує роботу на сервері та на клієнті, з використанням однієї мови – в даному випадку JS (JavaScript);
- добре підходить для масштабованих додатків.

Головним недоліком є монопотоківість, що є проблемою для CPU-інтенсивних задач, проте не стає на заваді виконання поточного встановленого завдання.

Firebase – це хмарна СКБД [31]. Дана СКБД має низку можливостей, до прикладу підтримку реального часу баз даних, хостинг, аналітику та звісно аутентифікацію. БД є NoSQL (Not only Structured Query Language) завдяки чому забезпечує синхронізацію даних в реальному часі.

Було визначено такі переваги:

- миттєве оновлення даних;

- легкість використання та інтеграції;
- можливість інтеграції з сервісами Google.

Недоліки СКБД є такими:

- робота з базою вимагає постійного зв'язку з мережею;
- не завжди зручна для складних реляційних даних.

JSX (JavaScript XML) – є синтаксичним розширенням мови програмування JavaScript та використовується задля написання React застосунків [32]. Дане розширення має на меті використання HTML-подібного (Hypertext Markup Language) синтаксису для створення елементів користувацького інтерфейсу, та за допомогою транслятора Babel перетворюється в виклики JavaScript функцій.

Щодо переваг, виділено такі:

- є досить зручним, адже дає змогу використовувати всім знайомий синтаксис HTML у JavaScript;
- є інтуїтивно зрозумілим в роботі, що полегшує використання;
- дає можливість без проблем вбудовувати вирази JavaScript.

Недоліки JSX є такими:

- потрібен транслятор, що перетворить код в виклики JS;
- може бути незвичайним для розробників, що звикли до традиційного підходу з HTML і JavaScript.

React Native – фреймворк для розроблення мобільних додатків, що забезпечує розроблення з використанням JS та React [33].

Визначено переваги фреймворку:

- забезпечує кросплатформну розроблення, адже для iOS та Android використовується єдина кодова база;
- забезпечує швидку розроблення;
- підтримує багато бібліотек та має велику спільноту розробників.

Основним недоліком є обмеженість нативної функціональності.

У ролі середовища програмування використовується Visual Studio Code (VS Code) [34]. Це редактор коду, що пропонує підтримку великого переліку мов

програмування, надає доступ до величезної кількості плагінів та розширень та також має зручний інтерфейс терміналу та підтримки Git.

До його переваг відносять таке:

- легкість використання;
- можливість розширення функціоналу через використання плагінів;
- інтеграція з терміналом та Git системою.

До недоліків варто віднести ресурсоемність, адже чим більший проєкт тим більше ресурсів він споживає.

Xcode – є інтегрованим середовищем розроблення ПЗ (Програмне забезпечення), що використовується на платформах Apple [35]. Дане IDE (Інтегроване середовище розроблення) підтримує розроблення для: iOS, watchOS, iPadOS, macOS та представляє собою одразу і редактор коду, і інтерфейсний редактор, і інструменти аналізу продуктивності. Система пропонує широкий вибір симуляторів для запуску та тестування додатків на різних версіях ОП та різних моделях пристроїв. Також, будучи частиною екосистеми Apple, підтримує Swift, Objective-C та подібні мови.

Його переваги є такими:

- наявність всіх необхідних інструментів для повної розроблення додатків;
- є офіційним середовищем від Apple та забезпечує повну сумісність з платформою;
- наявність симуляторів для тестування;
- містить інструменти для налагодження та тестування.

До недоліків варто віднести таке:

- вимога macOS, що обмежує використання;
- ресурсоемність.

Отже, для розроблення даної системи, було обрано наступну комбінацію засобів розроблення:

- платформа Node.js;
- редактор коду VS Code;

- СКБД Firebase;
- фреймворк користувацького інтерфейсу React Native;
- синтаксичне розширення JSX;
- інтегроване середовище розроблення (IDE) Xcode.

Подані засоби забезпечують повноцінну ефективну розроблення, швидку обробку даних у режимі реального часу та можливість підтримки мобільного застосунку. Описані інструменти було обрано, як зручні та ефективні засоби для розроблення кросплатформеного мобільного застосунку, з швидкою роботою з базою даних та інтеграцією нейронних мереж.

3.3 Структура та функціональне призначення програмних складових системи

Основною функціональністю розробленого застосунку, що націлений на управління фінансами, є наступні ключові модулі (рисунок 3.1).

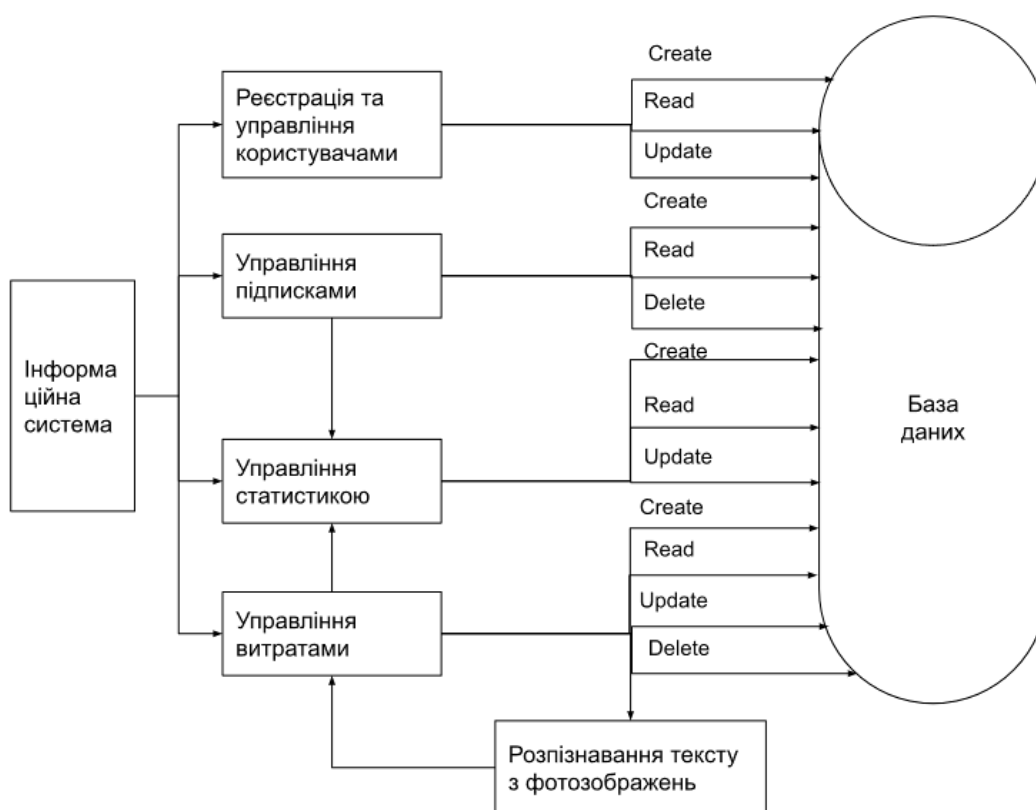


Рисунок 3.1 – Діаграма компонентів програми

1. Реєстрація та управління користувачами. Користувачі застосунку можуть створювати облікові записи, через реєстрації через Google. Цей тип реєстрації спрощує процедуру входу та надає додаткову безпеку даних. Крім того, даний модуль дає змогу редагування даних користувача, таких як ім'я, прізвище, електронна пошта та аватарка.

2. Управління підписками. Застосунок дає змогу створювати, видаляти та переглядати підписки. Цей модуль допомагає слідкувати за наявними та минулими підписками, адже ці дані також записуються в статистику.

3. Управління витратами. Програма надає функціонал створення, редагування, видалення та перегляду витрат. Створення підписки дає змогу зберегти список продуктів з цінами під певною категорією.

4. Відображення статистики. Програма збирає статистичні дані з витрат та підписок користувачів та візуалізує їх. Таким чином, користувачі можуть переглядати дані статистики помісячно та бачити діаграму витрат.

5. Розпізнавання тексту з фотозображень. Завдяки використанню технології оптичного розпізнавання символів та нейромережі, програма може зчитувати текстову інформацію з фотозображень. Дана функція націлена на пришвидшене створення списку витрат, зокрема на швидше введення даних з товарних чеків.

Архітектура застосунку ґрунтується на основі компонентного підходу. Кожна функція реалізована у вигляді окремого модуля. Для реалізації використано шаблон Model-View-ViewModel із взаємодією таких елементів:

- model – використовується БД Firebase, що в першу чергу виконує функцію бази даних та зберігає усі необхідні дані;

- view – користувацький інтерфейс(UI), створений на основі React Native, що забезпечує зручну взаємодію користувачів з застосунком;

- viewModel – модулі логіки, які взаємодіють з Model та View, що забезпечує обробку та відображення даних у користувацькому інтерфейсі.

Передача даних між модулями системи здійснюється двома способами:

а) глобальний стан – вбудована функція React, що дає змогу компонентам керувати даними та зберегти їх у глобальному стані;

б) props – передача пропсів (props) вниз по ієрархії компонент для ефективного обміну між дочірніми та батьківськими компонентами.

Модулі програми є взаємопов'язаними.

Модуль аутентифікації забезпечує користувачам доступ до функціоналу застосунку та надає дані користувача іншим модулям. Модулі керування підписками та витратами передають дані модулю статистики, для їх подальшої візуалізації. Модуль розпізнавання тексту передає зчитаний текст до модуля керування витратами, допомагаючи автоматизувати процес введення даних. Таке архітектурне рішення забезпечує ефективність та гнучкість системи, дозволяючи легко змінювати та додавати новий функціонал.

Розглянемо взаємодію модулів системи на прикладі створення витрати. Діаграма зображена на рисунку 3.2.

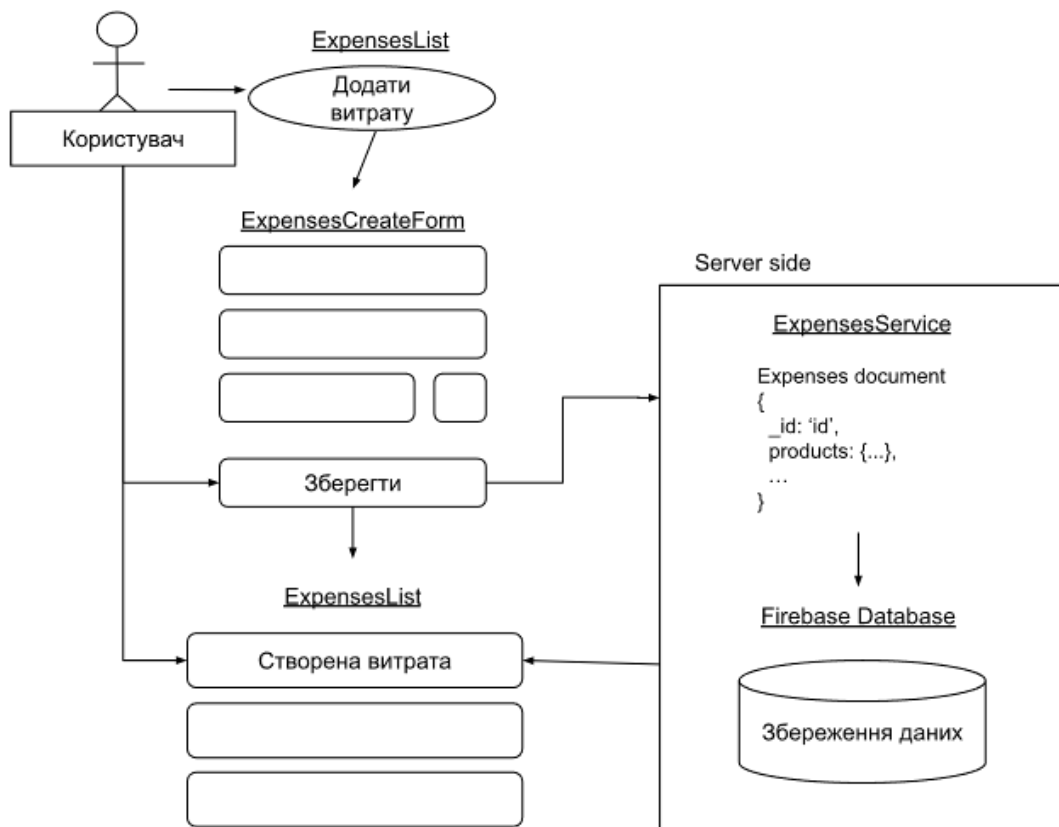


Рисунок 3.2 – Взаємодія модулів системи

На рисунку 3.2 бачимо, що користувач ініціює процес створення витрати, натиском відповідної кнопки на сторінці ExpensesList що відображає список усіх наявних витрат та надає змогу перейти на форму створення. Після цієї дії користувач потрапляє на компоненту створення ExpensesCreateForm, де вводить усі необхідні дані та натискає на кнопку збереження. Далі в користувацькому інтерфейсі відбувається перенаправлення на сторінку зі списком ExpensesList, де користувач бачить щойно створену ним витрату. Водночас модуль ExpenseService отримує дані з форми та взаємодіє з базою даних. Firebase Database отримує дані для нового документа, назву колекції та зберігає ці дані в відповідне місце, після чого повертає підтвердження успішного збереження.

Для збереження даних використано колекції Firebase (рисунок 3.3):

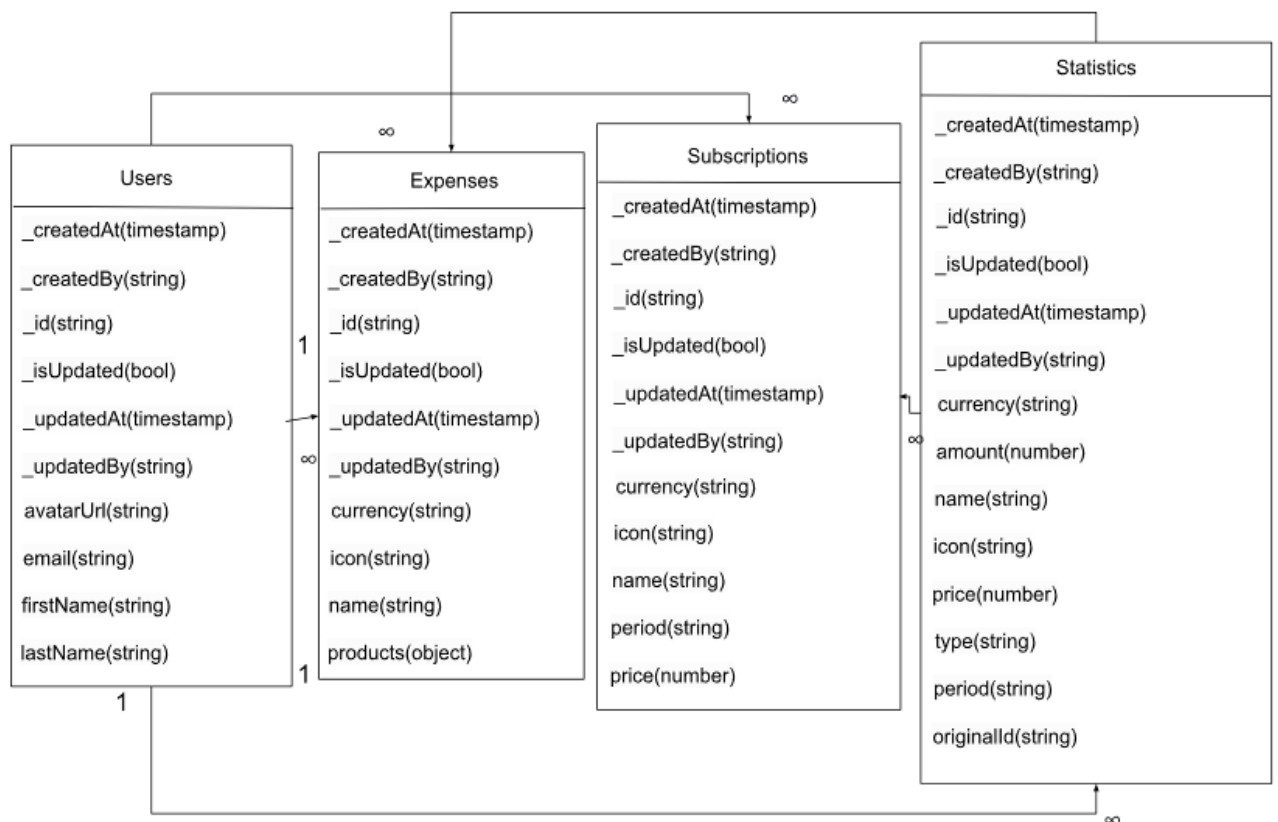


Рисунок 3.3 – Модель даних Firebase

- users – зберігає інформацію користувачів та дані облікових записів;
- subscriptions – зберігає інформацію про підписки;
- expenses – зберігає інформацію про витрати;

– statistics – зберігає інформацію про статистику витрат і підписок.

Розглянемо діаграму випадків використання, для кращого розуміння структури та функціональності розробленого застосунку. Діаграма, що зображена на рисунку 3.4, демонструє який функціонал доступний користувачам та візуалізує взаємодію користувача з додатком.

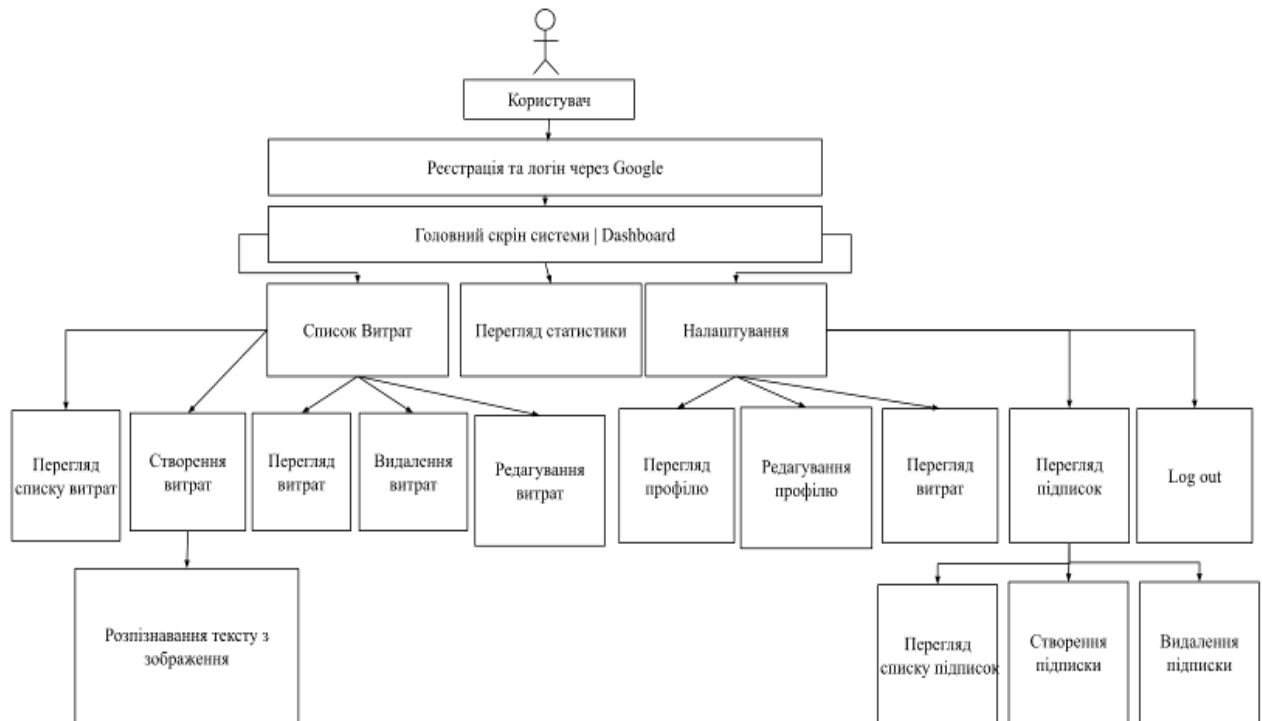


Рисунок 3.4 – Use Case Diagram

Дана діаграма ілюструє можливості взаємодії з додатком, такі як реєстрація, управління підписками, витратами, розпізнавання тексту, що допомагає краще зрозуміти процеси системи та взаємодію з нею.

Описана архітектура проєкту та створенні діаграми забезпечують розуміння функціональності програмного продукту, що закладає основи для подальшого розвитку проєкту та його оптимізації.

3.4 Особливості реалізації програмних складових системи

Розберемо особливості реалізації програмних складових системи. Далі буде наведено опис рішень певних завдань та алгоритмів, які було реалізовано для написання програми.

Мета даного застосунку – спростити користувачам контроль над своїми фінансами. В процесі розроблення було інтегровано відомі сервіси та рішення. Основний функціонал застосунку – авторизація користувача, робота за підписками та витратами, відображення статистики та алгоритм розпізнавання тексту з фотозображень.

Розглянемо імплементацію авторизації та управління користувачем. Механізм реєстрації реалізовано через інтеграції Google OAuth. Після відкриття застосунку, користувач бачить сторінку авторизації, де може увійти або створити акаунт за допомогою свого наявного облікового запису Google. Після аутентифікації, Firebase отримує його дані та зберігає.

1. Ініціалізація: користувач натискає кнопку “Login with Google” (рисунок 3.5).

2. Аутентифікація: користувача перенаправлено на сторінку аутентифікації Google.

3. Отримання токена доступу: після аутентифікації, повертається токен доступу від Google.

4. Збереження: дані користувача збережено в базі даних, користувач попадає на головну сторінку застосунку (рисунок 3.6).

Розглянемо як відбувається взаємодія функціональних елементів з базою даних. Цей процес включає наступні етапи реалізації.

1. Ініціалізація: першим кроком необхідно ініціалізувати Firestore. На цьому етапі налаштовується з'єднання з Firebase, налаштовуються та підключаються конкретні модулі: авторизація, БД, зберігання тощо.

2. Введення даних: через інтерфейс застосунку користувач вводить дані.



Рисунок 3.5 – Сторінка авторизації

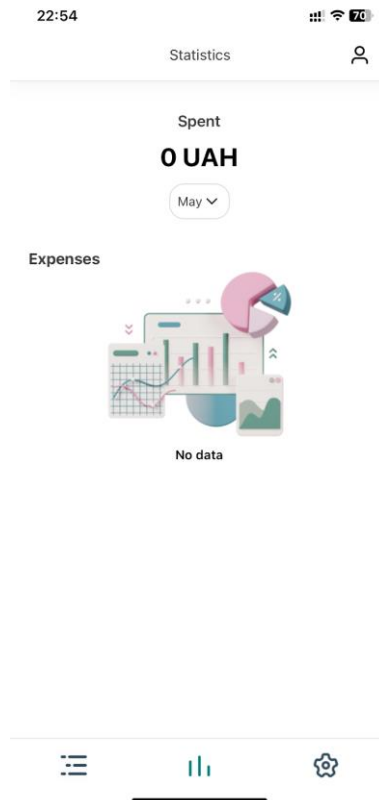


Рисунок 3.6 – Головна сторінка застосунку

3. Валідація: дані валідуються, щоб перевірити чи поля заповнені коректно.

4. Запит: формується запит до Firestore. Запит включає назву колекції, куди будуть зберігатися дані та дані у вигляді JSON-об'єкту.

5. Відповідь: після отримання запиту, Firestore обробляє його та повертає відповідь.

6. Оброблення помилок: якщо Firestore повертає помилку, вона обробляється належним чином.

Оновлення інтерфейсу: після отримання відповіді від Firestore інтерфейс користувача оновлюється, відповідно до результату запиту.

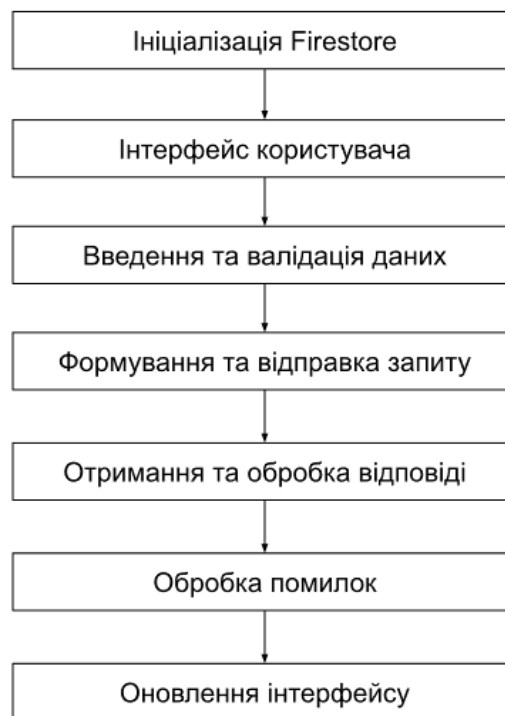


Рисунок 3.7 – Схема алгоритму взаємодії з базою даних

За подібним сценарієм здійснюються всі CRUD операції над сутностями, що забезпечує зручний процес взаємодії з базою даних.

Розглянемо алгоритм збору та відображення статистики:

1. Збір даних – включає дані з витрат та підписок, разом з додатковими полями, які вказують на кількість продуктів, період та оригінальний ідентифікатор документу, зокрема:

– коли користувач додав нову підписку або новий список витрат, дані записуються в документ статистики;

– у випадку з підпискою, відповідно до вказаної періодичності, створюються запити до бази даних, що періодично записують новий документ поки підписка дійсна;

– у випадку з витратами, в яких доступна функція редагування, документ статистики оновлюється одночасно з документом витрат;

– дані зберігаються в колекції "Statistics"; ця колекція містить всю інформацію про поточні та минулі фінансові дані користувача та не видаляється.

2. Відображення статистики – статистика візуалізована в вигляді діаграми та списку витрат за вказаний місяць. Місяць обирається користувачем на головному екрані програми (рисунок 3.8):



Рисунок 3.8 – Відображення статистики

– фільтрація: користувач обирає місяць по якому хоче побачити статистику, після чого програма фільтрує запити у колекції "Statistics" та витягує дані відповідно до обраного періоду;

– відображення списку витрат: дані статистики трансформуються, за потреби, та відображаються у вигляді списку, де кожен елемент містить: іконку, назву, кількість продуктів та ціну з вказаною валютою;

– відображення діаграми: дані статистики трансформуються, за потреби, визначається сума коштів, що була витрачена на кожен день місяця; якщо ціни були вказані в різних валютах, відбувається конвертація згідно з курсом до гривні; в результаті формується діаграма, що ілюструє витрати.

Наведений алгоритм забезпечує ефективний збір статистики та відображення зібраних даних, що дає змогу користувачам отримувати детальний огляд та контролювати свої витрати.

Розглянемо алгоритм, що використовується для розпізнавання тексту. Даний алгоритм використовує технології OCR задля зчитування тексту з фотозображень, що дає змогу пришвидшити запис даних при створенні списку витрат.

1. Вхідні дані: користувач завантажує або робить фото чеку покупок (рисунок 3.9).
2. Оброблення: зображення обробляється методами OCR та нейромережі.
3. Вихідні дані: відбувається розпізнавання та зчитування тексту.
4. Додавання витрат: розпізнаний текст додається до списку витрат (рисунок 3.10).



Рисунок 3.9 – Завантаження фото чеку для зчитування тексту

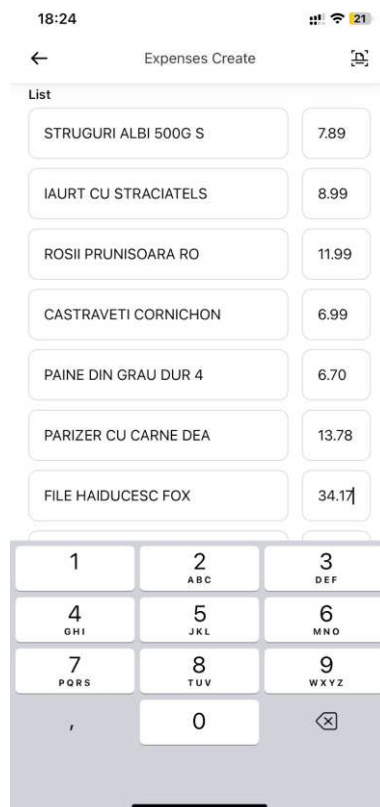


Рисунок 3.10 – Результат додавання списку покупок з завантаженого фотозображення

Вище було розглянуто опис програмної реалізації основних модулів застосунку. Було продемонстровано опис ключових алгоритмів та особливості їх реалізації.

3.5 Тестування програмної реалізації та вимоги до розгортання

Тестування будь-якої програмної реалізації є важливим кроком в розробленні будь якого застосунку, адже це дає повне розуміння таких аспектів: а) наявність баг в застосунку, б) зручність користування та в) інтуїтивне розуміння основного функціоналу підходи до користування програмою.

Таким чином, для того, щоб перевірити коректність роботи даної програми, було проведено функціональні тестування та юніт-тестування, задля всебічної перевірки функцій.

Розглянемо найпростіший та найпоширеніший тип тестування – Functional Testing та його результати. Розглянемо функціональність роботи із підпискою.

Щоб створити підписку, необхідно перейти на сторінку налаштувань та натиснути “Subscriptions” (рисунок 3.11), після цього вас буде перенаправлено на сторінку зі списком створених підписок, якщо вони наявні.

На цій сторінці необхідно натиснути на плюстик (рисунок 3.12), щоб потрапити на форму створення.

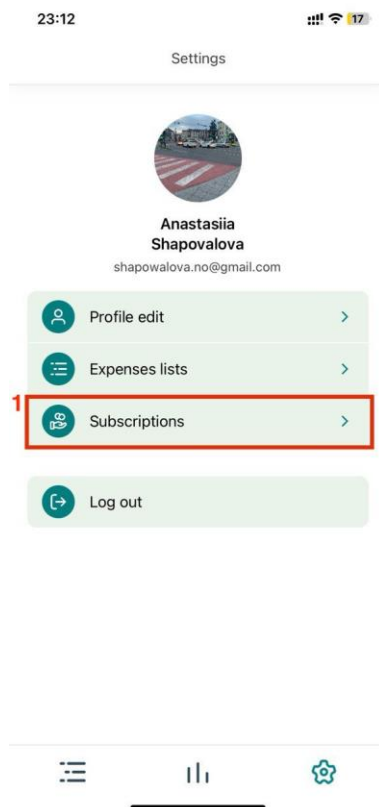


Рисунок 3.11 – Сторінка налаштувань з кнопкою “Subscriptions”: 1) блок переходу на список підписок

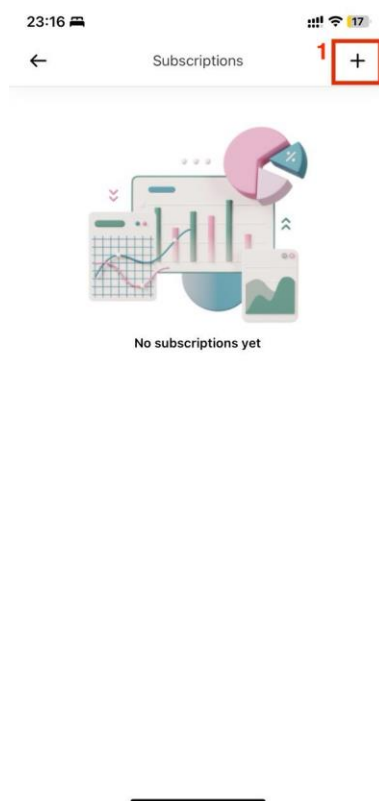


Рисунок 3.12 – Список підписок: 1) кнопка додавання підписки

На формі створення необхідно заповнити всі дані, а саме: обрати сервіс, ввести ціну та обрати валюту, та вказати період списання коштів. Після цього потрібно натиснути кнопку “Submit” (рисунок 3.13).

The screenshot shows a mobile application interface for creating a subscription. At the top, the time is 23:16 and the battery is at 17%. The title bar says "Subscription Create". The form consists of five numbered sections:

- Icon:** A grid of 10 circular icons representing different services, including Netflix, Spotify, and others.
- Service:** A text input field containing the word "Medium".
- Price:** A text input field containing the number "28" and a dropdown menu for currency set to "USD".
- Period:** Three radio button options: "weekly", "monthly", and "yearly".
- Submit:** A large teal button with the text "Submit".

Рисунок 3.13 – Форма створення підписок: 1) блок вибору іконки, 2) блок вводу назви сервісу, 3) блок введення ціни та валюти, 4) блок вибору періоду, 5) кнопка для збереження даних

Ця дія повинна запуснути функцію збереження даних в базу, після чого спрацює редірект на список підписок, серед яких має відобразитися щойно створена (рисунок 3.14).

Для подальшої демонстрації перевірки ПЗ буде продемонстровано результати тест-кейсів.

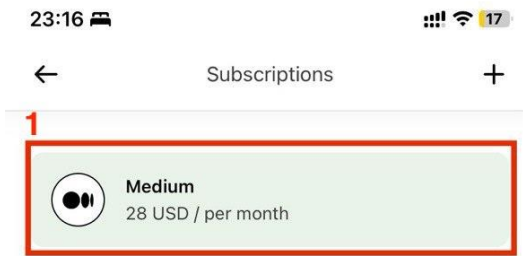


Рисунок 3.14 – Результат створення підписки: 1) блок підписки

Тест-кейс 1: Перевірка можливості реєстрації/логіну в застосунок.

Передумови: запуск програми. Кроки тест-кейс 1 є такими:

- 1) натиснути кнопку “Continue with Google”;
- 2) обрати Google-акаунт та виконати вхід.

Очікуваний результат: якщо користувач вперше в системі – його дані збережуться в системі, як дані нового користувача, після цього користувач буде перенаправлений на головну сторінку. Якщо користувач вже наявний в системі – він залогіниться в свій акаунт та буде перенаправлений на головну сторінку (таблиця 3.1).

Таблиця 3.1 – Тест-кейс TS0001

Тест-кейс ID: TS0001	Пріоритет: 1	Створено: 10.05.2024, Шаповалова А.О.
Назва: Перевірка функціоналу – реєстрації/логін.		
Кроки	Очікуваний результат	
1. Запустити програму. 2. Натиснути кнопку “Continue with Google”.	Якщо користувач вперше в системі – його дані збережуться, як дані нового користувача, після цього його буде перенаправлено на головну сторінку. Якщо користувач вже наявний в системі – він зареєструється в свій акаунт та буде перенаправлений на головну сторінку.	
Результат виконання тест-кейсу: пройдено успішно		

Тест-кейс 2: Перевірка можливості створення сутності “Expenses”.

Передумови: запуск програми, реєстрація/логін в застосунку. Кроки тест-кейс 2 є такими:

- 1) перейти на сторінку витрат;
- 2) натиснути кнопку плюс;
- 3) заповнити форму та натиснути на кнопку підтвердження створення.

Очікуваний результат: користувач буде перенаправлений на сторінку зі списком витрат, де буде відображатися створена витрата (таблиця 3.2).

Тест-кейс 3: Перевірка можливості редагування сутності “Expenses”.

Передумови: запуск програми, реєстрація/логін в застосунку, створення сутності. Кроки тест-кейс 3 є такими:

- 1) перейти на сторінку витрати;
- 2) натиснути кнопку редагування;
- 3) після потрапляння на форму змінити необхідні дані та натиснути на кнопку підтвердження.

Таблиця 3.2 – Тест-кейс TS0002

Тест-кейс ID: TS0002	Пріоритет: 1	Створено: 10.05.2024, Шаповалова А.О.
Назва: Перевірка функціоналу – створення сутності “Expenses”.		
Кроки	Очікуваний результат	
1. Запустити програму. 2. Пройти реєстрацію/логін в застосунку. 3. Перейти на сторінку витрат. 4. Натиснути кнопку плюс. 5. Заповнити форму та натиснути на кнопку підтвердження створення.	Користувач буде перенаправлений на сторінку зі списком витрат, де буде відображатися створена витрата.	
Результат виконання тест-кейсу: пройдено успішно		

Очікуваний результат: користувач буде перенаправлений на сторінку з витратою, де будуть відображатися оновлені дані (таблиця 3.3).

Тест-кейс 4: Перевірка можливості видалення сутності “Expenses”.

Передумови: запуск програми, реєстрація/логін в застосунку, створення сутності. Кроки тест-кейс 4 є такими:

- 1) перейти на сторінку з списком витрат;
- 2) потягнути витрату, яку потрібно видалити, вліво;
- 3) після того, як з’явиться червона кнопка видалення – натиснути на неї.

Очікуваний результат: Витрату буде видалено, вона зникне зі списку (таблиця 3.4).

Таблиця 3.3 – Тест-кейс TS0003

Тест-кейс ID: TS0003	Пріоритет: 2	Створено: 10.05.2024, Шаповалова А.О.
Назва: Перевірка функціоналу – редагування сутності “Expenses”.		
Кроки	Очікуваний результат	
1. Запустити програму. 2. Пройти реєстрацію/логін в застосунку. 3. Перейти на сторінку витрати. 4. Натиснути кнопку редагування. 5. Після потрапляння на форму змінити необхідні дані та натиснути на кнопку підтвердження.	Користувач буде перенаправлений на сторінку зі витратою, де будуть відображатися оновлені дані.	
Результат виконання тест-кейсу: пройдено успішно		

Тест-кейс 5: Перевірка можливості видалення сутності “Subscription”.

Передумови: запуск програми, реєстрація/логін в застосунку, створення підписки. Кроки тест-кейс 5 є такими:

- 1) перейти на сторінку з списком підписок;
- 2) потягнути підписку, яку потрібно видалити, вліво;
- 3) після того, як з’явиться червона кнопка видалення – натиснути на неї.

Очікуваний результат: Підписку буде видалено, вона зникне зі списку (таблиця 3.5).

Таблиця 3.4 – Тест-кейс TS0004

Тест-кейс ID: TS0004	Пріоритет: 2	Створено: 10.05.2024, Шаповалова А.О.
Назва: Перевірка функціоналу – видалення сутності “Expenses”.		
Кроки	Очікуваний результат	
1. Запустити програму. 2. Пройти реєстрацію/логін в застосунку. 3. Перейти на сторінку витрат. 4. Потягнути витрату, яку потрібно видалити, вліво. 5. Після того, як з’явиться червона кнопка видалення – натиснути на не	Витрату буде видалено, вона зникне зі списку.	
Результат виконання тест-кейсу: пройдено успішно		

Тест-кейс 6: Перевірка статистики.

Передумови: запуск програми, реєстрація/логін в застосунку, створення підписки, створення витрати. Кроки тест-кейс 6 є такими:

- 1) перехід на сторінку статистики;
- 2) вибір місяця в якому були додані будь-які витрати або підписки.

Очікуваний результат: користувачу відобразиться статистика по витратам у вигляді графіку та списку (таблиця 3.6).

Таблиця 3.5 – Тест-кейс TS0005

Тест-кейс ID: TS0005	Пріоритет: 2	Створено: 10.05.2024, Шаповалова А.О.
Назва: Перевірка функціоналу – видалення сутності “Subscription”.		
Кроки	Очікуваний результат	
1. Запустити програму. 2. Пройти реєстрацію/логін в застосунку. 3. Перейти на сторінку підписок. 4. Потягнути підписку, яку потрібно видалити, вліво. 5. Після того, як з’явиться червона кнопка видалення – натиснути на неї.	Підписку буде видалено, вона зникне зі списку.	
Результат виконання тест-кейсу: пройдено успішно		

Тест-кейс 7: Перевірка можливості редагування інформації користувача.

Передумови: запуск програми, реєстрація/логін в застосунку. Кроки тест-кейс 7 є такими:

- 1) перейти на сторінку налаштувань;
- 2) натиснути кнопку “profile edit”;
- 3) змінити дані та натиснути кнопку “save”.

Очікуваний результат: користувача буде перенаправлено на сторінку налаштувань, де будуть відображатися його оновлені дані (таблиця 3.7).

Таблиця 3.6 – Тест-кейс TS0006

Тест-кейс ID: TS0006	Пріоритет: 2	Створено: 10.05.2024, Шаповалова А.О.
Назва: Перевірка функціоналу – статистика.		
Кроки	Очікуваний результат	
1. Запустити програму. 2. Пройти реєстрацію/логін в застосунку. 3. Перейти на сторінку статистики. 4. Вибір місяця в якому були додані будь-які витрати або підписки.	Користувачу відобразиться статистика по витратам у вигляді графіку та списку.	
Результат виконання тест-кейсу: пройдено успішно		

Таблиця 3.7 – Тест-кейс TS0007

Тест-кейс ID: TS0007	Пріоритет: 3	Створено: 10.05.2024, Шаповалова А.О.
Назва: Перевірка функціоналу – редагування інформації користувача.		
Кроки	Очікуваний результат	
1. Запустити програму. 2. Пройти реєстрацію/логін в застосунку. 3. Перейти на сторінку	Користувача буде перенаправлено на сторінку налаштувань, де будуть відобразатися його оновлені дані.	

налаштувань. 4. Натиснути кнопку “Profile edit”. 5. Змінити дані та натиснути кнопку “Save”.	
Результат виконання тест-кейсу: пройдено успішно	

Тест-кейс 8: Перевірка можливості вийти з системи.

Передумови: запуск програми, реєстрація/логін в застосунку. Кроки тест-кейс 8 є такими:

- 1) перейти на сторінку налаштувань;
- 2) натиснути кнопку “Log out”.

Очікуваний результат: Користувача має вилогінити з системи та перенаправити на екран логіну (таблиця 3.8).

Таблиця 3.8 – Тест-кейс TS0008

Тест-кейс ID: TS0008	Пріоритет: 3	Створено: 10.05.2024, Шаповалова А.О.
Назва: Перевірка функціоналу – редагування інформації користувача.		
Кроки	Очікуваний результат	
1. Запустити програму. 2. Пройти реєстрацію/логін в застосунку. 3. Перейти на сторінку налаштувань. 4. Натиснути кнопку “Log out”.	Користувача має вилогінити з системи та перенаправити на екран логіну.	

Результат виконання тест-кейсу: пройдено успішно

У результаті проведеного тестування програмного продукту некоректно працюючих функцій виявлено не було.

Розглянемо вимоги до програмної та апаратної бази. Програмні вимоги до застосунку наведено нижче.

1. Операційна система: Android 6.0+ або iOS 13.0+.
2. Бібліотеки, фреймворки:
 - React Native;
 - ML Kit від Firebase, Vision Framework;
 - інші сторонні сервіси/бібліотеки.
3. Програмне забезпечення для розроблення застосунків:
 - середовище для розроблення Android Studio або Xcode 11+;
 - yarn або npm для керування пакетами.

На основі сумісності характеристик та ефективності розроблення, враховуючи програмні вимоги, було визначено мінімальні апаратні вимоги:

- процесор: 4-ядерний з тактовою частотою 1.8 ГГц;
- оперативна пам'ять: 2ГБ;
- місце для інсталювання застосунку;
- камера.

Виконані тести програмного забезпечення підтвердили здатність застосунку виконувати функції, які було заплановано. Отримані результати вказують на надійність та ефективність виконаного рішення, та на придатність до практичного застосування. Наведені вимоги, до мінімального програмного та апаратного забезпечення, вказують на список необхідних засобів задля стабільної роботи з додатком.

3.6 Аналіз програмної реалізації розробленого методу

Задля кращого розуміння користувачем системи, наведемо коротку покрокову інструкцію з використання застосунку.

1. Надання дозволів. При користуванні програма попросить такі дозволи:

1) доступ до камери, 2) доступ до мікрофона, 3) доступ до галереї.

2. Обліковий запис: створення облікового запису або логін в існуючий, за допомогою Google авторизації.

3. Використання програми: 1) взаємодія з витратами, 2) взаємодія з підписками, 3) редагування профілю, 4) робота з сканером товарних чеків та 5) перегляд статистики.

4. Вимоги: 1) мобільний інтернет або Wi-Fi, 2) смартфон з Android або iOS та 3) обліковий запис Google.

Отже, було коротко описано функціональність застосунку, що доступне користувачеві, та визначено необхідні вимоги для використання та взаємодії з функціями програми.

3.7 Результати досліджень

У роботі використано ML Kit від Firebase для Android та Vision Framework для IOS з можливостями технологій OCR.

Розглянемо безпосередньо етапи розроблення, яких було дотримано протягом впровадження застосунку:

а) планування: визначення вимог системи, план задач;

б) розроблення: написання кодової частини та інтерфейсу користувача, інтеграція сторонніх сервісів;

в) тестування: функціональне тестування програмного забезпечення, виявлення та виправлення несправностей.

Кроки алгоритму оброблення зображень включали такі етапи:

1) завантаження фотозображення чеку – зображення чеку подається як вхідні дані;

2) оброблення зображення – підвищення якості зображення;

3) розпізнавання тексту – виявлення тексту на зображенні;

4) парсинг та інтерпретація розпізнаного тексту – використання методів OCR для зчитування тексту та алгоритму парсингу вихідного тексту в список покупок;

5) збереження та візуалізація даних у список – відображення список із зчитування зображення в формі створення витрат.

Під час тестування було знайдено такі основні проблеми:

а) формат чеку: кожен чек може мати свій формат що впливає на кінцевий результат; для усунення проблеми було додано парсинг тексту, який допомагає повертати тільки потрібну користувачу інформацію;

б) мова чеку: у випадку сканування товарних чеків різних країн, було виявлено що парсинг тексту не достатньо точно видозмінює результат, що впливає на вихідні дані; задля покращення роботи було додано розпізнавання другої мови.

Надалі частина методу може бути покращена розширеною інтеграцією ШІ, що впливало б також на вихідний текст попри мову.

Задля оцінювання продуктивності даного підходу було проведено низку тестувань написаного функціоналу з розпізнавання тексту з різного набору даних. Вимірювання точності розпізнавання тексту проводилося на десяти різноманітних чеках. Графік показує успішність та повноту розпізнавання залежно від якості фотозображення (рисунок 3.15).

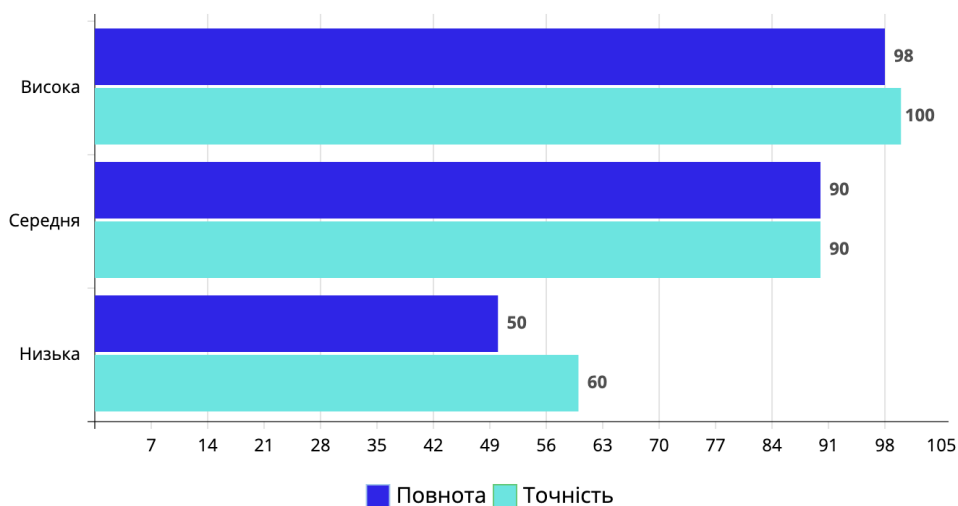


Рисунок 3.15 – Графік точності та повноти розпізнавання фотозображення для високої, середньої та низької якостей зображення

Результат тестування подано нижче.

1. Висока якість зображення:

- вхідні дані: використовувались чіткі, добре освітлені фотозображення з товарних чеків в доброму стані;
- розпізнано близько 98% тексту з максимальною точністю; отже 98% символів на зображенні були правильно ідентифіковані та витягнуті;
- точність обчислювалася за кількістю правильно розпізнаних символів, порівняно з загальною кількістю символів на чеку.

2. Середня якість зображення:

- вхідні дані: використовувались менш чіткі зображення з дещо гіршим освітленням та невеликою кількістю шумів;
- повнота та точність розпізнавання складала близько 90%, що вказує на те, що система правильно ідентифікувала близько 90% символів;
- точність та повнота обчислювалися аналогічно якісним зображенням.

3. Низька якість зображення:

- вхідні дані: використовувалися зображення з високим рівнем шуму, розмитістю, поганим освітленням.
- розпізнано близько 50% тексту, з точністю розпізнавання близько 60%; лише 50% символів були правильно ідентифіковані, а з них 60% відповідали правильному тексту;
- точність обчислювалася через порівняння розпізнавання символів з фактичною кількістю символів на чеку.

Отримані в процесі тестування та вимірювання працездатності методу результати показали досить високу точність та якість обробки зображення. Було підтверджено доцільність використання нейромереж та OCR методу задля вирішення задач даного типу.

Отже, розроблене ПЗ є робочим інструментом для зберігання інформації про витрати, за допомогою автоматизації процесу розпізнавання тексту з

фотозображень товарних чеків, що було підтверджено у результаті тестувань. Це створює передумови для значного покращення якості та функціональності даного програмного забезпечення, що зможе ще більше задовольнити потреби користувачів.

3.8 Висновки до розділу 3

Було описано та розглянуто структуру, функціональність та реалізацію застосунку. Наведено перелік засобів, що використовувалися для розроблення та швидкої обробки даних у режимі реального часу, а також можливість підтримки мобільного застосунку. Подані засоби були обрані з огляду на їхню зручність та ефективність в роботі.

Проведено аналіз архітектури програмного забезпечення, описані основні компоненти і їх взаємодія. Архітектура та діаграми, подані в розділі 3.3, описують функціональність програмного застосунку та вказують на основи для оптимізації та розвитку проєкту.

Розглянуто особливості програмної реалізації, а також апаратні та програмні вимоги для розгортання проєкту. Наведено ключові алгоритми з детальним описом та їх особливостями.

Проведено тестування готового застосунку, яке показало, що програма працює справно та без збоїв. Застосунок визнано ефективним інструментом для зберігання інформації про витрати, завдяки автоматизації процесу розпізнавання тексту з фотозображень товарних чеків. Це створює передумови для значного покращення якості та функціональності програмного забезпечення, що може ще більше задовольнити потреби користувачів.

Загальні висновки

У результаті виконання кваліфікаційної роботи бакалавра було успішно досягнуто мету роботи, а саме підвищено якість розпізнавання тексту на фотозображеннях через розроблення методу розпізнавання тексту та створення на його основі мобільного застосунку.

Розроблений метод розпізнавання тексту використовує технології машинного навчання для оброблення фотозображень, розпізнавання та витягування тексту з них, що дало можливість підвищити якість розпізнавання текстових даних. Мобільний застосунок, який розроблено на основі спроектованого методу розпізнавання тексту, використовує технології OCR, ML Kit від Firebase та Vision Framework від Apple.

Мобільний застосунок забезпечує виконання основних функцій додавання, зберігання в статистику, видалення підписок та витрат тощо. Застосунок також надає користувачам можливість завантаження та знімання зображень товарних чеків, попереднього оброблення зображень для покращення якості, розпізнавання тексту за допомогою OCR, корекції та форматування розпізнаного тексту, збереження та організації даних та візуалізації витрат.

Проведено успішне експериментальне тестування розроблених методу розпізнавання тексту та мобільного застосунку, що підтвердило високу точність та швидкість роботи застосунку. Розроблений метод розпізнавання тексту та його програмна реалізація повністю відповідають поставленим завданням, підвищуючи якість розпізнавання тексту на фотозображеннях товарних чеків.

Насамкінець, подане у цій роботі програмне рішення має високий потенціал для впровадження та подальшого розвитку. Можливі напрями вдосконалення включають розширення підтримуваних форматів товарних чеків та мови парсингу, інтеграцію з іншими фінансовими сервісами, розділення витрат на персональні та комерційні та багато іншого.

Перелік посилань

1. Artificial intelligence in business-to-customer fashion retail: A Literature Review / A. Goti et al. *Mathematics*. 2023. Vol. 11, no. 13. P. 2943. URL: <https://doi.org/10.3390/math11132943>
2. Deep learning in food category recognition / Y. Zhang et al. *Information Fusion*. 2023. Vol. 98. P. 101859. URL: <https://doi.org/10.1016/j.inffus.2023.101859>
3. Бевзюк Р, Лященко О. Система розпізнавання об'єктів для автоматизації складів з використанням згорткової нейронної мережі. *Системи управління, навігації та зв'язку. Збірник наукових праць*. 2023. Т. 2, № 72. С. 68–74. URL: <https://doi.org/10.26906/sunz.2023.2.068>
4. Analysis of deep learning methods in adaptation to the small data problem solving / I. Krak et al. In: Babichev, S., Lytvynenko, V. (eds) *Lecture Notes in Data Engineering, Computational Intelligence, and Decision Making*. ISDMCI–2022. Springer, Cham. 2023. Vol. 149. P. 333–352. URL: https://doi.org/10.1007/978-3-031-16203-9_20
5. Information system for public places and institutions visualization with opportunities of inclusive access and optimal routing / O. Pavlova et al. *Computer systems and information technologies*. 2022. Vol. 1, No. 6. P. 62–68. DOI: <https://doi.org/10.31891/CSIT-2022-1-8>
6. Explainable Deep Learning: A visual analytics approach with transition matrices / P. Radiuk et al. *Mathematics*. 2024. Vol. 12, no. 7. P. 1024. URL: <https://doi.org/10.3390/math12071024>
7. Intelligent integrated system for fruit detection using multi-UAV imaging and deep learning / O. Melnychenko et al. *Sensors*. 2024. Vol. 24, no. 6. P. 1913. URL: <https://doi.org/10.3390/s24061913>
8. Radiuk P.M. Application of a genetic algorithm to search for the optimal convolutional neural network architecture with weight distribution. *Herald of Khmelnytskyi National University. Technical sciences*. 2020. Vol. 281, No. 1. P. 7–11. DOI: <https://doi.org/10.31891/2307-5732-2020-281-1-7-11>

9. Niu Z., Zhong G., Yu H. A review on the attention mechanism of deep learning. *Neurocomputing*. 2021. Vol. 452. P. 48–62. URL: <https://doi.org/10.1016/j.neucom.2021.03.091>
10. Van Houdt G., Mosquera C., Nápoles G. A review on the long short-term memory model. *Artificial Intelligence Review*. 2020. Vol. 53, no. 8. P. 5929–5955. URL: <https://doi.org/10.1007/s10462-020-09838-1>
11. Exploration in deep reinforcement learning: A survey / P. Ladosz et al. *Information Fusion*. 2022. Vol. 85. P. 1–22. URL: <https://doi.org/10.1016/j.inffus.2022.03.003>
12. TrOCR: Transformer-Based Optical Character Recognition with Pre-trained Models / M. Li et al. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2023. Vol. 37, no. 11. P. 13094–13102. URL: <https://doi.org/10.1609/aaai.v37i11.26538>
13. Lin C.-J., Liu Y.-C., Lee C.-L. Automatic receipt recognition system based on artificial intelligence technology. *Applied Sciences*. 2022. Vol. 12, no. 2. P. 853. URL: <https://doi.org/10.3390/app12020853>
14. CamScanner. CamScanner: text and image scanning and recognition, PDF to Word, document format conversion, online editor. URL: <https://www.camscanner.com/>
15. CamScanner: програма для сканування документів за допомогою хмарного сервісу. URL: <https://androidayuda.com/uk/застосування/рекомендується/CamScanner/>
16. Adobe Scan. URL: <https://www.adobe.com/ua/acrobat/mobile/scanner-app.html>
17. Про Adobe Scan. URL: <https://helpx.adobe.com/ua/mobile-apps/help/adobe-scan-faq.html>
18. Office Lens for Windows – Microsoft Support. Microsoft Support. URL: <https://support.microsoft.com/en-gb/office/office-lens-for-windows-577ec09d-8da2-4029-8bb7-12f8114f472a>

19. Які переваги та недоліки має Office Lens? URL: <https://trucoteca.com/uk/que-ventajas-y-desventajas-tiene-office-lens/>

20. Методи розпізнавання зображень для системи персонального обліку витрат. URL: <https://openarchive.nure.ua/entities/publication/96f172d3-4995-4dd1-83a5-02e462cceb9b>

21. ML Kit | Google for Developers. Google for Developers. URL: <https://developers.google.com/ml-kit>

22. Vision | Apple Developer Documentation. Apple Developer Documentation. URL: <https://developer.apple.com/documentation/vision>

23. Recognizing Text in Images | Apple Developer Documentation. Apple Developer Documentation. URL: https://developer.apple.com/documentation/vision/recognizing_text_in_images

24. Text Recognition in Vision Framework – Apple Developer. Apple Developer. URL: <https://developer.apple.com/videos/play/wwdc2019/234/>

25. React Native Firebase | React Native Firebase. React Native Firebase | React Native Firebase. URL: <https://rnfirebase.io/>

26. Moment.js | Home. Moment.js | Home. URL: <https://momentjs.com/>

27. React Native Firebase | React Native Firebase. React Native Firebase | React Native Firebase. URL: <https://rnfirebase.io/>

28. react-native-text-recognition. npm. URL: <https://www.npmjs.com/package/react-native-text-recognition>

29. @react-native-google-signin/google-signin. npm. URL: <https://www.npmjs.com/package/@react-native-google-signin/google-signin>

30. Node.js – About Node.js. Node.js – Run JavaScript Everywhere. URL: <https://nodejs.org/en/about> (дата звернення: 16.05.2024).

31. Firebase | Google’s Mobile and Web App Development Platform. Firebase. URL: <https://firebase.google.com/>

32. Writing Markup with JSX – React. React. URL: <https://react.dev/learn/writing-markup-with-jsx#jsx-putting-markup-into-javascript>

33. Introduction React Native. React Native · Learn once, write anywhere.

URL: <https://reactnative.dev/docs/getting-started>

34. Microsoft. Visual Studio Code – Code Editing. Redefined. Redefined.

URL: <https://code.visualstudio.com/>

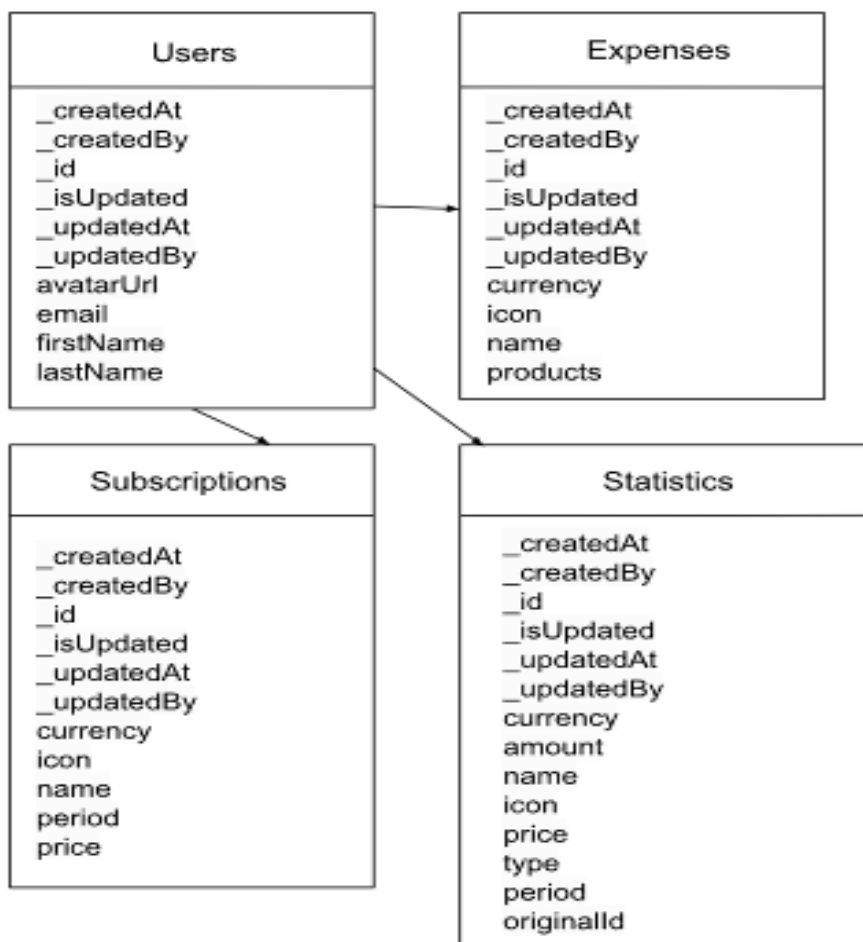
35. Xcode | Apple Developer Documentation. Apple Developer

Documentation. URL: <https://developer.apple.com/documentation/xcode>

ДОДАТКИ

Додаток А

Структура бази даних програмного забезпечення



Додаток Б

Правила бази даних

```
rules_version = '1';

service cloud.firestore {
  match /databases/{database}/documents {
    function signedIn() {
      return request.auth.uid != null;
    }

    match /users/{userId} {
      allow read: if signedIn();
      allow create: if true;
      allow update: if signedIn();
    }

    match /subscriptions/{subscriptionId} {
      allow read, create, update, delete: if signedIn();
    }

    match /expenses/{expensesId} {
      allow read, create, update, delete: if signedIn();
    }

    match /statistics/{statisticId} {
      allow read, create, update, delete: if signedIn();
    }
  }
}
```

Додаток В

Лістинг програмного коду

Лістинг BillScanner.screen.js

```

import React, {useState} from 'react'
import {
  View,
  StyleSheet,
  TouchableOpacity,
  SafeAreaView,
  Dimensions,
  ScrollView,
} from 'react-native'
import {EXPENSE_CREATE} from '../../_constants_'
import {RNCamera} from 'react-native-camera'
import FastImage from '@qonsoll/react-native-fast-image'
import {Spinner, Text} from 'native-base'
import {launchImageLibrary} from 'react-native-image-picker'
import {ArrowsArrowNarrowLeft, Close, Gallery} from '../../_constants_/assets'
import {useTextRecognition} from '../../_hooks_'

const TextRecognise = ({selectedImage, setSelectedImage}) => {
  const {recognizedText, loading, takePicture, cameraRef, navigation} =
    useTextRecognition(selectedImage)
  const receiptData = recognizedText?.length ? parseReceipt(recognizedText) : []

  return (
    <SafeAreaView style={styles.container}>
      <View
        style={{
          flexDirection: 'row',
          justifyContent: 'space-between',
          alignItems: 'center',
          marginHorizontal: 24,
          marginVertical: 14,
        }}>
        <TouchableOpacity onPress={() => navigation.goBack()} activeOpacity={0}>
          <View>
            <FastImage
              source={ArrowsArrowNarrowLeft}
              style={{width: 24, height: 24}}
              tintColor={'white'}
            />
          </View>
        </TouchableOpacity>
        <TouchableOpacity
          onPress={() => {

```

```

        const params = {initialData: receiptData}
        navigation.navigate(EXPENSE_CREATE, params)
    })
    activeOpacity={0}>
    <Text color="white" variant="h5">
        Done
    </Text>
</TouchableOpacity>
</View>

{loading && (
    <View
        style={{
            width: '100%',
            height: Dimensions.get('window').height,
            backgroundColor: 'rgba(0,0,0,0.5)',
            position: 'absolute',
            zIndex: 1,
            justifyContent: 'center',
            alignItems: 'center',
        }}>
        <Spinner size="md" color="black" />
    </View>
)}

{recognizedText ? (
    <ScrollView
        stickyHeaderIndices={[0]}
        keyboardShouldPersistTaps="always"
        showsVerticalScrollIndicator={false}
        style={{
            width: '100%',
            height: Dimensions.get('window').height - 220,
            backgroundColor: 'black',
            padding: 16,
            paddingTop: 0,
        }}
        contentContainerStyle={styles.contentContainerStyle}>
    {receiptData.length > 0 && (
        <View>
            <View
                style={{
                    display: 'flex',
                    flexDirection: 'row',
                    justifyContent: 'space-between',
                    padding: 10,
                    backgroundColor: 'black',
                }}>
            </View>
        </View>
    )}

```

```

        <Text color="white" variant="body1">
            Product
        </Text>
    </View>
    <View>
        <Text color="white" variant="body1">
            Price
        </Text>
    </View>
</View>
</View>
)}
<View
    style={{
        paddingBottom: 16,
    }}>
    {receiptData.length > 0 ? (
        receiptData.map((item, index) => (
            <View
                key={index}
                style={{
                    justifyContent: 'space-between',
                    flexDirection: 'row',
                    padding: 10,
                }}>
                <Text color="white" variant="h5">
                    {item.name}
                </Text>
                <Text color="white" variant="h5">
                    {item.price}
                </Text>
            </View>
        ))
    ) : (
        <Text color="white" variant="h5" style={{padding: 10}}>
            No items found in the receipt
        </Text>
    )}
</View>
</ScrollView>
) : (
    <RNCamera
        ref={ref => {
            cameraRef = ref
        }}
        style={styles.preview}
        type={RNCamera.Constants.Type.back}
        flashMode={RNCamera.Constants.FlashMode.off}
    >

```

```

        captureAudio={false}
    />
  )}

<View
  style={{
    flex: 0,
    flexDirection: 'row',
    justifyContent: recognizedText ? 'center' : 'space-between',
    alignItems: 'center',
    margin: 24,
  }}>
  {recognizedText ? (
    <TouchableOpacity
      onPress={() => {
        setRecognizedText(null)
        setSelectedImage(null)
      }}
      activeOpacity={0}>
      <FastImage
        source={Close}
        style={{width: 32, height: 32}}
        tintColor={'white'}
      />
    </TouchableOpacity>
  ) : (
    <>
      <ImageUploader onImageSelected={setSelectedImage} />

      <TouchableOpacity
        onPress={takePicture}
        style={{
          height: 60,
          width: 60,
          borderRadius: 50,
          backgroundColor: 'white',
        }}
      />
      <View
        style={{
          height: 40,
          width: 40,
        }}
      />
    </>
  )}
</View>
</SafeAreaView>

```

```

)
}

const ImageUploader = ({onImageSelected}) => {
  const handleImageUpload = () => {
    launchImageLibrary({mediaType: 'photo'}, response => {
      if (response.didCancel) {
        console.log('User cancelled image picker')
      } else if (response.error) {
        console.log('ImagePicker Error: ', response.error)
      } else {
        const uri = response.assets[0].uri
        onImageSelected(uri)
      }
    })
  }

  return (
    <TouchableOpacity onPress={handleImageUpload}>
      <FastImage source={Gallery} style={{width: 40, height: 40}} />
    </TouchableOpacity>
  )
}

const BillScanner = () => {
  const [selectedImage, setSelectedImage] = useState(null)

  return (
    <TextRecognise
      selectedImage={selectedImage}
      setSelectedImage={setSelectedImage}
    />
  )
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    flexDirection: 'column',
    backgroundColor: 'black',
    paddingVertical: 10,
  },
  title: {
    fontSize: 18,
    fontWeight: 'bold',
  },
  preview: {
    flex: 1,

```

```

    justifyContent: 'flex-end',
    alignItems: 'center',
  },
})

```

```
export default BillScanner
```

Лістинг useTextRecognition.js

```

import { useState, useEffect, useRef } from 'react';
import { useNavigation } from '@react-navigation/native';
import TextRecognition from 'react-native-text-recognition'

const useTextRecognition = (selectedImage) => {
  const [recognizedText, setRecognizedText] = useState(null);
  const [loading, setLoading] = useState(false);
  const navigation = useNavigation();
  let cameraRef = useRef(null);

  useEffect(() => {
    const recognizeText = async () => {
      if (selectedImage) {
        setLoading(true);
        try {
          const text = await TextRecognition.recognize(selectedImage);
          setRecognizedText(text);
        } catch (error) {
          console.error('Error recognizing text:', error);
        }
        setLoading(false);
      }
    };
    recognizeText();
  }, [selectedImage]);

  const takePicture = async () => {
    if (cameraRef.current) {
      setLoading(true);
      const options = { quality: 0.5, base64: true };
      const data = await cameraRef.current.takePictureAsync(options);

      try {
        const text = await TextRecognition.recognize(data?.uri);
        setRecognizedText(text.length ? text : []);
      } catch (error) {
        console.error('Error recognizing text:', error);
      }
    }
  };
}



```

```
    }  
    setLoading(false);  
  }  
};  
  
return {  
  recognizedText,  
  loading,  
  takePicture,  
  cameraRef,  
  navigation  
};  
};  
  
export default useTextRecognition;
```

Додаток Г

Презентаційний матеріал

1



Кваліфікаційна робота бакалавра

Метод розпізнавання тексту за фотозображеннями нейромережевими засобами

Виконала:
студентка 4 курсу, групи
КН-20-1
Шаповалова Анастасія Олегівна

Керівник:
Доктор філософії, старший викладач
кафедри комп'ютерних наук
Радюк Павло Михайлович

2

Актуальність

На сьогодні галузь роздрібної торгівлі спонукає продавців генерувати масу товарних чеків.

Оброблення та аналіз товарних чеків є важливим для кінцевих користувачів, оскільки чеки часто містять багато цінної інформації, яку, проте, важко вилучати та обробляти вручну. Цей ручний процес займає багато часу та може призвести до помилок, що призводить до втрати даних і збільшення адміністративного навантаження.

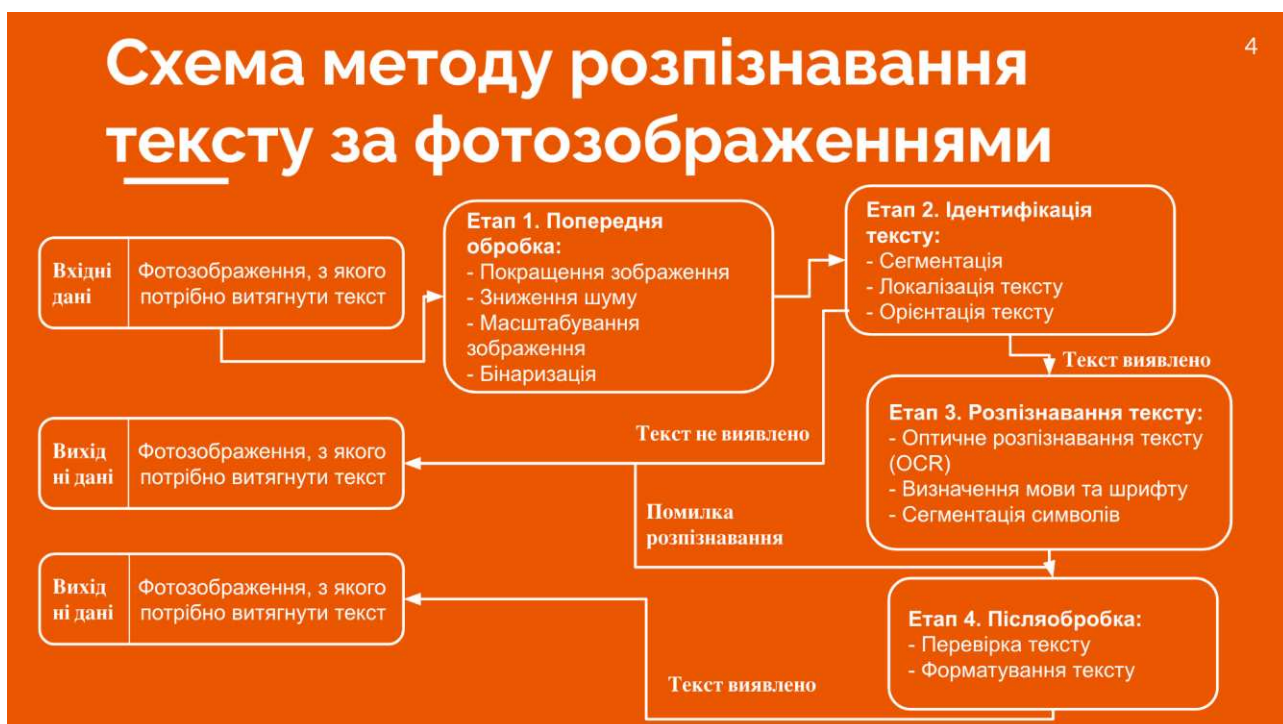
Для вирішення цієї проблеми важливо розробити метод розпізнавання тексту на фотозображеннях із використанням нейромережових технологій та розробити відповідний мобільний застосунок. Це рішення дасть можливість покращити процес розпізнавання, підвищивши її точність, що надалі спростить оброблення та використання товарних чеків.

Мета і задачі роботи

Мета кваліфікаційної роботи бакалавра – підвищення якості розпізнавання тексту на фотозображеннях нейромережевими засобами.

Завдання кваліфікаційної роботи бакалавра:

- обґрунтувати покращення розпізнавання тексту на фотозображеннях чеків за допомогою неймереж;
- розробити відповідний метод і архітектуру застосунку;
- створити модуль для зберігання та редагування тексту;
- реалізувати застосунок з інтуїтивним інтерфейсом;
- протестувати метод і застосунок;
- оцінити точність і швидкість OCR;
- продемонструвати покращення якості розпізнавання.



5

Діаграма активностей процесу розпізнавання тексту



Інформаційна структура системи

6

Підсистема роботи з відомостями бази даних

- Робота з даними підписок користувачів(підписка, валюта, вартість)
- Робота з даними витрат користувачів(список витрат - продукт та ціна, назва, валюта)

- Робота з користувачами(ім'я, прізвище, емейл)
- Робота з статистикою(збір статистики з кожного створеного документа)

Підсистема користувацького створення

- Створення акаунту - логін через Google
- Створення підписки

- Створення списку витрат, вручну користувачем або автоматизоване зчитування тексту з чеків.

Підсистема авторизації

- Авторизація через Google(реєстрація та логін)

Підсистема зчитування тексту з фотозображень

- Вхідні дані - фотозображення
- Попередня обробка вхідних даних
- Ідентифікація тексту

- Розпізнавання тексту
- Перевірка та корекція вихідного тексту
- Вивід результату

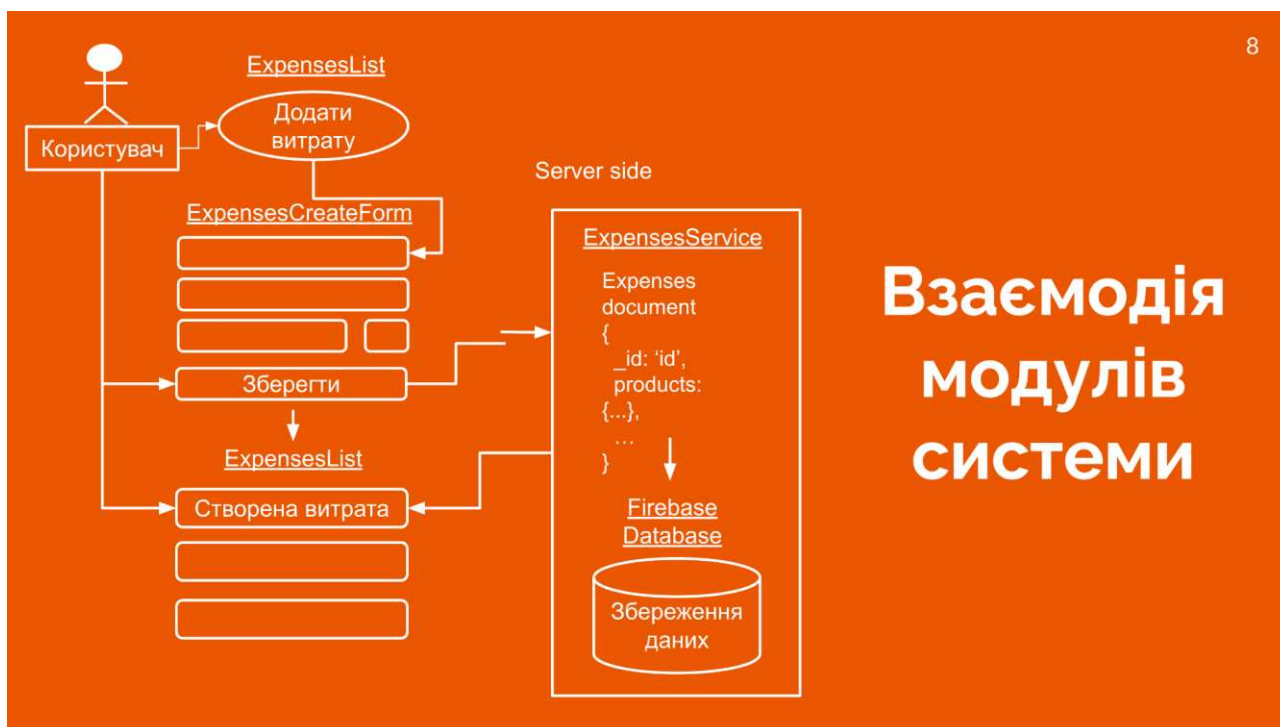
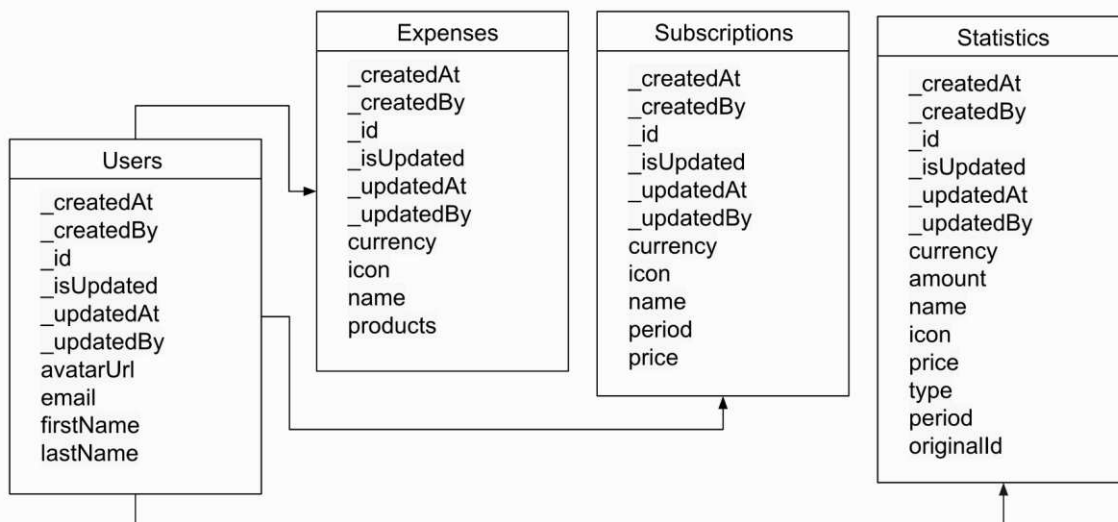
Підсистема статистики

- Зберігання даних статистики(підписок та витрат)
- Калькуляція даних для статистики

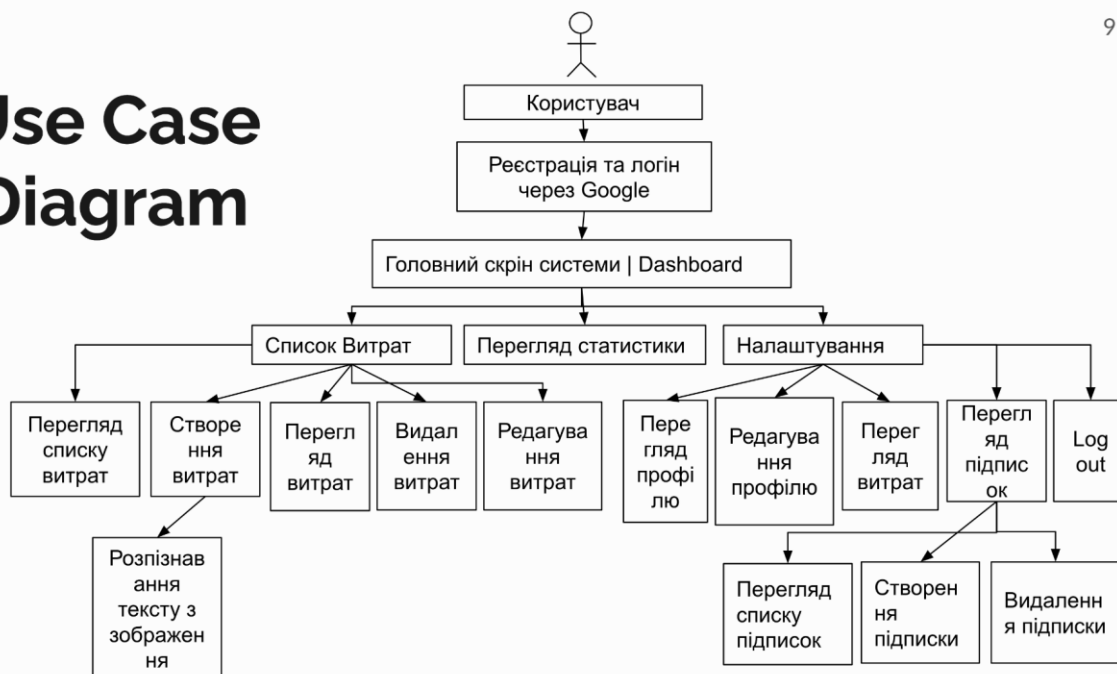
- Вивід витрат за конкретний місяць(у вигляді списків та графіків)
- Калькуляція суми витрат у гривнях

БД

Даталогічна модель бази даних

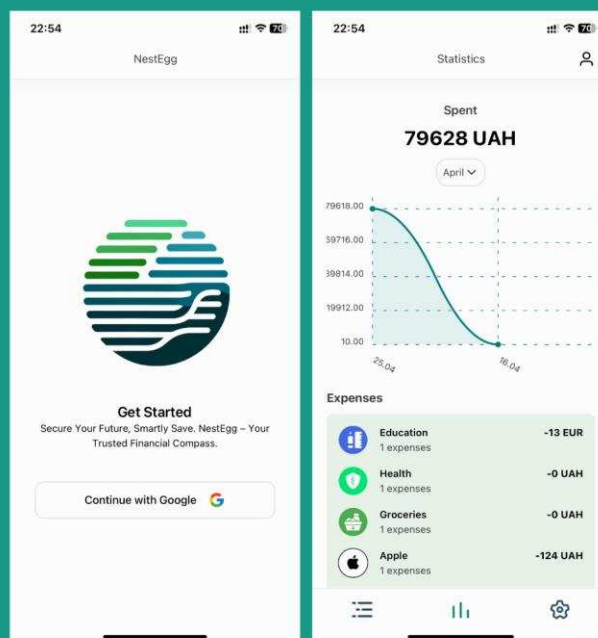


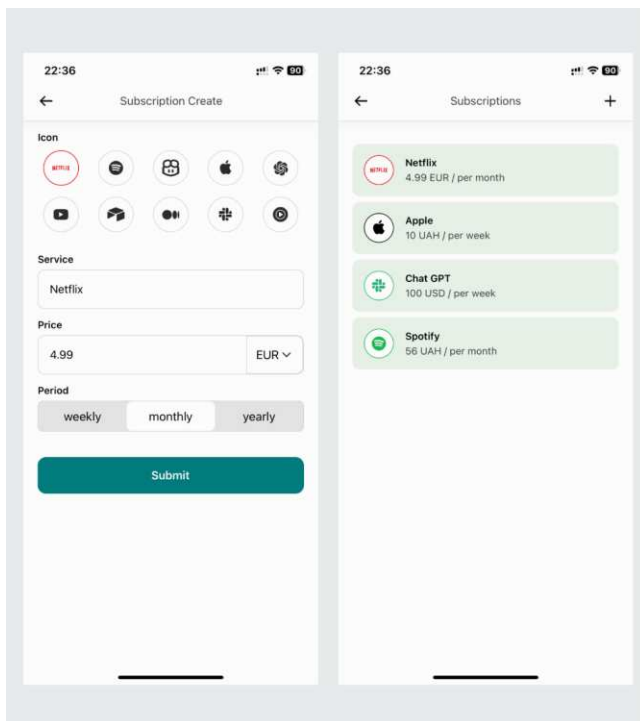
Use Case Diagram



10

Програмна реалізація



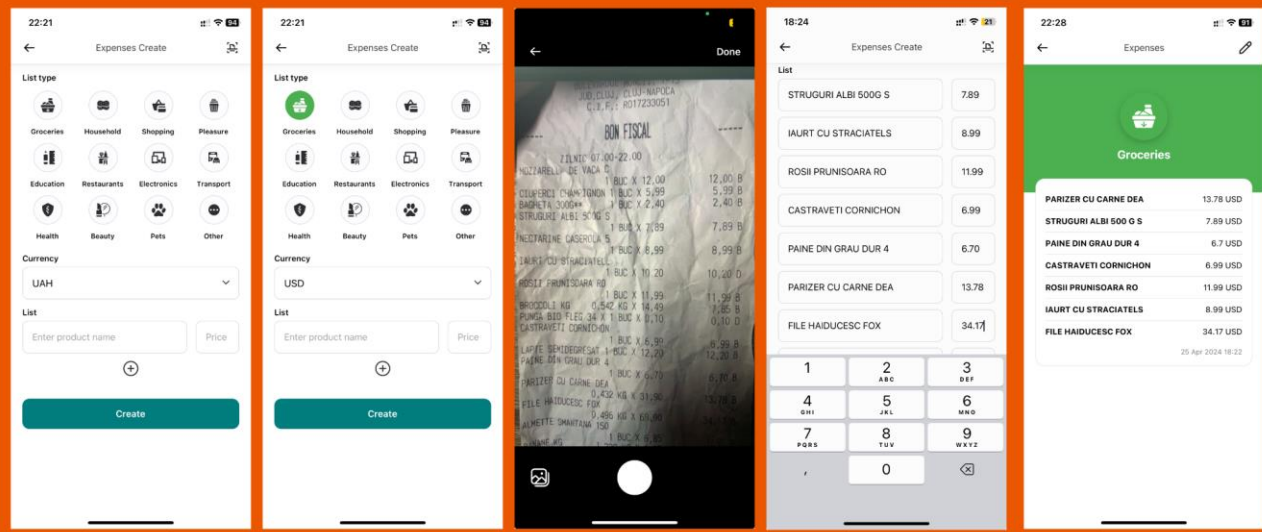


11

Програмна реалізація

Програмна реалізація розпізнавання тексту з чеків

12



13

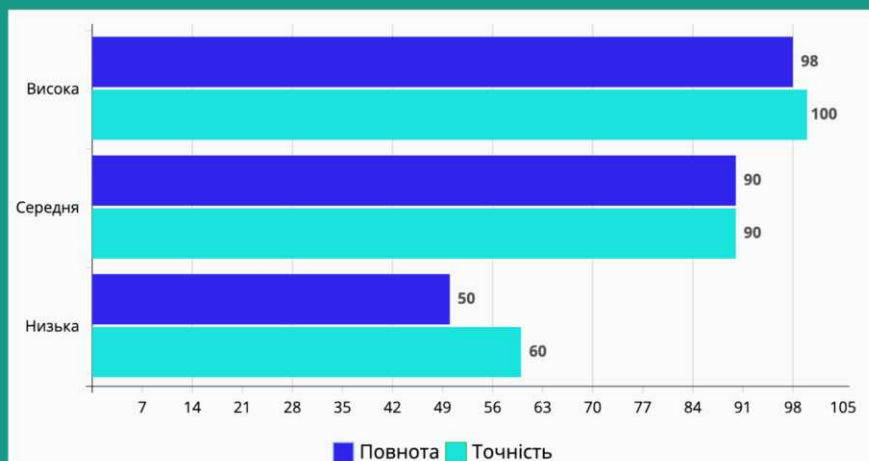
Тестування застосунку

У результаті проведеного тестування програмного продукту було успішно перевірено основні функціональні можливості, включаючи реєстрацію та логін користувачів, створення, редагування, видалення витрат та підписок, а також перегляд статистики. Усі тест-кейси були виконані успішно, що підтверджує коректність роботи програми без виявлених багів. Програма демонструє високу якість та зручність користування, відповідаючи всім вимогам та очікуванням.

Тест-кейс ID: TS0003	Пріоритет: 2	Створено: 21.05.2024, Шаповалова А.О.
Назва: Перевірка функціоналу — редагування “Expenses”.		
Кроки	Очікуваний результат	
<ol style="list-style-type: none"> Запустити програму. Пройти реєстрацію/логін в застосунку. Перейти на сторінку витрати. Натиснути кнопку редагування. Після потрапляння на форму змінити необхідні дані та натиснути на кнопку підтвердження. 	Користувач буде перенаправлений на сторінку зі витратою, де будуть відображатися оновлені дані.	
Результат виконання тест-кейсу: пройдено успішно		

14

Тестування розпізнавання тексту з чеків



Висновок

Метою даної роботи було підвищення якості розпізнавання тексту на фотозображеннях нейромережевими засобами.

Мету роботи досягнуто у вигляді розробленого методу розпізнавання тексту на фотозображеннях нейромережевими засобами за допомогою ML Kit та Vision Framework. Виконано програмну реалізацію методу розпізнавання у вигляді мобільного застосунку з використанням React Native та Firebase.

Розроблений метод розпізнавання тексту та його програмна реалізація повністю відповідають поставленим завданням, підвищуючи якість розпізнавання тексту на фотозображеннях товарних чеків.

Програмне рішення, представлене в цій роботі, має високий потенціал для впровадження та розвитку, з можливостями вдосконалення, включаючи розширення підтримуваних форматів чеків, мови парсингу, інтеграцію з іншими фінансовими сервісами та розділення витрат на персональні та комерційні.

Дякую за увагу!

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 4.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 11%**

ID: 130515 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод розпізнавання тексту на фотозображеннях нейромережевими засобами Додано в БД: 2024-06-14 Автора: Анастасія ШАПОВАЛОВА Керівники: Павло РАДЮК Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	72962	1145	4417 (6%)	68 (6%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:
Кафедра КН

Дата перевірки:
14.06.2024 13:09:15 EEST

Дата звіту:
14.06.2024 13:46:55 EEST

ID перевірки:
1016360158

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005671

Назва документа: КН-20-1_Шаповалова_ЗАПИСКА

Кількість сторінок: 82 Кількість слів: 12312 Кількість символів: 98776 Розмір файлу: 2.04 MB ID файлу: 1016164898

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

8.76%
Схожість

Найбільша схожість: 2.97% з джерелом з Бібліотеки (ID файлу: 1016162161)

6.77% Джерела з Інтернету 708 Сторінка 84

5.67% Джерела з Бібліотеки 139 Сторінка 88

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 2

Підозріле форматування 13 сторінок

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод розпізнавання тексту на фотозображеннях нейромережевими засобами

Автор: студентка групи КН-20-1 Шаповалова Анастасія

Спеціальність: 122 Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: док. філ., ст. викл. Радюк П.М.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<i>відповідає</i>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, що виявлені в роботі Шаповалової А.О., не є плагіатом, оскільки: запозичення розміщені в розділі огляду наявних підходів, не описують безпосередньо авторську роботу й не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять цитування джерел у переліку посилань; поміж запозичень є загальновідомі терміни та скорочення.

Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:

– за системою Anti-Plagiarism: 4.0 %;

– за системою Unicheck: 8.8 %.

Отже, запозичення є допустимими та належать до описаних вище й адресуються до першоджерел, що, з урахуванням наведених обґрунтувань, свідчить на користь кваліфікаційної роботи.

Керівник роботи



Павло РАДЮК

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ МОН УКРАЇНИ



Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студентки КН-20-1 Шаповалової Анастасії Олегівни
за темою Метод розпізнавання тексту на фотозображеннях нейромережевими засобами

1. Актуальність теми

На сьогодні онлайн та офлайн покупки є неодмінною частиною нашого життя. Тому вміння стежити за витратами є вкрай важливим. Раніше для цього вели паперові записи та зберігали чеки, що займало багато часу й місця. З розвитком інтернет-шопінгу та штучного інтелекту, зокрема нейромереж, автоматизація контролю витрат стає ще більш актуальною. Сканування чеків і витяг інформації дають можливість спростити відстеження фінансів і підвищити фінансову грамотність. Впровадження таких технологій до оброблення товарних чеків допоможе ефективно планувати бюджет, зменшуючи витрати часу та ресурсів, що є на сьогодні особливо важливим.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Відповідно до стандарту бакалавра вищої освіти України спеціальності 122 Комп'ютерні науки, описом предметної галузі, об'єктом та предметом вивчення є математичні, інформаційні та імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи й технології отримання, зберігання, обробки, передачі та використання інформації. Метою поданої роботи є підвищення якості розпізнавання тексту на фотозображеннях нейромережевими засобами. Мету роботи досягнуто внаслідок використання методів, способів та алгоритмів розв'язання теоретичних і прикладних задач, що виникають у процесі проєктування та розроблення інформаційних технологій. Отже, результати виконання кваліфікаційної роботи відповідають стандарту бакалавра спеціальності 122 Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

Під час виконання кваліфікаційної роботи студентка Шаповалова Анастасія Олегівна проявила себе кваліфікованою фахівчиною та дисциплінованою студенткою, вчасно виконуючи поставлені перед нею завдання. Як у процесі розроблення прикладного програмного забезпечення, так і під час написання пояснювальної записки студентка

успішно засвоїла компетентності та результати навчання. Шаповалова А.О. опанувала професійні навички та компетентності, що відповідають виконанню освітньо-професійної програми рівня вищої освіти «Бакалавр» за спеціальністю 122 Комп'ютерні науки.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Результати роботи та їхня обґрунтована практична значущість одержані та обумовлені студентом особисто, як наслідок виконання нею усіх поставлених завдань.

5. Ступінь оволодіння методами дослідження

У процесі розроблення методу розпізнавання тексту на фотозображеннях нейромережевими засобами та створення на його основі мобільного застосунку студентка Шаповалова А.О. продемонструвала достатній рівень компетентностей та володіння необхідними інструментами й обладнанням, методами, методиками та технологіями галузі 12 Інформаційні технології.

6. Повнота та якість розкриття теми роботи

Тема роботи повністю обґрунтована й розкрита; актуальність предметної галузі та відомі дослідження щодо обраної тематики проаналізовані достатньо. Усі поставлені завдання в роботі виконані повністю. Розроблене програмне забезпечення для валідування та верифікування спроектованого методу розпізнавання тексту на фотозображеннях відповідає технічним вимогам спеціальності 122 Комп'ютерні науки.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Матеріал кваліфікаційної роботи Шаповалової А.О. подано логічно, послідовно, аргументовано та є таким, що відповідає поставленій меті. Мова та стиль викладення роботи відповідають стандартам, що забезпечує доступність сприймання матеріалу і відповідає вимогам до сучасних кваліфікаційних робіт.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Спроектований у роботі метод розпізнавання тексту на фотозображеннях товарних чеків може бути використаний у різних галузях, зокрема у фінансовому менеджменті для автоматизації та покращення процесу обліку витрат.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

З огляду на рівень виконання та забезпечення всіх необхідних вимог, вважаю, що кваліфікаційна робота Шаповалової Анастасії Олегівни може бути допущена до захисту. Рекомендована оцінка – «добре».

Керівник _____



док. філ., ст. викл. каф. КН Павло РАДЮК



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студентки КН-20-1 Шаповалової Анастасії Олегівни
за темою: Метод розпізнавання тексту на фотозображеннях нейромережевими засобами

1. Актуальність обраної теми

Автоматизація розпізнавання тексту на фотозображеннях чеків покупок є актуальною темою в сучасному суспільстві, де швидкість та точність оброблення інформації є критично важливими. Використання нейромереж для цього завдання дає можливість покращити цей процес та мінімізувати людські помилки. Ця технологія має великий потенціал у фінансовому управлінні, де важливо швидко та точно обробляти великі обсяги текстових даних.

2. Повнота розкриття мети та завдань роботи

У процесі виконання кваліфікаційної роботи бакалавра були ретельно розкриті мета та завдання. Авторкою проведено глибоке дослідження предметної області, спроектовано метод розпізнавання тексту на фотозображеннях чеків покупок та реалізовано на його основі відповідне програмне забезпечення. У підсумку, мета та завдання поданої до захисту роботи є розкритими повністю.

3. Зміст кожного розділу роботи

У першому розділі кваліфікаційної роботи проведено дослідження сучасних методів та підходів до розпізнавання тексту на фотозображеннях. Другий розділ присвячений проектуванню методу розпізнавання тексту за допомогою нейромереж та проектуванню мобільного застосунку загалом. У третьому розділі наведено та описано програмну реалізацію застосунку для керування витратами з функцією розпізнавання тексту на фотозображеннях. Насамкінець оформлено висновки до виконаних у роботі завдань.

4. Оцінка розробленого програмного забезпечення, його практична цінність

Спроектований метод розпізнавання тексту на фотозображеннях чеків покупок та його програмна реалізація можуть бути застосовані до автоматизації обліку витрат та покращення аналізу покупок клієнтів. Запропонований авторкою мобільний застосунок дає змогу швидко перетворювати текст з фото чеків у редагований електронний формат та забезпечує підвищення якості розпізнавання тексту на фотозображеннях.

5. Якість оформлення кваліфікаційної роботи бакалавра

Робота відповідає вимогам науково-технічного стилю написання, має чітку структуру та включає всі необхідні розділи, як от, перелік скорочень, вступ, огляд літератури, методологія, результати експериментів, висновки та перелік посилань. Кожен розділ повністю виконує своє функціональне призначення і має чіткий зміст. Використані джерела належно процитовані й включені до списку використаної літератури, відповідно до наукових стандартів цитування. Використана авторкою структура та оформлення сприяють легкому сприйняттю та розумінню матеріалу.

6. Недоліки кваліфікаційної роботи бакалавра

Робота також містить кілька недоліків. Зокрема, у роботі бракує опису оцінювання спроектованого методу за статистичними показниками точності, влучності, повноти тощо. Під час проєктування методу розпізнавання тексту авторка покладалася на технологію Vision Framework від компанії Apple, що робить її рішення щодо розпізнавання тексту важко відтворюваним та орієнтованим виключно на одного корпоративного вендора. У тексті трапляються скорочення, які не наведено у переліку скорочень.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

З огляду на рівень виконання та забезпечення всіх необхідних вимог вважаю, що подана кваліфікаційна робота бакалавра може бути допущена до захисту. Рекомендована оцінка – «добре».

Рецензент

доц. кер. ПТЗ Ямане О.М. 