МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ САДІВНИЦТВА
Факультет економіки і підприємництва
Кафедра інформаційних технологій

Допущено до захисту
"_____" _____ 2025 р.
Зав. кафедри, к.т.н., доцент
_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ АНАЛІЗУ ВИТРАТ КОРИСТУВАЧА З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Виконав – студент 2 курсу, групи 22к-кн
освітньої програми "Комп’ютерні науки"
спеціальності 122 Комп’ютерні науки
Підручний Денис Максимович

Науковий керівник – к.т.н., доцент
Ліщук Роман Ігорович

Умань – 2025

Зміст

ВСТУП 3

РОЗДІЛ 1 ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ 5

1.1. Аналіз стану мобільних застосунків з елементами штучного інтелекту для аналізу особистих фінансів 5

1.2. Огляд наявних аналогів 7

1.3. Постановка задачі 12

РОЗДІЛ 2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ МОБІЛЬНОГО ЗАСТОСУНКУ 15

2.1. Аналіз і обґрунтування вибору засобів для реалізації застосунку . 15

2.2. Розробка методу управління даними користувача 17

2.3. Розробка моделі системи та методу аналізу особистих витрат 19

2.4. Розробка модуля для управління даними користувача 24

2.5. Розробка модуля для аналізу та отримання даних від моделі штучного інтелекту 27

РОЗДІЛ 3 РЕАЛІЗАЦІЯ МОДУЛІВ МОБІЛЬНОГО ЗАСТОСУНКУ 29

3.1. Реалізація графічного модуля застосунку 29

3.2. Реалізація штучного інтелекту для аналізу витрат 33

3.3. Тестування мобільного застосунку для аналізу індивідуальних витрат на основі штучного інтелекту 36

ВИСНОВКИ 42

СПИСОК ЛІТЕРАТУРИ 44

ДОДАТКИ 48

ВСТУП

У сучасному світі, де фінансова грамотність набуває дедалі більшого значення, особисте управління бюджетом стало невід'ємною частиною повсякденного життя. Швидкий розвиток цифрових технологій, зокрема мобільних пристроїв та програмного забезпечення, відкриває широкі можливості для створення зручних і ефективних інструментів контролю витрат. Однак більшість наявних застосунків для ведення особистого бюджету є обмеженими у функціональності, потребують ручного введення даних і не враховують індивідуальні фінансові звички користувачів. У зв’язку з цим постає потреба у впровадженні інноваційних технологій, зокрема методів штучного інтелекту, які здатні автоматизувати аналіз витрат, прогнозувати фінансову поведінку та надавати персоналізовані рекомендації.

Застосування штучного інтелекту у сфері особистих фінансів дозволяє створювати більш гнучкі, адаптивні та розумні системи. Штучний інтелект може виявляти приховані закономірності у витратах користувача, прогнозувати майбутні фінансові потреби та оптимізувати бюджетування. Особливої ваги набуває розробка мобільних рішень, які поєднують простоту використання з потужним функціоналом аналізу даних у реальному часі. Таким чином, дослідження, спрямоване на створення мобільного застосунку для аналізу витрат із використанням ШІ, є актуальним, інноваційним і має значний практичний потенціал.

Мета дослідження полягає в розробці мобільного застосунку, що дозволяє користувачам ефективно аналізувати власні витрати шляхом автоматизації збору, обробки та інтерпретації фінансових даних із застосуванням методів штучного інтелекту, з метою надання рекомендацій для раціонального управління бюджетом.

Об’єкт дослідження – процес цифрової підтримки особистого фінансового менеджменту користувача за допомогою мобільних програмних засобів.

Предмет дослідження – методи та технології застосування штучного інтелекту для аналізу, прогнозування та оптимізації особистих витрат у межах мобільного застосунку.

Методи дослідження, використані в роботі, включають: аналітичні методи – аналіз наукових публікацій, сучасних мобільних застосунків, тенденцій розвитку технологій штучного інтелекту та фінансового планування; емпіричні методи – збір та аналіз вимог користувачів, вивчення сценаріїв використання, анкетування та опитування потенційних споживачів; методи проєктування архітектури мобільного застосунку, реалізація інтерфейсу користувача, інтеграція алгоритмів машинного навчання; експериментальні методи – тестування роботи застосунку, порівняння ефективності різних моделей аналізу даних, оцінка точності рекомендацій.

Результати дослідження можуть бути корисними для подальшої розробки персоналізованих цифрових сервісів у сфері фінансової грамотності, а також для вдосконалення мобільних інтерфейсів користувача з підтримкою штучного інтелекту.

РОЗДІЛ 1 ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1. Аналіз стану мобільних застосунків з елементами штучного інтелекту для аналізу особистих фінансів

У сучасних умовах диджиталізації фінансова грамотність та здатність ефективно управляти власними коштами стають критично важливими навичками для широкого кола користувачів. З огляду на зростаючу складність фінансових операцій, а також поширення безготівкових розрахунків і мультибанкінгу, виникає потреба в інструментах, що дозволяють автоматизувати облік, аналіз і прогнозування особистих витрат. У зв’язку з цим на ринку активно з’являються мобільні застосунки, які пропонують функціональність для ведення бюджету та управління витратами. Особливої уваги заслуговують ті рішення, які інтегрують елементи штучного інтелекту (ШІ) з метою поглибленого аналізу даних і надання персоналізованих фінансових рекомендацій.

Загалом, мобільні застосунки для управління фінансами поділяються на дві основні категорії: ті, що функціонують переважно як цифрові реєстри витрат, і ті, що реалізують розширені функції аналітики, прогностичного моделювання та рекомендацій за допомогою ШІ. Останні належать до інноваційного сегменту фінтех-індустрії, який швидко розвивається завдяки досягненням у сфері машинного навчання, обробки природної мови, кластеризації та поведінкової аналітики.

Серед найбільш відомих рішень на ринку варто відзначити Mint, PocketGuard, Emma, Cleo та інші подібні застосунки. Наприклад, Mint (розроблений компанією Intuit) має можливість автоматично підключатися до банківських рахунків користувача, імпортувати транзакції та на основі історичних даних здійснювати категоризацію витрат. Він також формує прогнози на майбутній період і надає рекомендації, що частково реалізуються через алгоритми машинного навчання. У той же час рівень персоналізації цих

порад є досить загальним і не завжди враховує поведінкові особливості конкретного користувача.

Застосунок PocketGuard робить акцент на оптимізації витрат та допомагає користувачу зменшити непотрібні витрати, аналізуючи повторювані платежі, підписки та звички. Він використовує елементи ШІ для виявлення «прихованих витрат» та порівняння цін на послуги. Emma, у свою чергу, зосереджена на виявленні дубльованих витрат, обробці інформації з декількох рахунків одночасно, а також відображенні статистики в інтерактивному форматі.

Окрему нішу займають застосунки, які використовують чат-боти на основі штучного інтелекту, такі як Cleo. Взаємодія з користувачем відбувається у вигляді розмовного інтерфейсу, що створює ілюзію «живого» фінансового радника. ШІ-бот аналізує стиль витрат користувача, надає поради щодо заощадження та навіть використовує ігрові елементи й гумор для підвищення залучення користувача. Однак і в цьому випадку аналіз часто здійснюється на основі загальних шаблонів без глибокої адаптації до індивідуальних фінансових контекстів.

У процесі аналізу функціональних можливостей вказаних застосунків було виявлено спільні риси:

- автоматичне зчитування транзакцій з банківських рахунків;
- попередньо налаштовані категорії витрат;
- формування графічної аналітики (діаграм, звітів, панелей моніторингу);
- наявність базової функції прогнозування майбутніх витрат;
- інтеграція з банками та платіжними системами у країнах з високим рівнем розвитку фінтех-інфраструктури.

Разом з тим, більшість наявних застосунків мають низку істотних обмежень:

1. Обмежений рівень персоналізації. Рекомендації часто базуються на загальних статистичних даних, а не на індивідуальних патернах витрат користувача.

2. Низький рівень адаптивності моделей. Використовуються переважно базові моделі машинного навчання, які не мають здатності до самонавчання або до побудови складних поведінкових моделей.
3. Відсутність локалізації. Значна частина застосунків не підтримує українську мову, гривню та не інтегрується з банками, що діють на території України.
4. Обмежений доступ до повного функціоналу. У більшості випадків інтелектуальні функції доступні лише у платних версіях, що зменшує потенційну аудиторію користувачів.

Отже, на ринку мобільних застосунків з елементами штучного інтелекту для аналізу особистих фінансів існує чітка тенденція до автоматизації та розумної підтримки користувача. Проте більшість існуючих рішень зосереджені на типових сценаріях, не забезпечуючи достатнього рівня індивідуалізації, адаптивності до локального контексту та інтегрованості з національними платіжними системами. Ці фактори обґрунтовують необхідність подальшого дослідження та розробки нового мобільного застосунку, який би забезпечив більш глибоку персоналізацію, використовував сучасні методи штучного інтелекту та був орієнтований на реальні потреби локального користувача.

1.2. Огляд наявних аналогів

На ринку мобільних застосунків для аналізу особистих фінансів представлено чимало рішень, які намагаються задовольнити потребу користувачів у контролі витрат, веденні бюджету та покращенні фінансової дисципліни. Серед них особливе місце займають застосунки, що поєднують традиційні фінансові інструменти з елементами штучного інтелекту. Зокрема, до числа таких рішень належать

- Mint.
- PocketGuard.
- Emma.

– Cleo.

Незважаючи на спільну мету – допомогти користувачу краще розуміти свою фінансову поведінку, – кожен із цих застосунків має власну концепцію, функціональність і підхід до аналізу даних.

Mint, розроблений компанією Intuit, є одним із найстаріших і найбільш визнаних фінансових застосунків у Північній Америці. Його функціональність охоплює широкий спектр інструментів: від автоматичного імпорту транзакцій з банківських рахунків до побудови бюджету, аналізу витрат, створення фінансових цілей та моніторингу кредитного рейтингу. Застосунок використовує алгоритми для категоризації витрат і прогнозування перевищення бюджетних лімітів. Проте ці алгоритми досить загальні, і хоча вони враховують історичні дані користувача, можливості глибокої персоналізації або адаптивного навчання моделей є обмеженими. Крім того, значна частина функцій орієнтована на користувачів із США, оскільки інтеграція з банками інших країн або підтримка інших валют реалізовані частково або відсутні. (рис. 1.1).

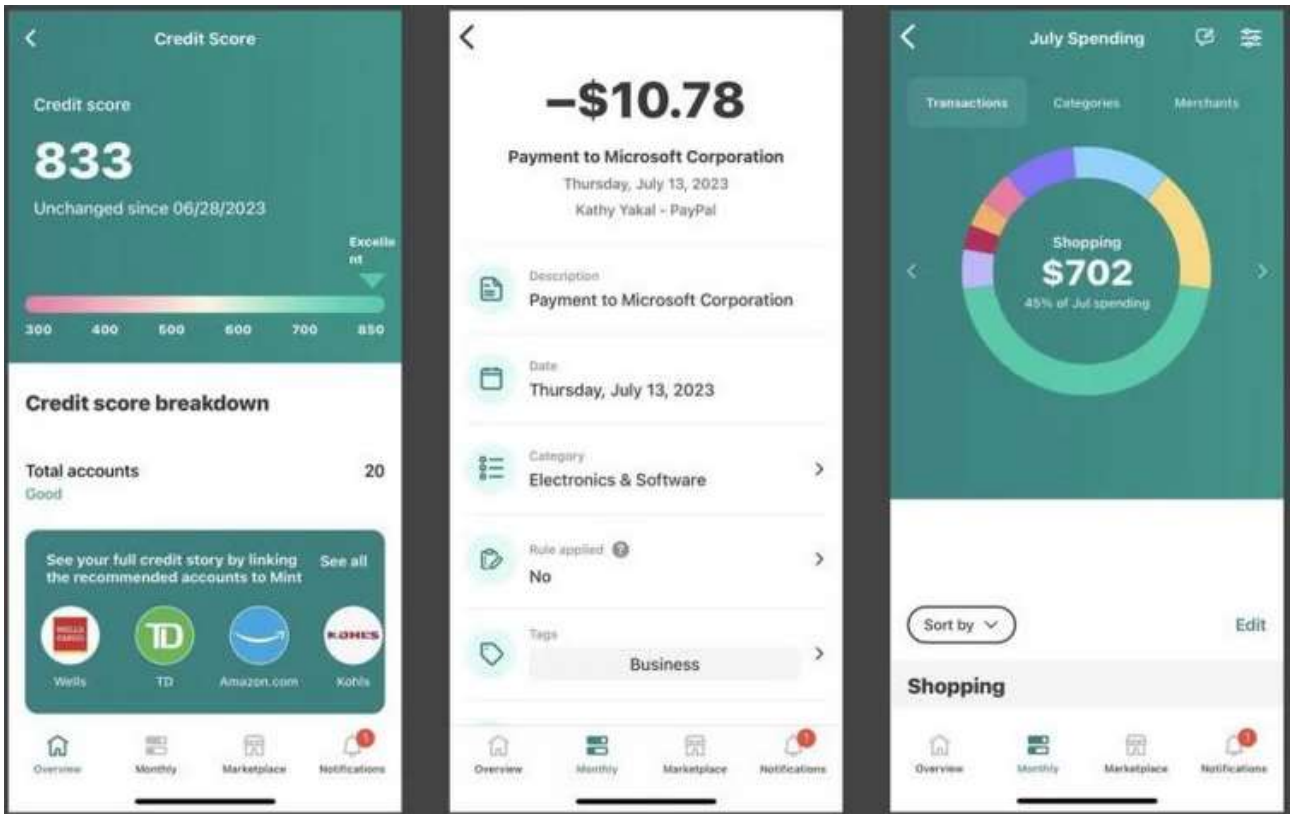


Рисунок 1.1 – Застосунок Mint

Emma [6] – це відносно новий фінансовий застосунок, який позиціонує себе як «фінансовий помічник». Його ключова функціональність включає агрегування інформації з різних банківських рахунків, виявлення дубльованих транзакцій, аналіз змін у витратах і доходах, а також створення індивідуальних фінансових оглядів. Emma також пропонує модулі для контролю підписок, моніторингу боргів і заощаджень. Штучний інтелект у цьому застосунку використовується переважно для автоматичного розпізнавання шаблонів транзакцій, кластеризації витрат за категоріями та формування звітності. Інтерфейс користувача сучасний, зручний і орієнтований на молодіжну аудиторію. У порівнянні з Mint та PocketGuard, Emma надає більше інструментів для гнучкої аналітики, хоча її інтелектуальні функції також залишаються в межах базового рівня машинного навчання. (рис. 1.2).

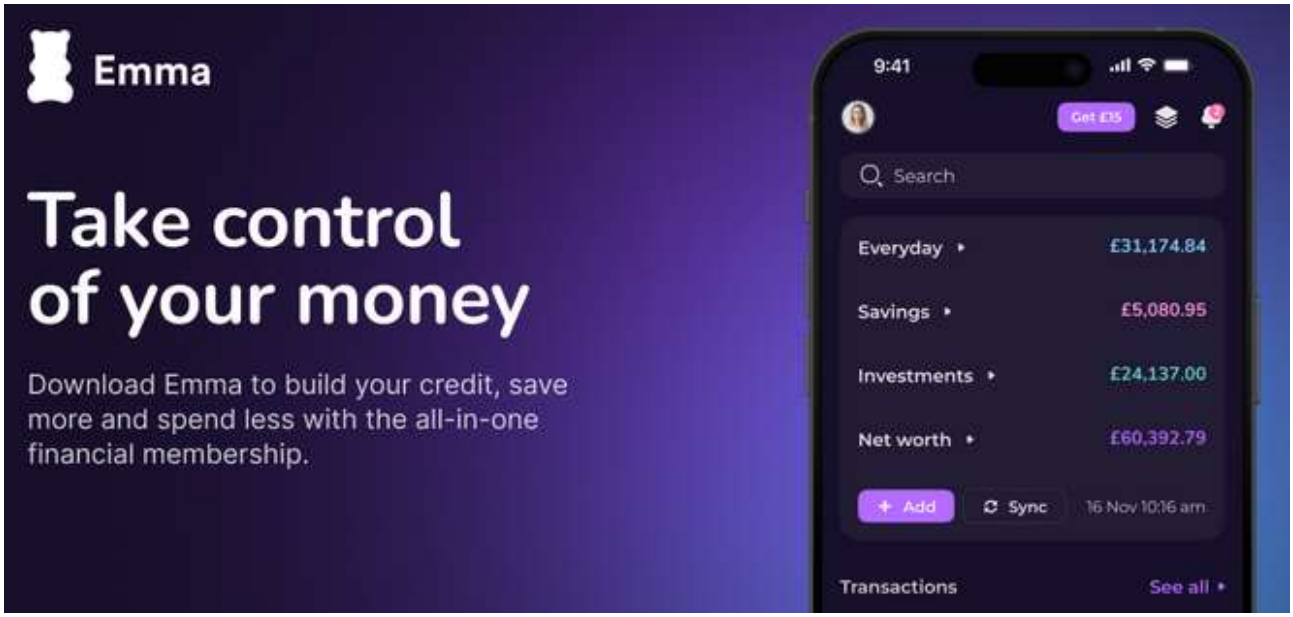


Рисунок 1.2 – Приклад "Emma"

PocketGuard позиціонує себе як інструмент, що фокусується на допомозі в економії грошей. Основна ідея полягає у відстеженні наявних коштів після врахування обов’язкових витрат, регулярних платежів та запланованих транзакцій. У цьому сенсі застосунок виконує роль фінансового асистента, який повідомляє, скільки коштів можна безпечно витратити. Алгоритми PocketGuard аналізують підписки, постійні витрати, повторювані платежі та пропонують

способи їх оптимізації. Хоча застосунок використовує елементи штучного інтелекту, наприклад у виявленні зайвих витрат, його інтелектуальні функції залишаються відносно поверхневими: аналіз ґрунтується переважно на структурі транзакцій, а не на складній поведінковій моделі користувача. Проте застосунок зручний у повсякденному використанні завдяки простому інтерфейсу та високій швидкості обробки даних (рис. 1.3).



Рисунок 1.3 – Приклад додатку PocketGuard

Cleo вирізняється серед аналогів тим, що реалізована у вигляді чат-бота. Її ключова концепція полягає у використанні штучного інтелекту для створення віртуального фінансового помічника, з яким користувач спілкується в текстовому режимі. За допомогою інтеграції з месенджерами Cleo надає аналітичні відповіді на запити користувача, наприклад: «Скільки я витратив на їжу цього тижня?» або «Скільки я можу витратити до кінця місяця?». Крім того, Cleo генерує звіти, проводить аналіз витрат, ставить фінансові цілі, пропонує поради щодо заощаджень і навіть може жартувати чи

використовувати елементи гейміфікації для підвищення залученості. Хоча глибина аналітики в Cleo обмежена, її основна перевага – це зручність взаємодії та можливість отримувати фінансову інформацію у форматі діалогу. Інтелектуальні алгоритми використовуються для розуміння запитів природною мовою (NLP) та формування відповідей на основі аналізу даних. (рис. 1.4).

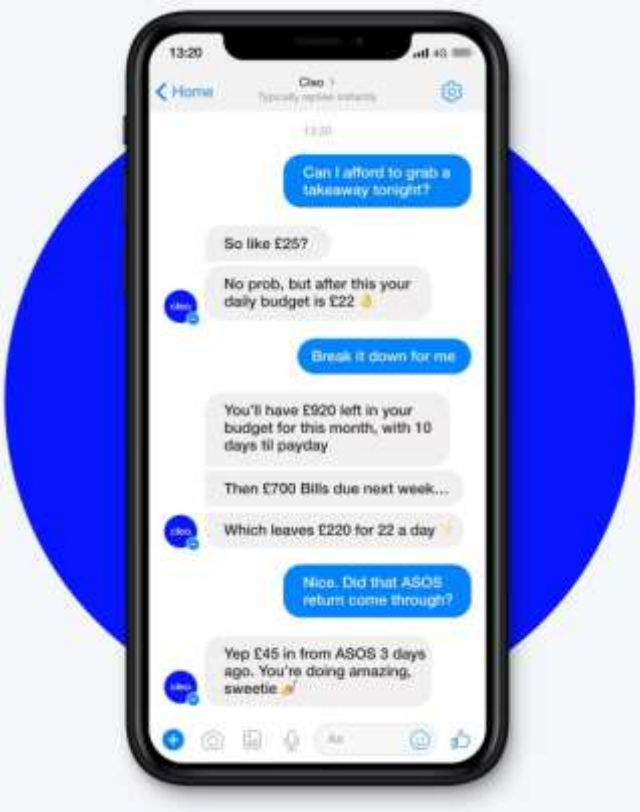


Рисунок 1.4 – Приклад Cleo

Порівнюючи ці застосунки між собою, можна зробити кілька висновків. Mint пропонує найбільш широкий функціонал, але водночас є найменш персоналізованим і адаптованим для користувачів поза межами США. PocketGuard є оптимальним для тих, хто шукає простоту та зосередженість на економії, але не пропонує розширених функцій прогнозування чи індивідуальної аналітики. Emma – гнучкий і візуально привабливий застосунок, що надає достатньо інформації для самостійного аналізу, хоча рівень автономного «розумного» аналізу у ньому обмежений. Cleo, у свою чергу, вирізняється унікальним підходом до інтерфейсу, але функціональність її

рекомендаційної системи поки що не досягає рівня глибокої аналітики чи довгострокового прогнозування.

Таким чином, жоден із розглянутих застосунків не забезпечує одночасно повної персоналізації, високого рівня інтелектуальної адаптації до поведінки користувача, гнучкого управління даними та локалізації під регіональні фінансові системи. Це свідчить про наявність вільної ніші для нових розробок, які б поєднували ці характеристики та максимально відповідали вимогам сучасного користувача – не лише як споживача фінансових сервісів, а як активного учасника цифрової економіки.

1.3. Постановка задачі

Розроблення мобільного інтелектуального застосунку, орієнтованого на аналіз особистих фінансових витрат, вимагає ґрунтовного дослідження актуальних підходів до реалізації програмних модулів, з урахуванням технічної доцільності, гнучкості архітектури та перспектив масштабування. З метою обґрунтування вибору відповідної стратегії розробки було проаналізовано два основних підходи: інтеграцію готових програмних рішень та створення власного функціонального модуля з нуля.

Перший підхід полягає у використанні наявних бібліотек, фреймворків або модульних рішень, розроблених третіми сторонами. Основною перевагою цього варіанту є висока швидкість інтеграції: програмісти можуть імпортувати готову бібліотеку, мінімізуючи витрати часу на реалізацію базового функціоналу. Однак застосування сторонніх компонентів супроводжується низкою істотних обмежень:

– правові ризики, пов’язані з умовами ліцензування;

– обмежений або повністю закритий доступ до вихідного коду, що унеможливлює глибоку адаптацію;

– залежність від оновлень, підтримки та стабільності стороннього розробника.

Альтернативним є підхід повної автономної розробки власного модуля, що передбачає реалізацію програмної логіки «з нуля». Цей метод вимагає значно більших часових і технічних ресурсів, проте забезпечує повний контроль над архітектурою, що, у свою чергу, відкриває широкі можливості для модифікації, адаптації до специфічних потреб, а також подальшого масштабування або комерційного розповсюдження. Серед ключових переваг:

– максимальна гнучкість у реалізації функціоналу відповідно до вимог проєкту;

– прогнозована технічна підтримка та обслуговування;

– незалежність від зовнішніх постачальників ПЗ;

– потенційна підтримка мультиплатформенних рішень.

Основним недоліком, притаманним даному підходу, є тривалість розробки, оскільки процес створення, тестування та оптимізації функціоналу може потребувати значного часу та людських ресурсів.

У результаті порівняльного аналізу було прийнято рішення про доцільність реалізації власного модуля як більш перспективного варіанту з погляду довгострокової підтримки, масштабування та можливості персоналізації системи під специфіку користувацьких сценаріїв. Такий підхід дозволить сформувати стабільну основу для подальшого розвитку інтелектуального мобільного застосунку.

На основі проведеного аналізу сформульовано перелік основних технічних та інженерних завдань, які необхідно реалізувати в межах проєкту:

– визначення та впровадження оптимальної моделі штучного інтелекту для аналізу фінансових даних користувача;

– розробка модуля керування та зберігання персональних даних;

– створення аналітичного ядра для взаємодії з AI-моделлю та генерації інтерпретованих результатів;

– реалізація інтерфейсного модуля з графічним відображенням результатів аналізу;

– проведення комплексного тестування функціональних блоків відповідно до поставлених вимог.

Усі вищезазначені компоненти повинні бути реалізовані з дотриманням принципів безпеки, масштабованості та взаємодії між модулями, що дозволить створити конкурентоспроможний програмний продукт на перетині фінансових технологій і штучного інтелекту.

РОЗДІЛ 2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ МОБІЛЬНОГО ЗАСТОСУНКУ

2.1. Аналіз і обґрунтування вибору засобів для реалізації застосунку

У процесі розроблення мобільного інтелектуального застосунку для аналізу особистих витрат надзвичайно важливим є обґрунтований вибір технологічного стеку, що забезпечить як функціональну ефективність, так і масштабованість системи. Підбір мов програмування, бібліотек, фреймворків та інструментів штучного інтелекту має відбуватися на основі сучасних технічних тенденцій та практичних переваг у сфері мобільної розробки.

Мова програмування Kotlin є основною мовою для створення застосунків на платформі Android, що офіційно підтримується корпорацією Google. Вона поступово витіснила попередню домінуючу мову – Java – завдяки низці переваг: високий рівень лаконічності синтаксису, підвищена безпека типізації та ефективність обробки винятків. Дослідження показують, що використання Kotlin сприяє зниженню частоти збоїв у роботі Android-додатків більш ніж удвічі, а також дозволяє впроваджувати новітні технології, що прискорюють розробку програмного забезпечення [13].

Фреймворк Jetpack Compose виступає сучасним декларативним інструментом побудови користувацького інтерфейсу в Android-додатках. На відміну від традиційної XML-верстки, Compose пропонує зручні API, що суттєво спрощують процес розробки інтерфейсу, дозволяючи розробникам створювати адаптивні UI-компоненти швидше та з меншою кількістю коду [14]. Таким чином, використання Compose сприяє підвищенню продуктивності розробки та полегшує подальшу підтримку застосунку.

Важливою складовою запропонованого рішення є застосування штучного інтелекту на базі **моделі** ChatGPT, розробленої OpenAI. Дана модель побудована на архітектурі трансформерів (GPT – Generative Pre-trained Transformer) і спеціалізується на обробці та генерації текстових повідомлень у діалоговому форматі. Переваги ChatGPT полягають у здатності обробляти

широкий спектр тем завдяки навчанню на великому корпусі текстових даних, універсальності застосування (від генерації відповідей до контекстного аналізу) та високій здатності до утворення смислово релевантного контенту на основі заданого контексту [15].

Для реалізації локального зберігання даних доцільним є використання бібліотеки и, що забезпечує зручний рівень абстракції над механізмом SQLite. Room дозволяє виконувати перевірку SQL-запитів на етапі компіляції, використовує анотації для автоматизації генерації коду та підтримує ефективні стратегії міграції баз даних, що є критично важливим для забезпечення надійного функціонування додатку в довгостроковій перспективі [16; 17].

Для реалізації механізмів автентифікації та управління користувачами передбачено використання платформи Firebase. Вона відзначається простотою інтеграції, високим рівнем надійності, масштабованістю, а також підтримкою безкоштовного плану, що робить її зручним інструментом для мобільних застосунків на початковому етапі.

Як середовище розробки обрано Android Studio – офіційно рекомендовану Google інтегровану середу розробки (IDE) для Android-платформи, що забезпечує повноцінну підтримку Kotlin, Compose, Firebase та інших компонентів, необхідних для розроблення застосунку.

Синергетичне поєднання вищезгаданих технологій – Kotlin, Jetpack Compose, ChatGPT, Firebase та Room – створює міцну технічну основу для реалізації інтелектуального мобільного рішення у сфері персональних фінансів. Kotlin і Compose забезпечують чисту, надійну та зручну для супроводу кодову базу з адаптивним графічним інтерфейсом. ChatGPT виконує роль аналітичного модуля з обробки фінансової інформації користувача. Firebase реалізує механізми безпечної ідентифікації та керування користувачами, тоді як Room забезпечує ефективне зберігання і обробку даних.

Таким чином, обґрунтований вибір інструментів розроблення мобільного застосунку зі складовими штучного інтелекту є визначальним фактором для створення функціонального, стабільного та масштабованого продукту.

Комбінація згаданих технологій дозволяє ефективно реалізувати цілі проєкту в межах сучасної мобільної екосистеми.

2.2. Розробка методу управління даними користувача

У межах розроблення мобільного інтелектуального застосунку для аналізу особистих фінансів передбачено реалізацію двошарової архітектури управління даними, що включає шари data та domain [10]. Такий підхід зумовлений необхідністю підвищення модульності програмного забезпечення, спрощення процесів обробки даних, а також забезпечення чіткої логічної структуризації відповідно до принципів розділення відповідальностей. Для забезпечення зв’язку між зазначеними шарами впроваджуються спеціальні перетворювальні компоненти – класи типу Mapper [11], які відповідають за коректну трансформацію об’єктів з одного шару до іншого без втрати семантики або інформаційного змісту.

Шар data виконує функції взаємодії з джерелами зберігання даних (локальною базою даних) і компонентами штучного інтелекту. Основною його характеристикою є оперування примітивними типами даних, такими як String, Float, Long, Int, що відповідає вимогам щодо серіалізації та сумісності з механізмами збереження та передачі інформації в зовнішні системи. Структури цього шару мають в іменах маркер Entity, що слугує для чіткого відокремлення їх від об'єктів інших рівнів.

Шар domain, натомість, призначений для опрацювання більш абстрактних, контекстно насичених об'єктів, що безпосередньо застосовуються на рівні бізнес-логіки додатку. Наприклад, у моделях цього шару можуть одночасно зберігатися назва категорії, її візуальне представлення (у вигляді зображення) та колірне кодування. Об'єкти domain-рівня не мають додаткових приставок у назві, оскільки вважаються фінальними представленнями даних, готовими до обробки та відображення у графічному інтерфейсі користувача.

Процес взаємодії між шарами можна проілюструвати на прикладі створення транзакції користувачем. На етапі введення інформації у відповідній

формі застосунку активуються структури domain-рівня, які репрезентують змістовно збагачені об’єкти. Після підтвердження збереження даних викликається механізм перетворення – Mapper, який трансформує ці об'єкти у формат, сумісний з шаром data, після чого інформація зберігається у локальному сховищі. Зворотний процес відбувається при необхідності візуалізації збережених транзакцій: дані з бази, представлені у вигляді Entity-структур, перетворюються за допомогою відповідного Mapper-класу у domain-моделі, що адаптовані для графічного відображення.

Схематичне представлення описаної архітектури наведено на рисунку 2.1, де відображено зв’язки між шарами та етапи трансформації даних у межах життєвого циклу транзакції користувача.

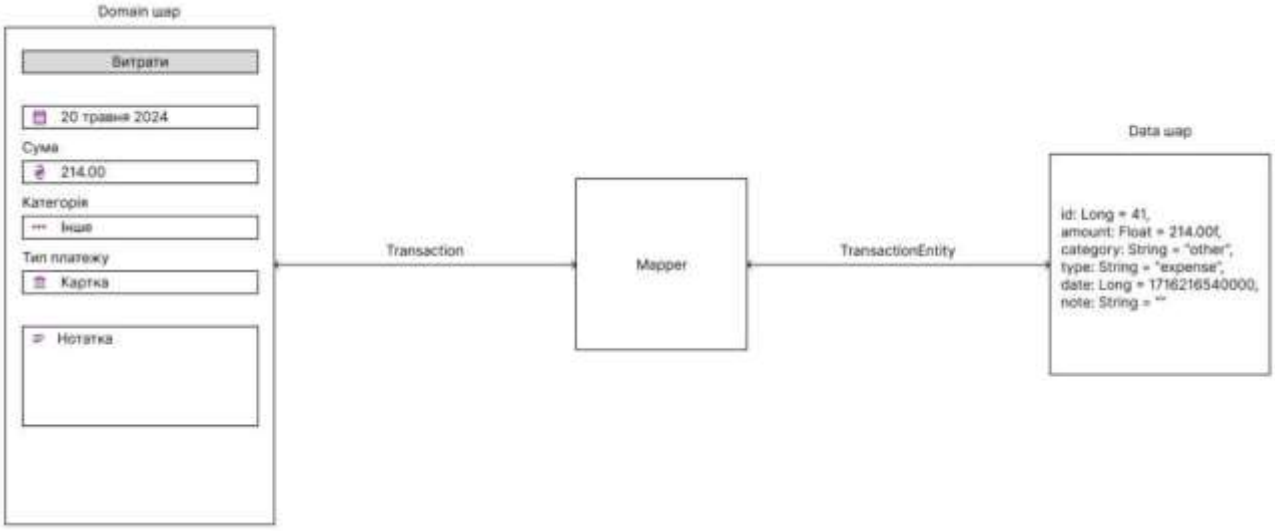


Рисунок 2.1 – Схематичне представлення архітектури

Подальший етап реалізації програмного забезпечення мобільного інтелектуального застосунку для аналізу особистих фінансів передбачає поетапну розробку ключових функціональних модулів. На початковому етапі планується створення модуля управління користувацькими даними, який відповідатиме за акумулювання, обробку та захищене зберігання персональної фінансової інформації. Реалізація даного модуля є критично важливою для забезпечення високого рівня конфіденційності, цілісності даних і ефективної взаємодії з іншими компонентами системи, зокрема модулями штучного інтелекту.

Після завершення процесу збору повної фінансової інформації користувача (включно з витратами, доходами та іншими релевантними параметрами), розроблятиметься модуль аналітики та взаємодії з інтелектуальною моделлю. Його основним завданням буде обробка вхідних даних і генерація релевантних висновків або рекомендацій на основі машинного аналізу.

Наступним етапом стане реалізація модуля графічного відображення інформації. Оскільки результати, отримані від моделі штучного інтелекту, можуть бути складними для сприйняття пересічним користувачем, візуалізація відіграє ключову роль у забезпеченні зручного, інтуїтивно зрозумілого та естетично привабливого інтерфейсу.

Таким чином, розробка мобільного інтелектуального застосунку для моніторингу особистих фінансів є багатоетапним процесом, який вимагає чіткої структуризації, професійного проєктування та високоякісного програмного втілення. За умови належного технічного супроводу та дотримання принципів інженерного проєктування, можливо створити інноваційний інструмент, здатний суттєво покращити користувацький досвід управління особистими фінансами. Успішна реалізація поставленої мети залежить від ефективної інтеграції, тестування та оптимізації всіх функціональних модулів системи.

2.3. Розробка моделі системи та методу аналізу особистих витрат

З метою більш глибокого розуміння функціональної логіки роботи мобільного інтелектуального застосунку для аналізу особистих витрат було розроблено концептуальну модель процесів у вигляді UML-діаграми діяльності [12] (рис. 2.2). Така візуалізація дозволяє формалізувати послідовність дій користувача та внутрішню логіку взаємодії між компонентами системи.

Згідно з представленою діаграмою, після запуску застосунку користувач розпочинає взаємодію з системою шляхом введення фінансових транзакцій, які слугують вхідними даними для подальшого аналізу. Наступним кроком є надання додаткової інформації, необхідної для коректного функціонування

модуля штучного інтелекту, зокрема – параметрів, які уточнюють контекст фінансової поведінки користувача.

Після заповнення всіх обов’язкових полів користувач активує процес аналізу, ініціюючи генерацію рекомендацій. На цьому етапі система перевіряє наявність автентифікації користувача. У разі відсутності авторизації відбувається автоматичне перенаправлення на екран входу. Якщо ж автентифікація пройдена успішно, запускається механізм формування порад на основі введених даних та алгоритмів машинного навчання.

Після завершення процесу обчислення результати стають доступними для перегляду в інтерфейсі застосунку. Таким чином, розроблена UML-діаграма діяльності дає змогу узагальнено відобразити життєвий цикл користувацької взаємодії з системою, а також забезпечує основу для подальшого технічного проєктування та тестування функціональних сценаріїв.

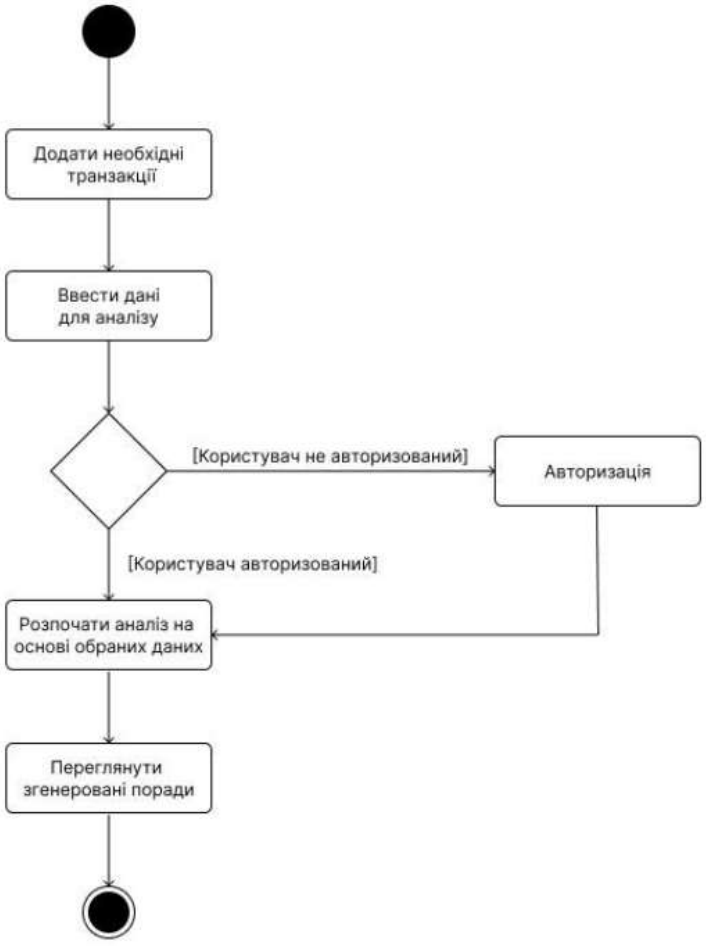


Рисунок 2.2 – UML-діаграма функціональної активності мобільного застосунку

Для реалізації мобільного інтелектуального застосунку, призначеного для аналізу особистих витрат, необхідним є формування узагальненого алгоритму функціонування, що визначатиме логіку роботи системи (рис. 2.3).

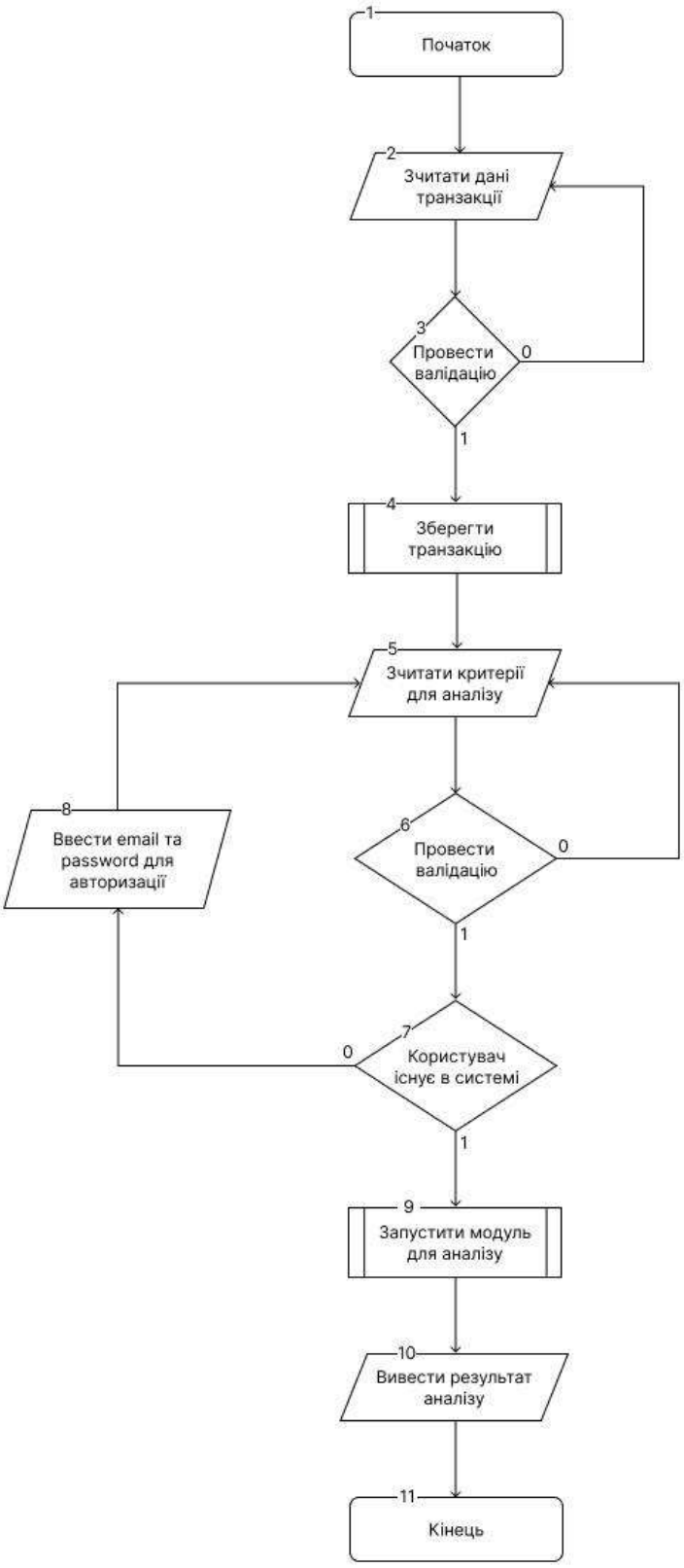


Рисунок 2.3 –Алгоритм роботи мобільного додатку

Відповідно до сформованого алгоритму, початковим етапом взаємодії користувача з мобільним інтелектуальним застосунком є введення наявних фінансових транзакцій. Перед збереженням інформації здійснюється перевірка відповідності введених даних визначеним критеріям (валідація). У разі успішного проходження перевірки дані зберігаються у локальному сховищі для подальшого використання модулем штучного інтелекту.

На наступному етапі користувач обирає параметри, відповідно до яких бажає отримати персоналізовані рекомендації. Зазначені параметри також підлягають процедурі валідації. Після цього система перевіряє автентичність користувача – у випадку відсутності облікового запису, здійснюється автоматичне перенаправлення на інтерфейс авторизації. Якщо обліковий запис існує, програма отримує збережені дані та передає їх до модуля штучного інтелекту для подальшого аналізу й генерації рекомендацій. Таким чином, під час створення транзакцій активується модуль управління користувацькими даними, тоді як під час запиту до системи штучного інтелекту викликається відповідний аналітичний модуль, відповідальний за обробку та інтерпретацію даних. З метою деталізації процесу функціонування модуля управління даними було створено окремий алгоритмічний блок (рис. 2.4).



Рисунок 2.4 – Алгоритм модуля управління даними

Відповідно до сформованого алгоритму, початковим етапом є зчитування вхідних даних із полів введення інтерфейсу користувача. Після проходження процедури валідації введені дані конвертуються у відповідні типи, що відповідають вимогам локального сховища, після чого здійснюється їх збереження у базі даних пристрою.

Збережена інформація у подальшому може використовуватись не лише для обробки та аналізу за допомогою модуля штучного інтелекту, а й для формування спискових подань, візуалізації у вигляді графіків або перегляду деталізованої інформації про транзакції. Окрім створення нових записів, система підтримує можливість редагування наявних транзакцій. Для цього дані вибірково витягуються з локальної бази, проходять трансформацію у формат, зручний для подальшого редагування, після чого користувач вносить необхідні зміни. Оновлені значення знову проходять перевірку на коректність, трансформуються у відповідну структуру та повторно зберігаються у базі даних згідно описаної логіки.

З метою деталізації логіки функціонування модуля, що забезпечує взаємодію із системою штучного інтелекту для отримання та обробки аналітичних результатів, було розроблено окремий алгоритмічний блок (рис. 2.5).

Після наджодження даних із локального сховища, вони проходять процес трансформації у спеціалізовані структуровані моделі, придатні для подальшої обробки засобами штучного інтелекту. Після надсилання запиту та завершення аналітичного обчислення, модель повертає результати, які, у свою чергу, підлягають повторній трансформації – цього разу у форму, зручну та зрозумілу для кінцевого користувача. Завершальний етап передбачає підготовку згенерованих рекомендацій до виведення в інтерфейсі мобільного застосунку.



Рисунок 2.5 – Алгоритм генерації рекомендацій за допомогою штучного інтелекту

2.4. Проєктування модуля керування користувацькими даними

Одним із ключових завдань мобільного застосунку є забезпечення безпеки та коректної обробки персональних даних користувача. З цією метою було реалізовано модуль керування користувацькими даними, функціональність якого охоплює створення та редагування фінансових транзакцій.

Алгоритмічна послідовність створення транзакції представлена на рисунку 2.6.

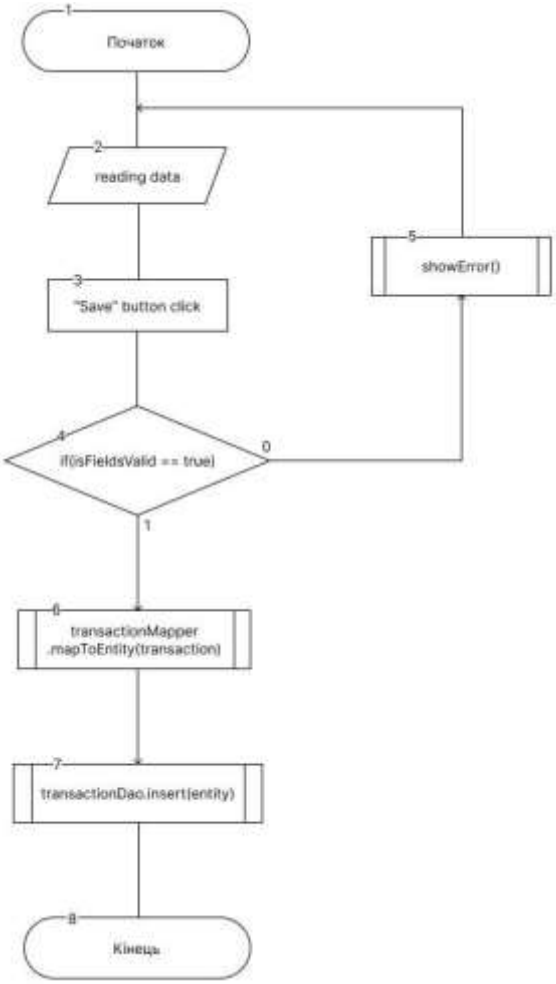


Рисунок 2.6 – Алгоритм створення фінансової транзакції

Розглянемо поетапну логіку реалізації процесу створення фінансової транзакції в рамках функціональності мобільного інтелектуального застосунку:

1. Ініціація процедури створення транзакції.
2. Зчитування вхідних даних, введених користувачем (сума, дата, категорія тощо).
3. Фіксація події взаємодії з кнопкою підтвердження (збереження).
4. Проведення перевірки коректності введених значень: відповідність формату числових полів, правильність структури дати тощо.
5. У разі виявлення помилок – відображення відповідного повідомлення про помилку та повернення до етапу введення даних.
6. За умови успішної валідації дані трансформуються у формат, сумісний з вимогами локальної бази даних.
7. Оброблені дані зберігаються у локальному сховищі (базі даних).
8. Завершення процедури створення транзакції.

У подальшому, для забезпечення гнучкості роботи із збереженими записами, передбачено механізм редагування транзакцій. Відповідний алгоритм представлено на рисунку 2.7.

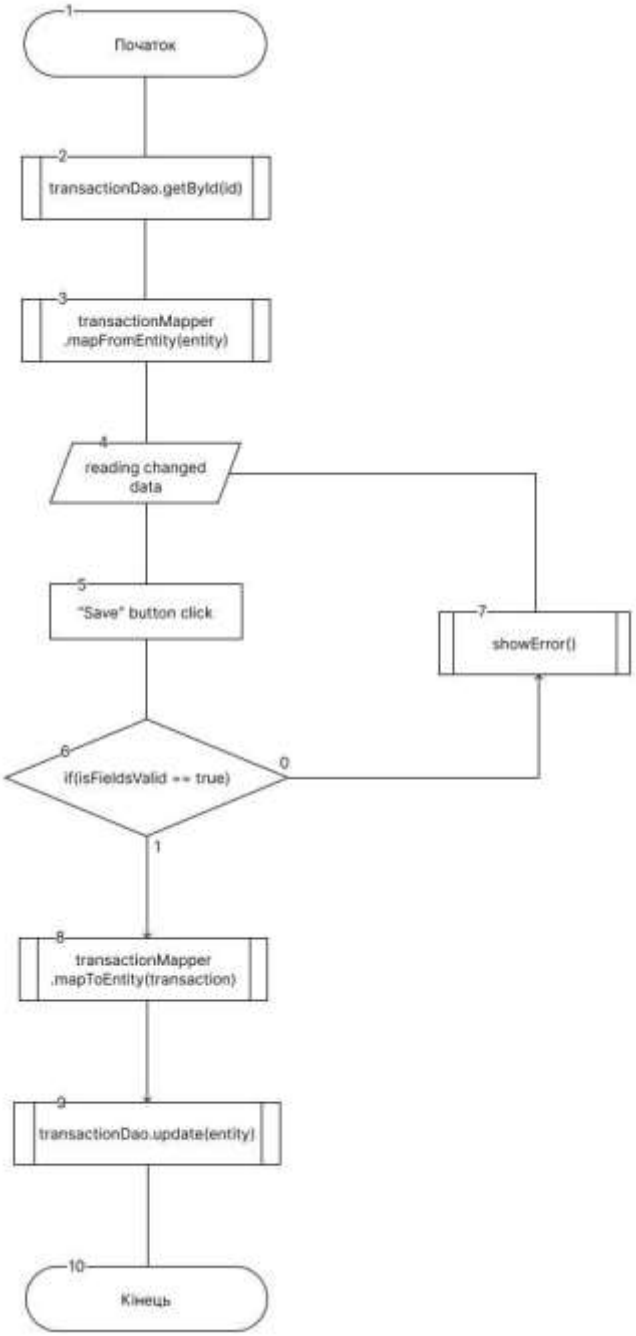


Рисунок 2.7 – Редагування фінансової транзакції

Алгоритм редагування транзакції передбачає наступну послідовність дій:

1. Ініціація процесу редагування.
2. Завантаження відповідної транзакції з локальної бази даних.
3. Трансформація отриманих даних у формат, зручний для відображення у графічному інтерфейсі користувача.

4. Внесення змін до параметрів транзакції користувачем.
5. Фіксація дії натискання кнопки «Збереження».
6. Проведення перевірки введених значень на відповідність встановленим вимогам (формат числових даних, коректність дати тощо).
7. У випадку виявлення некоректних даних – виведення повідомлення про помилку та запобігання збереженню змін.
8. За умови успішного проходження валідації – трансформація оновлених даних у структуру, придатну для збереження в базі даних.
9. Збереження модифікованої транзакції у локальному сховищі.
10. Завершення процедури редагування.

2.5. Проєктування модуля для аналітичної обробки та отримання результатів від системи штучного інтелекту

Основною функціональною метою розробленого модуля є отримання рекомендацій від моделі штучного інтелекту на основі попередньо проаналізованих даних користувача. З цією метою було створено відповідний алгоритм взаємодії із системою штучного інтелекту, що забезпечує послідовне опрацювання вхідних даних та формування відповідей (рис. 2.8).

Опис основних етапів алгоритму:

1. Ініціація процесу.
2. Введення та зчитування вхідних параметрів, що надаються користувачем.
3. Активація генерації рекомендацій шляхом натискання кнопки «Згенерувати».
4. Перевірка валідності введених даних (наприклад, правильність формату, наявність усіх обов’язкових полів).
5. У випадку невдалої перевірки – виведення повідомлення про помилку та повернення до етапу введення даних.
6. За умови коректного введення – перевірка наявності авторизації користувача в системі.

7. У разі відсутності авторизації – перенаправлення користувача на інтерфейс входу в систему, після чого виконання алгоритму продовжується з етапу зчитування даних.
8. Формування сесії взаємодії з ШІ на основі введених параметрів.
9. Перенаправлення користувача на інтерфейс чату.
10. Запуск процесу генерації рекомендацій за допомогою моделі штучного інтелекту.
11. Інтеграція сформованих порад до поточної сесії.
12. Завершення виконання алгоритму.

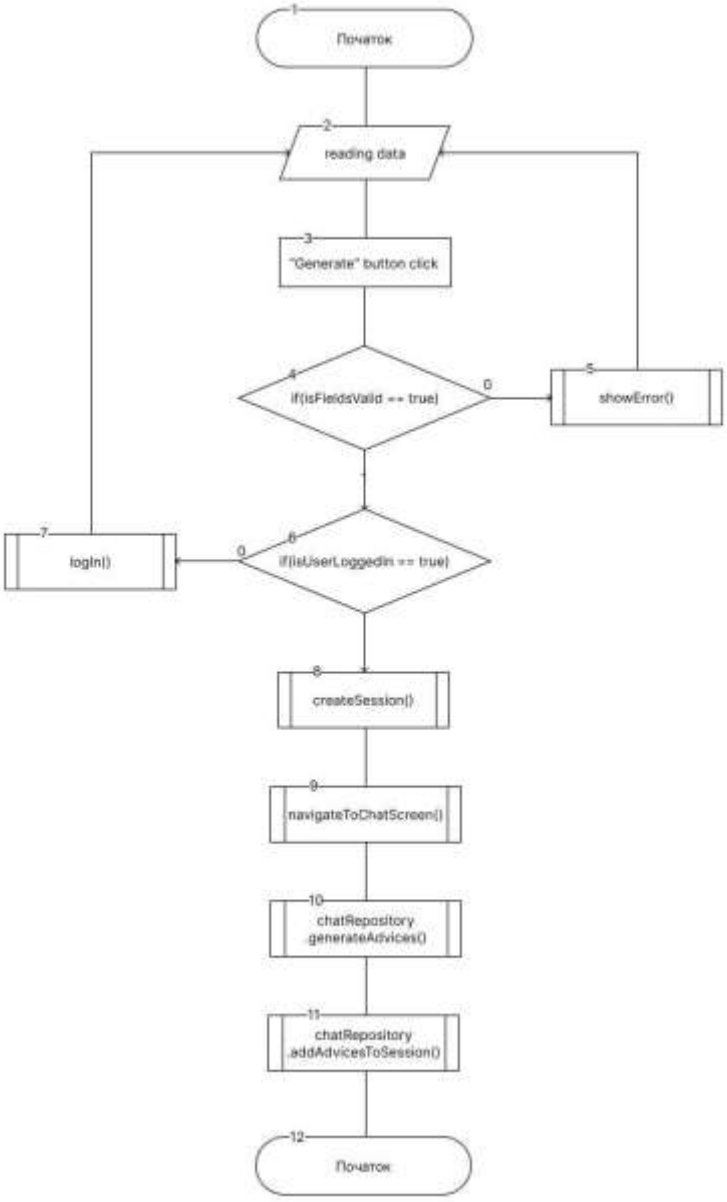


Рисунок 2.8 – Алгоритм взаємодії із системою штучного інтелекту

РОЗДІЛ 3 РЕАЛІЗАЦІЯ МОДУЛІВ МОБІЛЬНОГО ЗАСТОСУНКУ

3.1. Реалізація користувацького інтерфейсу мобільного інтелектуального застосунку

Було прийнято рішення про створення інтерфейсу мобільного застосунку з акцентом на простоту та зручність взаємодії для кінцевого користувача. В якості основної кольорової палітри обрано чорно-білу гаму, що сприяє уникненню візуальних конфліктів та дозволяє ефективно акцентувати увагу на ключових елементах інтерфейсу.

Після запуску застосунку користувачеві відображається екран завантаження із логотипом програми. У цей момент у фоновому режимі виконується підготовка необхідних даних для початкового відображення, а також здійснюється перевірка наявності авторизованого користувача в системі (рис. 3.1).

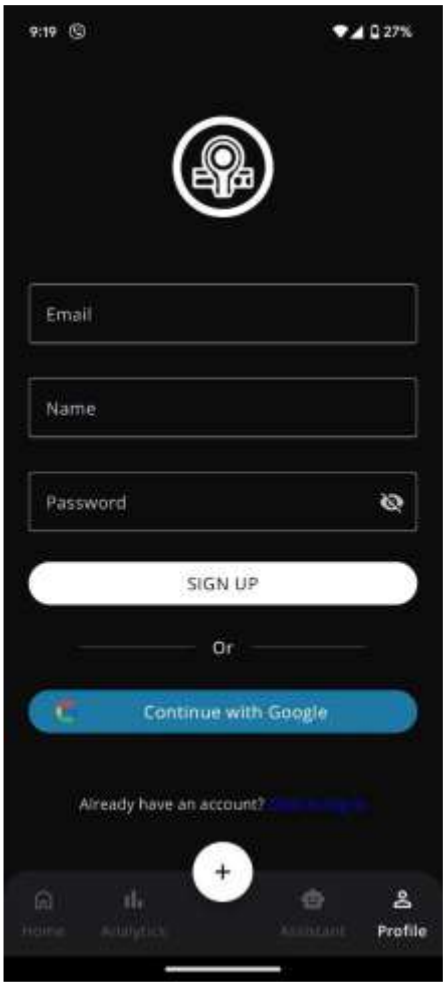


Рисунок 3.5 – Інтерфейс реєстрації

У мобільному застосунку передбачено механізм авторизації користувачів, що є необхідним для забезпечення персоналізованого доступу та безпеки даних. З цією метою було спроєктовано та реалізовано інтерфейс реєстрації нового користувача в системі. Для створення облікового запису користувачу необхідно ввести ім’я, адресу електронної пошти та пароль.

Після успішного завершення реєстрації система перенаправляє користувача на екран авторизації (рис. 3.2). Для входу в систему передбачено два способи: за допомогою електронної пошти та пароля або через інтеграцію з обліковим записом Google.

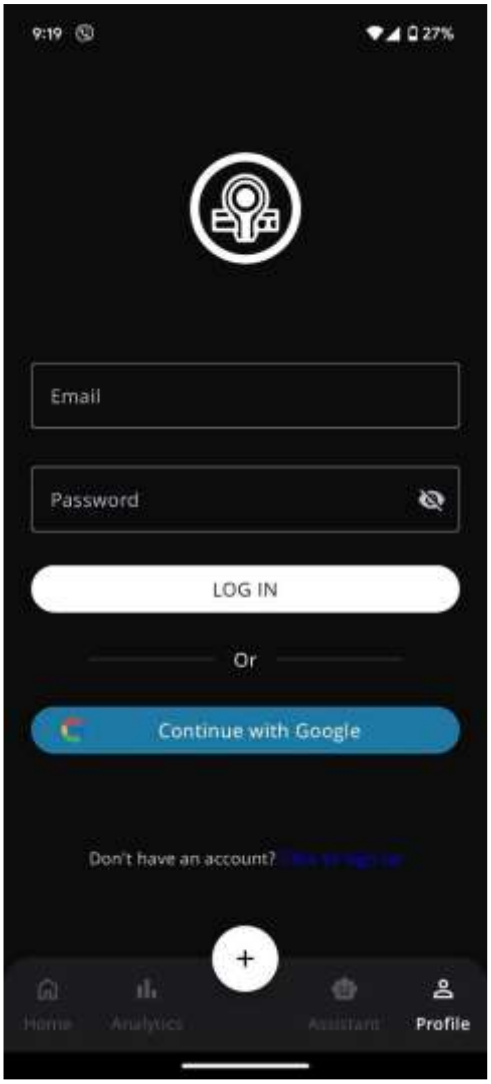


Рисунок 3.2 – Інтерфейс авторизації

На головному екрані застосунку (рис. 3.3) користувачу надається можливість переглянути поточний баланс, загальну суму витрат і доходів за поточний місяць, а також список останніх здійснених транзакцій.

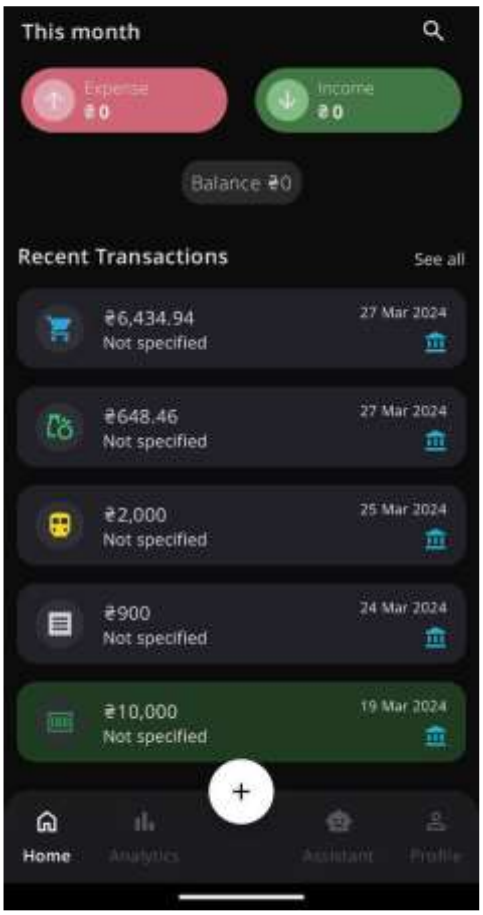


Рисунок 3.3 – Інтерфейс екрану

Після натискання на іконку пошуку відкривається інтерфейс для пошуку транзакцій, який дозволяє здійснювати пошук за назвою та застосовувати різноманітні фільтри (рис. 3.4).

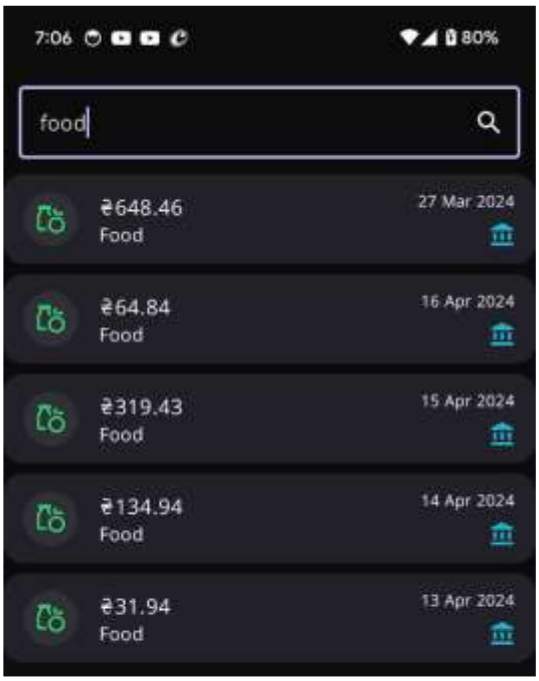


Рисунок 3.4 – Інтерфейс пошуку

Користувач також має доступ до окремого екрана, призначеного для перегляду всіх наявних транзакцій (рис. 3.5). Перехід на цей екран здійснюється шляхом натискання на текстове посилання "See all".

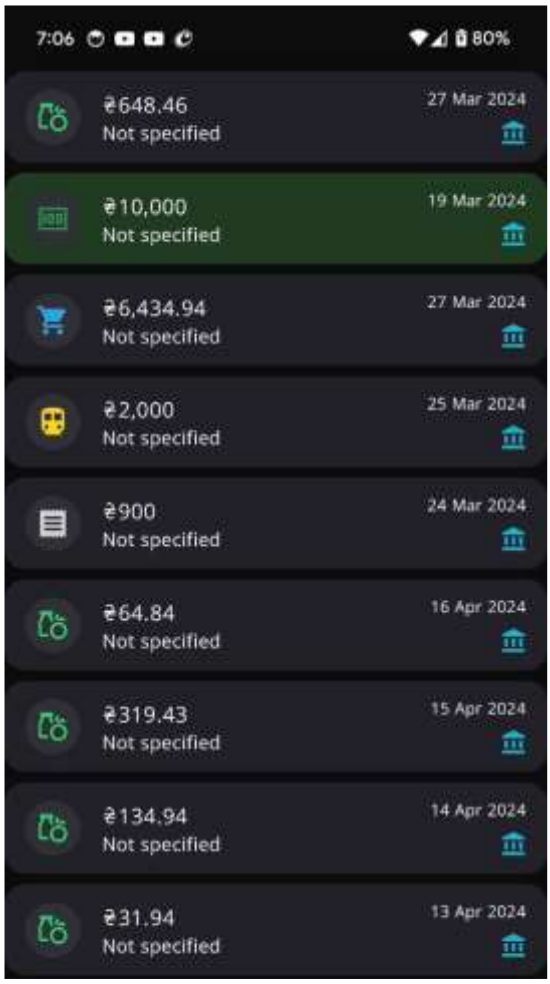


Рисунок 3.5 – Інтерфейс всіх транзакцій

На другій вкладці навігаційної панелі розміщено екран аналітики витрат і доходів. При його відкритті користувач отримує можливість обрати бажаний часовий інтервал для перегляду фінансової аналітики. Натискання на стрілку поруч із діаграмою перенаправляє користувача на окремий екран, де подається аналіз фінансів за обраний період, здійснений штучним інтелектом.

Для перегляду аналітики за поточний тиждень необхідно натиснути кнопку "Week" та вказати відповідний інтервал. Аналогічно, при виборі опцій "Month" або "Year" користувач отримує аналітичні дані за місяць або рік відповідно (рис. 3.6).



Рисунок 3.6 – Інтерфейс з аналізом фінансів за тиждень, місяць, рік

3.2. Реалізація штучного інтелекту для аналізу витрат

На третій вкладці інтерфейсу розміщено екран для вибору параметрів, необхідних для проведення аналізу та подальшої генерації рекомендацій за допомогою штучного інтелекту (рис. 3.7). Користувачеві доступні поля для введення назви запиту, вибору часових меж, в межах яких здійснюватиметься аналіз фінансових транзакцій, типу фінансової мети, зазначення бажаного розміру фінансової цілі у гривнях, очікуваної дати її досягнення, а також текстової нотатки, яка додатково враховується під час обробки даних системою.



Рисунок 3.7 – Інтерфейс введення даних для оптимізації

У випадку вибору цілі "оптимізація бюджету" (рис. 3.8), деякі поля введення, зокрема сума фінансової цілі у гривнях та бажана дата її досягнення, автоматично приховуються з інтерфейсу. Відповідно, ці дані не враховуються під час подальшого аналізу штучним інтелектом.

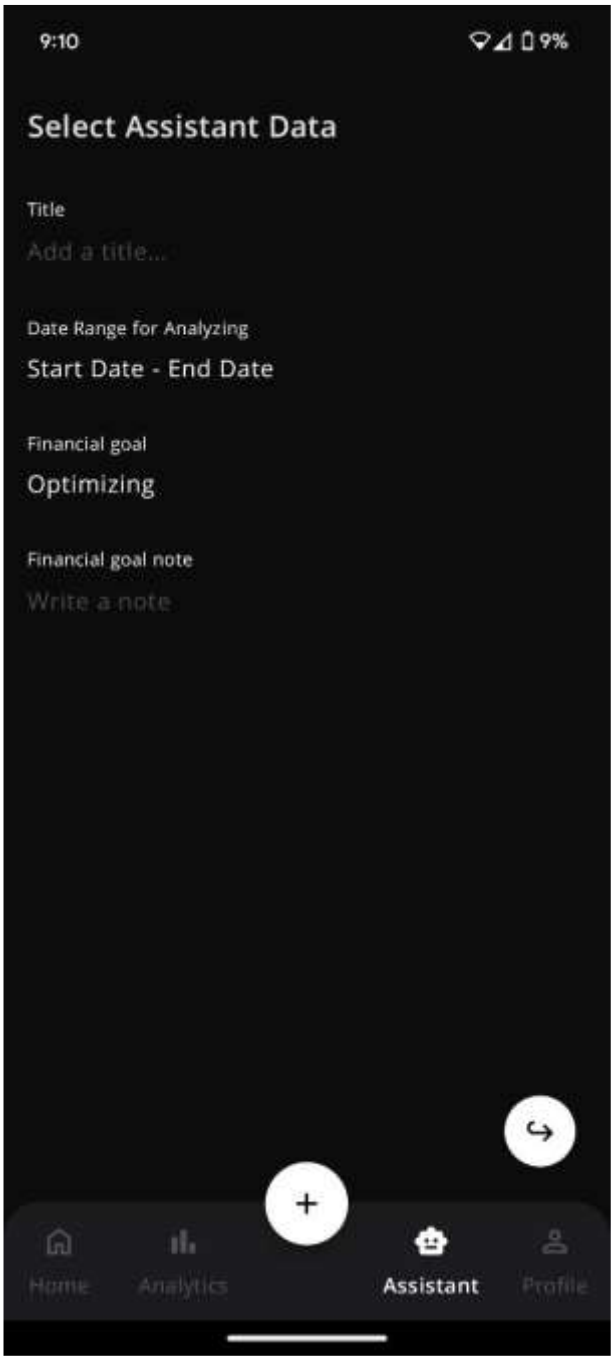


Рисунок 3.8 – Інтерфейс "Оптимізації"

Після заповнення всіх необхідних полів і натискання на білу іконку зі стрілкою, користувач буде перенаправлений на екран, де відображаються поради, згенеровані штучним інтелектом (рис. 3.9).

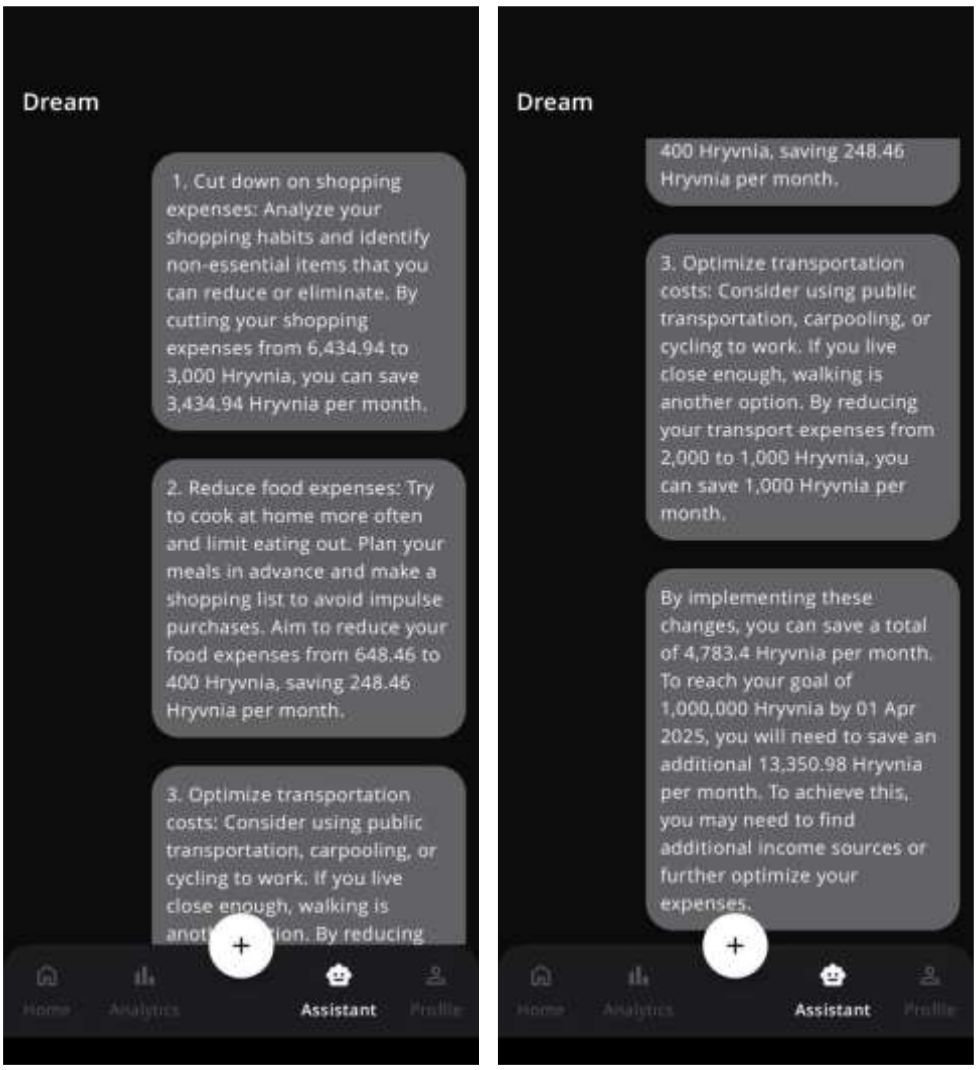


Рисунок 3.9 – Інтерфейс з порадами від ШІ

Модель штучного інтелекту за замовчуванням генерує три поради. На відповідному екрані користувач може ознайомитися не лише з отриманими рекомендаціями, але й з узагальненою назвою запиту та введеними раніше даними, що були обрані на попередньому етапі (рис. 3.9).

Для додавання нових транзакцій передбачено кнопку у вигляді білої іконки з плюсом, яка доступна з будь-якого екрану застосунку. Після натискання відкривається вікно з двома опціями: додавання витрат або доходів (рис. 3.10). В обох режимах користувач має можливість зазначити такі параметри: дату, суму, категорію, тип оплати та додати коментар.

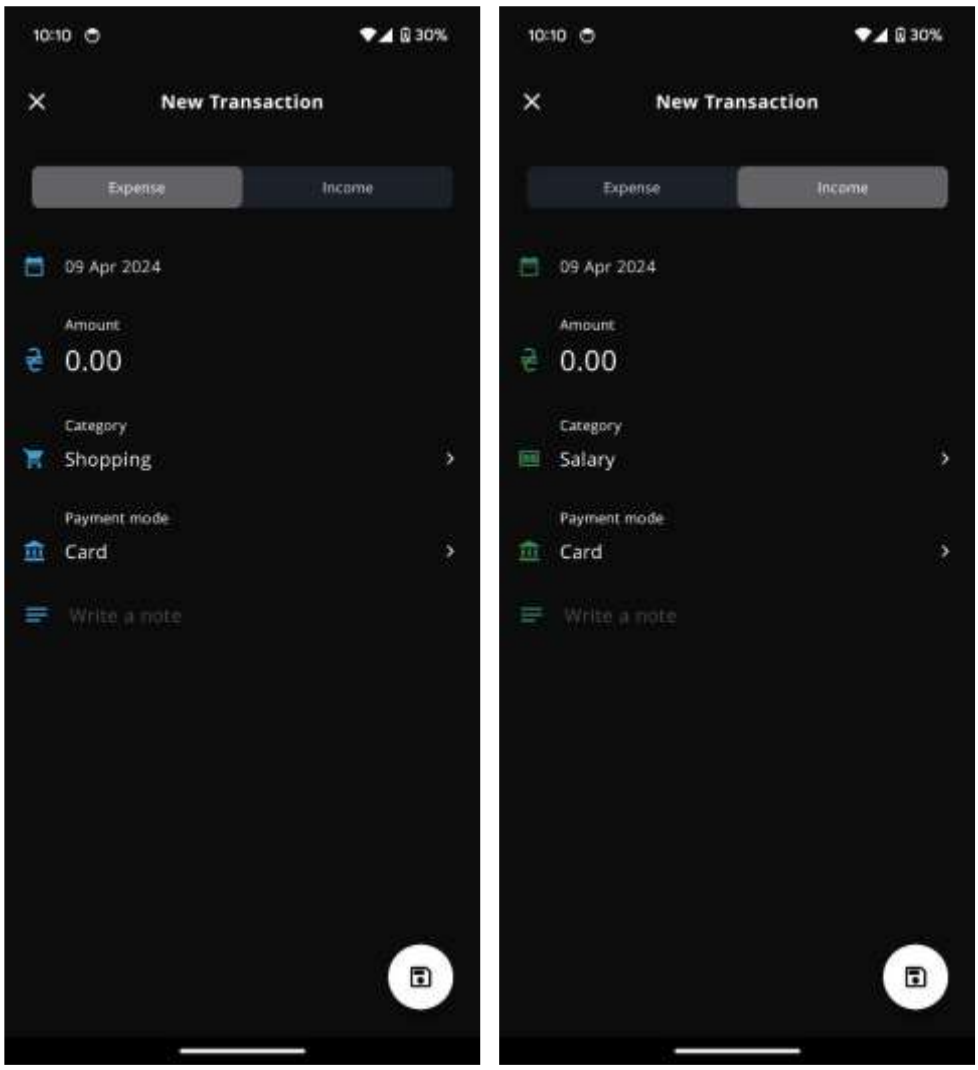


Рисунок 3.10 – Інтерфейс витрат та доходів

Розробка здійснювалося в середовищі розробки Android Studio із використанням Jetpack Compose [18].

3.3. Тестування мобільного застосунку для аналізу персональних витрат із використанням технологій штучного інтелекту.

Тестування мобільного застосунку на платформі Android [19] є критично важливим етапом розробки, оскільки саме воно визначає рівень якості програмного продукту, відповідність очікуванням користувачів та стабільність функціонування на різноманітних пристроях.

Першим кроком у процесі тестування є встановлення тестової версії застосунку на мобільний пристрій з операційною системою Android. Далі здійснюється проведення функціонального тестування, яке охоплює перевірку коректності введення та виведення даних, обробку неочікуваних вхідних значень, а також відповідність реалізації бізнес-логіки заданим вимогам. Підсумки проведеного тестування представлено на рисунку 3.11.



Рисунок 3.11 – Підсумки проведення тестів

Наступним етапом є проведення тестування безпеки застосунку, зосередженого на перевірці збереження цілісності даних під час виконання основних операцій – створення, редагування, видалення, а також у процесі міграції бази даних. Результати цього тестування наведено на рисунку 3.12.

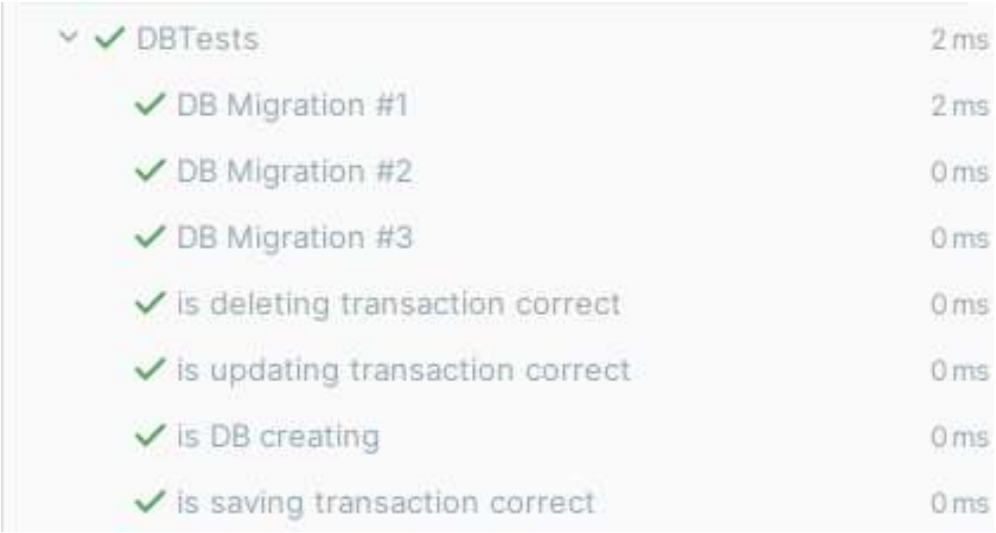


Рисунок 3.12 – Результат тестування безпеки

Наступним етапом є тестування графічних елементів інтерфейсу, метою якого є перевірка їх коректного відображення, відповідності розмірів та розташування, а також належної реакції на взаємодію з користувачем. Крім того, UI-тести [20] дають змогу виявити потенційні недоліки у зручності користування, такі як невдале розташування елементів управління або недостатньо зрозумілі підказки. Результати тестування наведено на рисунку 3.13.

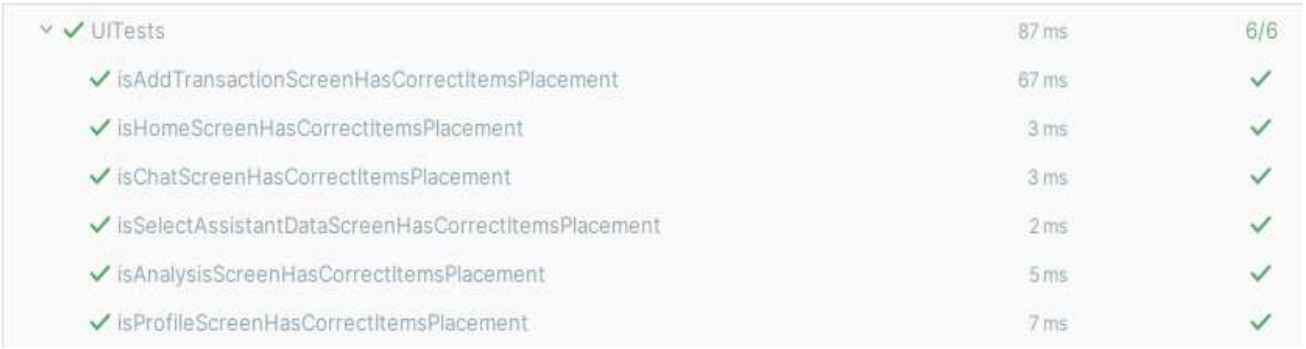


Рисунок 3.13 – Результат тестування графіки

Після успішного завершення всіх тестувань здійснюється загальний аналіз продуктивності мобільного застосунку за допомогою вбудованого інструменту Android Studio – Profiler [21]. Отримані результати наведено на рисунку 3.14.



Рисунок 3.14 – Результат тестування продуктивності мобільного застосунку

Наступним етапом є тестування екрану вибору даних для аналізу за допомогою штучного інтелекту. У випадку, якщо користувач намагається згенерувати поради без зазначення назви, система виведе повідомлення про помилку та запропонує ввести назву, оскільки це є обов’язковою умовою для продовження роботи застосунку (рис. 3.15).

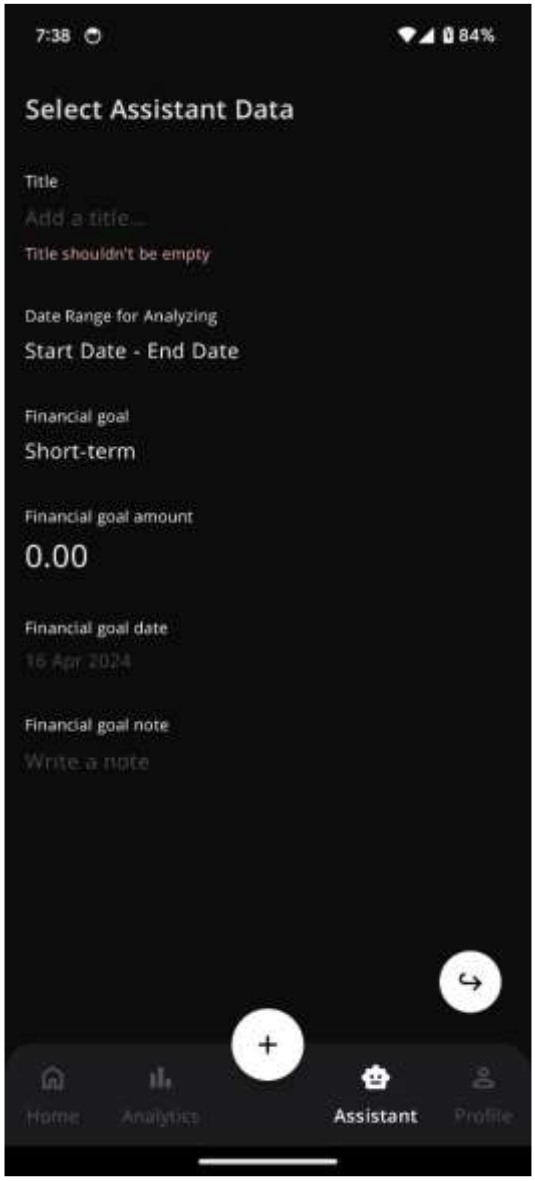


Рисунок 3.15 – Процес генерувати поради без зазначення назви

Аналогічна валідація передбачена також для таких полів, як вибір часових рамок аналізу (рис. 3.16), зазначення суми фінансової цілі (рис. 3.17) та вибір бажаної дати досягнення цієї цілі (рис. 3.18).

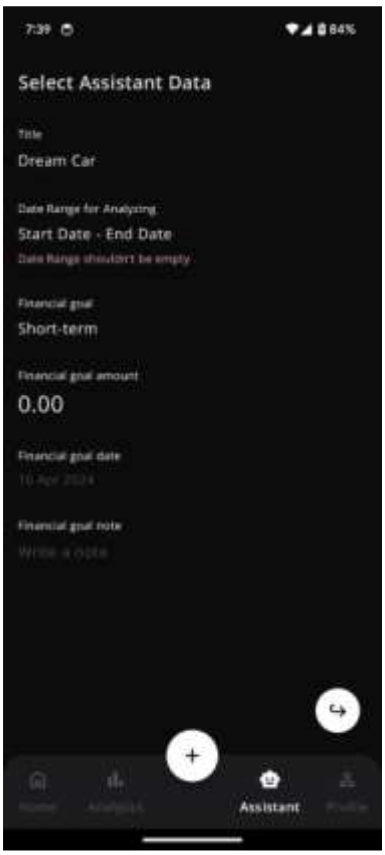


Рисунок 3.16 – Процес генерації порад у проміжку часу

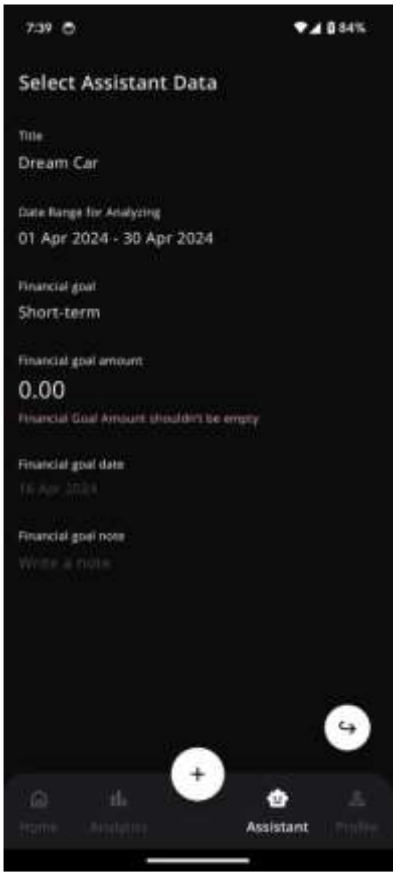


Рисунок 3.17 – Процес генерації порад без фінансової цілі

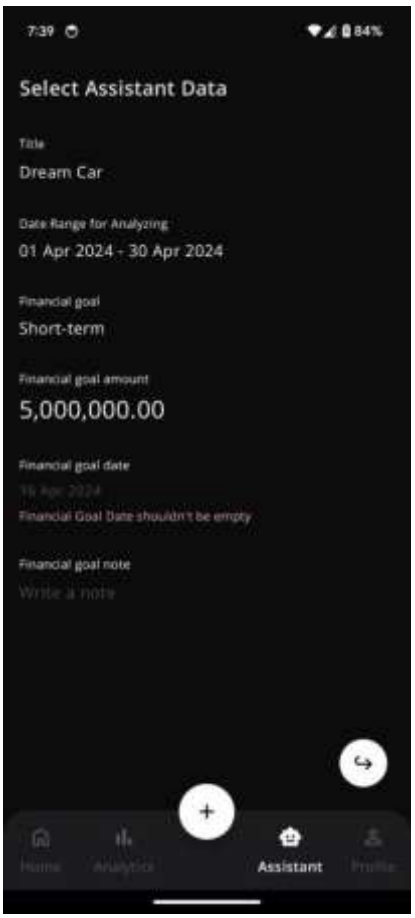


Рисунок 3.18 – Процес генерації порад без часового проміжку

Після запуску мобільного застосунку користувачу необхідно додати певну кількість транзакцій, на основі яких буде здійснено формування рекомендацій. Після генерації порад користувач отримує доступ до детальної фінансової статистики та графічного представлення своїх витрат і доходів. Для формування рекомендацій за допомогою штучного інтелекту необхідно заповнити відповідні поля з інформацією для аналізу та пройти авторизацію в системі. Без входу в обліковий запис взаємодія з моделлю штучного інтелекту є неможливою.

У результаті проведеного тестування програмного забезпечення мобільного застосунку з елементами штучного інтелекту для аналізу особистих фінансів було перевірено коректність валідації полів введення, збереження даних, а також функціональність взаємодії з моделлю ШІ.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра було розроблено мобільний інтелектуальний застосунок, основним призначенням якого є аналіз особистих фінансів користувача з використанням технологій штучного інтелекту. Розв'язання поставленої задачі вимагало всебічного підходу до проєктування системи, починаючи з архітектурного планування і закінчуючи розробкою та тестуванням усіх ключових модулів.

На першому етапі було проведено аналіз сучасних методів управління персональними фінансами та вивчено можливості інтеграції інтелектуальних систем у мобільні програмні продукти. Це дало змогу сформувати обґрунтовані вимоги до функціональності застосунку та визначити оптимальну архітектуру, засновану на поділі на шари даних – "data" та "domain" – з чітким розмежуванням ролей у зберіганні, обробці та відображенні інформації.

В рамках реалізації програмної частини було створено низку функціональних модулів, серед яких – модуль керування користувацькими транзакціями, система авторизації та реєстрації, графічний модуль візуалізації витрат, а також модуль взаємодії зі штучним інтелектом. Важливою особливістю розробки стало впровадження механізмів валідації введених даних, що гарантує цілісність і достовірність збереженої інформації. Всі транзакції проходять попередню перевірку перед збереженням у локальній базі даних, що підвищує надійність роботи системи.

Особлива увага приділялася інтерактивності та зручності інтерфейсу користувача. Реалізовано можливість перегляду детальної статистики, побудови графіків, а також отримання індивідуальних порад щодо оптимізації бюджету або досягнення фінансових цілей. Усі поради формуються на основі заданих користувачем параметрів, включаючи часовий інтервал, суму цілі та бажану дату її досягнення.

Під час завершального етапу роботи було проведено комплексне тестування застосунку, яке охоплювало як перевірку логіки взаємодії з базою даних, так і правильність роботи алгоритмів штучного інтелекту. Результати

тестування підтвердили стабільність і працездатність програмного продукту, а також коректність реалізованих механізмів перевірки даних та виводу результатів.

У підсумку, поставлені в роботі цілі були досягнуті повністю. Розроблений застосунок демонструє практичну ефективність запропонованого підходу до аналізу особистих фінансів із використанням сучасних інформаційних технологій та штучного інтелекту. Він може бути застосований у повсякденному користуванні для підвищення фінансової грамотності та покращення планування бюджету, а також слугувати основою для подальшого вдосконалення та розширення функціональних можливостей.

СПИСОК ЛІТЕРАТУРИ

1. Artificial Intelligence (AI) and ChatGPT: detailed history and timelines [Електронний ресурс] – Режим доступу до ресурсу: https://www.officetimeline.com/blog/artificial-intelligence-ai-and-chatgpt-history-and-timelines

2. The Implications of AI in the Mobile App Industry [Електронний ресурс] – Режим доступу до ресурсу: https://medium.com/@sandeepsrivastav_38514/the-airevolution-in-mobile-app-development-trends-and-predictions-for-2024-3e3e26f8e788

3. Педосенко А.Ю. Розробка програмного рішення для аналізу та оптимізації особистих фінансів на основі моделей штучного інтелекту / А.Ю. Педосенко, Катєльніков Д.І. // Матеріали VII Міжнародної науково-практичної конференції підрозділів Вінницького національного технічного університету, Вінниця, 2024. [Електронний ресурс] – Режим доступу до ресурсу: https://conferences.vntu.edu.ua/index.php/fiip/fiip2024/paper/view/20163

4. How AI is Supercharging Money Management [Електронний ресурс] – Режим доступу до ресурсу: https://intive.com/insights/how-ai-is supercharging-money-management

5. wally [Електронний ресурс] – Режим доступу до ресурсу: https://play.google.com/store/apps/details?id=com.kinpos.wallytech&hl=en&gl=US

6. Emma [Електронний ресурс] – Режим доступу до ресурсу: https://play.google.com/store/apps/details?id=com.emmaprod&hl=en&gl=US

7. AI Money Manager [Електронний ресурс] – Режим доступу до ресурсу: https://play.google.com/store/apps/details?id=com.expense.tracker.gpt

8. Expense Tracker [Електронний ресурс] – Режим доступу до ресурсу: https://play.google.com/store/apps/details?id=com.AI.expense.spending.tracker&hl=en_US

9. The pros and cons of using a framework [Електронний ресурс] – Режим доступу до ресурсу: https://devm.io/programming/pros-cons-usingframework-135901-001

10. Domain layer in Android [Електронний ресурс] – Режим доступу до ресурсу: https://developer.android.com/topic/architecture/domain-layer

11. Mapping of Data Objects? [Електронний ресурс] – Режим доступу доресурсу: https://www.baeldung.com/kotlin/data-objects

12. Все про UML-діаграми [Електронний ресурс] – Режим доступу до ресурсу: https://evergreens.com.ua/ua/articles/uml-diagrams.html

13. Atomic Kotlin Bruce Eckel and Svetlana Isakova. 2021 – 636 p. ISBN for EPUB version: 978-0-9818725-4-4

14. Android UI Development with Jetpack Compose: Bring declarative and native UIs to life quickly and easily on Android using Jetpack Compose. / Thomas Künneth. 2022 – 248 ст.

15. Advantages of Chat GPT [Електронний ресурс] – Режим доступу до ресурсу: https://www.appmatics.com/en/blog/vorteile-nachteile-chat-gpt

16. Save data in a local database using Room [Електронний ресурс] – Режим доступу до ресурсу: https://developer.android.com/training/datastorage/room

17. About SQLite [Електронний ресурс] – Режим доступу до ресурсу: https://www.sqlite.org/about.html

18. Jetpack Compose UI App Development Toolkit [Електронний ресурс] – Режим доступу до ресурсу: https://developer.android.com/develop/ui/compose

19. Fundamentals of testing Android apps [Електронний ресурс] – Режим доступу до ресурсу: https://developer.android.com/training/testing/fundamentals

20. Automate UI tests [Електронний ресурс] – Режим доступу до ресурсу: https://developer.android.com/training/testing/instrumented-tests/ui-tests

21. Profile your app performance [Електронний ресурс] – Режим доступу до ресурсу: https://developer.android.com/studio/profile

22. Jemerov D., Isakova S. Kotlin in Action. – Shelter Island : Manning Publications, 2017. – 360 с.

23. Hunt K., McLaughlin M. Kotlin Programming: The Big Nerd Ranch Guide. – Boston : Pearson Education, 2019. – 480 с.

24. Nagy M., Somogyi A. Kotlin Programming Cookbook. – Birmingham : Packt Publishing, 2019. – 372 с.

25. Milosovic I. Mastering Kotlin. – Birmingham : Packt Publishing, 2017. – 280 с.

26. Ali B. Jetpack Compose UI Development. – Birmingham : Packt Publishing, 2021. – 400 с.

27. Rydl B. Building Android UIs with Jetpack Compose. – Boston : O'Reilly Media, 2021. – 256 с.

28. OpenAI. ChatGPT Technical Report [Електронний ресурс]. – Режим доступу: https://openai.com/research/gpt-4 – Дата звернення: 10.05.2025.

29. Raj A. ChatGPT for Developers. – New York : Apress, 2023. – 180 с.

30. Dey N. Practical AI with ChatGPT and Python. – Birmingham : Packt Publishing, 2023. – 250 с.

31. Tennyson B. Prompt Engineering for ChatGPT [Електронний ресурс]. – Режим доступу: https://www.promptingguide.ai – Дата звернення: 10.05.2025.

32. Browne M. Exploring GPT: From Chatbots to Applications. – San Francisco : No Starch Press, 2024. – 200 с.

33. Vohra D. Firebase Essentials. – Birmingham : Packt Publishing, 2017. – 310 с.

34. Firebase Documentation [Електронний ресурс]. – Режим доступу: https://firebase.google.com/docs – Дата звернення: 10.06.2025.

35. Jangid R. Firebase for Android Development. – Independently Published, 2020. – 280 с.

36. Nayak A. Firebase Cookbook. – Birmingham : Packt Publishing, 2019. – 352 с.

37. Kumar A. Cloud Firestore in Firebase [Електронний ресурс]. – Режим доступу: https://firebase.google.com/docs/firestore – Дата звернення: 10.05.2025.

38. Android Developers. Room Persistence Library [Електронний ресурс]. – Режим доступу: https://developer.android.com/jetpack/androidx/releases/room – Дата звернення: 10.05.2025.

39. Banerjee S. Android Architecture Components. – Birmingham : Packt Publishing, 2018. – 256 с.

40. Späth S. Android Programming with Kotlin for Beginners. – Birmingham : Packt Publishing, 2019. – 676 с.

41. John A. Mastering Android Jetpack Architecture Components. – New York : Apress, 2020. – 420 с.

ДОДАТКИ