МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

Факультет менеджменту

Кафедра економічної кібернетики, комп'ютерних наук
та інформаційних технологій

Кваліфікаційна наукова
праця на правах рукопису

РОМБАК Василь Петрович

УДК 621.392.71:004.451.1(043.2)

# КВАЛІФІКАЦІЙНА РОБОТА

# РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ АВТОМАТИЗАЦІЇ ПЛАНУВАННЯ ОСОБИСТОГО БЮДЖЕТУ

Спеціальність 122 «Комп'ютерні науки»
Галузь знань – 12 «Інформаційні технології»

Подається на здобуття освітнього ступеня «Бакалавр»

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

Ромбак В.П

Науковий керівник: Крайній Володимир Олексійович,
кандидат економічних наук,в.о доцента

Завідувач кафедри: Тищенко Світлана Іванівна,
кандидат педагогічних наук, доцент

Миколаїв–2025

# АНОТАЦІЯ

Ця кваліфікаційна робота присвячена розробці мобільного додатку для автоматизації процесу планування особистого бюджету. В умовах стрімкого розвитку цифрових технологій питання ефективного управління фінансами набуває особливої актуальності, адже фінансова грамотність є ключовим чинником стабільного особистого добробуту. Метою дослідження є створення зручного, інтуїтивно зрозумілого та функціонального інструменту, який допоможе користувачам контролювати власні доходи та витрати, а також формувати грамотний підхід до планування бюджету.

У ході роботи було проведено ґрунтовний аналіз предметної області, зібрано та систематизовано функціональні та нефункціональні вимоги до системи. Спроєктовано архітектуру мобільного додатку, створено структуру бази даних для зберігання фінансової інформації та реалізовано основну програмну логіку. Окрему увагу приділено розробці користувацького інтерфейсу відповідно до сучасних принципів UX/UI-дизайну, що забезпечує зручність і доступність використання.

У результаті реалізовано повнофункціональний мобільний додаток, який дозволяє ефективно вести облік особистих фінансів, переглядати статистику витрат, формувати звіти та встановлювати фінансові цілі. Розробка має практичну цінність та сприяє підвищенню фінансової грамотності користувачів.

# ABSTRACT

This qualification work is dedicated to the development of a mobile application for automating the process of personal budget planning. In the context of rapid digital technology advancement, effective financial management becomes increasingly relevant, as financial literacy is a key factor in achieving personal well-being. The main objective of the study is to create a convenient, intuitive, and functional tool that helps users monitor their income and expenses, as well as adopt a thoughtful approach to budgeting.

The research involved a thorough analysis of the subject area, the collection and systematization of functional and non-functional system requirements. The architecture of the mobile application was designed, a database structure for storing financial information was created, and the core program logic was implemented. Particular attention was paid to the development of the user interface in accordance with modern UX/UI design principles, ensuring ease of use and accessibility.

As a result, a fully functional mobile application was developed, allowing users to effectively track their personal finances, view spending statistics, generate reports, and set financial goals. The project has practical value and contributes to improving users' financial literacy.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

| | | |
|---|---|---|
| с. | – | сторінка |
| р. | – | рік |
| рр. | – | роки |
| рис. | – | рисунок |

**ЗМІСТ**

**АНОТАЦІЯ** ..... **2**

**ABSTRACT** ..... **3**

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ** ..... **4**

**ВСТУП** ..... **6**

**РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ** ..... **9**

1.1 Поняття особистого бюджету та фінансового планування ..... 9

1.2 Існуючі методики та інструменти управління особистими фінансами ..... 11

1.3 Аналіз наявних програмних рішень (огляд аналогів) ..... 16

**РОЗДІЛ 2 ПРОЄКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ АВТОМАТИЗАЦІЇ ОСОБИСТОГО БЮДЖЕТУ** ..... **21**

2.1 Вибір інструментів розробки (мови програмування, фреймворки, середовище) ..... 21

2.2 Функціональні та нефункціональні вимоги до додатку ..... 28

2.3 UI/UX-дизайн інтерфейсу користувача ..... 32

**РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ “МОЇ ФІНАНСИ”** ..... **38**

3.1. Вибір середовища розробки та постановка задачі ..... 38

3.2. Розробка функціоналу додатку ..... 40

3.3. Тестування, оцінка ефективності та перспективи розвитку додатку ..... 46

**ВИСНОВКИ** ..... **50**

**СПИСОК ЛІТЕРАТУРИ** ..... **53**

**ДОДАТКИ** ..... **55**

**Код розробленого додатку** ..... **55**

# ВСТУП

– ***Актуальність теми*** Сьогодні цифрові технології активно проникають у повсякденне життя, зокрема у сферу управління особистими коштами. Зростання кількості транзакцій та фінансових інструментів створює потребу в простих та зручних мобільних рішеннях для контролю бюджету.

Серед великої кількості мобільних додатків, присвячених фінансам, не всі відповідають потребам сучасного користувача - багато з них мають складний інтерфейс, обмежений функціонал або відсутність адаптації до локального ринку. Розробка мобільного додатку для автоматизації планування особистого бюджету є актуальним завданням, яке потребує комплексного підходу, врахування потреб користувачів, сучасних підходів до UI/UX-дизайну та використання ефективних технологій розробки.

Актуальність створення такого додатку також зумовлена зростанням фінансової грамотності населення та бажанням користувачів мати під рукою зручний інструмент для аналізу доходів, витрат, заощаджень і планування фінансових цілей. Розробка ефективного мобільного додатку дозволяє користувачам краще розуміти свої фінансові звички, приймати обґрунтовані рішення та досягати стабільності у фінансовому житті.

– ***Мета дослідження:*** Метою дослідження є створення простого, функціонального мобільного застосунку, який би дозволив користувачу ефективно планувати особистий бюджет. Для цього проаналізовано сучасні технології, обрано інструменти розробки та протестовано готовий продукт.

– ***Завдання дослідження:*** Для досягнення поставленої мети були визначені наступні завдання:

Проаналізувати існуючі підходи до розробки мобільних додатків для управління особистими фінансами, з урахуванням сучасних технологічних тенденцій.

Визначити основні вимоги до функціональності та дизайну мобільного додатку, які забезпечать його зручність, ефективність та безпеку.

Розробити концепцію мобільного додатку, включаючи архітектуру, навігацію та інтерфейс користувача відповідно до актуальних стандартів і потреб користувачів.

Виконати технічну реалізацію додатку з використанням сучасних технологій мобільної розробки (наприклад, Flutter, Kotlin, Swift).

Провести тестування розробленого додатку з метою виявлення та усунення можливих недоліків, а також оцінити його ефективність за допомогою ключових показників зручності та продуктивності.

– ***Об'єкт дослідження:*** Об'єктом дослідження є процес розробки мобільного додатку для автоматизації планування особистого бюджету, що включає різні етапи - від планування функціоналу та дизайну до технічної реалізації й тестування. Особливу увагу приділено вивченню взаємодії користувачів з додатком, зручності використання та захисту персональних даних.

– ***Предмет дослідження:*** Предметом дослідження є конкретні методи та технології, що застосовуються при розробці мобільного додатку, включаючи інструменти для створення користувацького інтерфейсу, управління базами даних, збереження та обробки фінансової інформації, а також забезпечення продуктивності та безпеки роботи застосунку.

– Методи дослідження Для досягнення поставленої мети були використані як специфічні методи дослідження, такі як моделювання, тестування, аналіз взаємодії користувача з інтерфейсом, так і загальнонаукові методи, включаючи аналіз і синтез, індукцію і дедукцію, абстрагування, узагальнення та системне проєктування. Використання цих методів дозволило отримати достовірні наукові висновки та сформувати практичні рішення щодо розробки ефективного мобільного додатку.

– ***Інформаційна база дослідження:*** Інформаційна база дослідження включає нормативно-правові акти, наукові та навчально-методичні джерела, публікації у фахових вітчизняних та закордонних виданнях, матеріали з офіційної документації технологій мобільної розробки, а також актуальні ресурси спеціалізованих веб-сайтів, форумів і статистичних порталів. Також було

проаналізовано огляди та порівняльні дослідження популярних фінансових мобільних додатків.

– ***Практичне значення результатів дослідження:*** Практичне значення дослідження полягає у створенні функціонального мобільного додатку, який може використовуватись широким колом користувачів для ефективного управління особистими фінансами. Результати роботи можуть бути корисними розробникам, підприємцям, студентам та всім, хто цікавиться підвищенням фінансової грамотності за допомогою цифрових рішень.

– ***Апробація одержаних результатів***: Результати дослідження неодноразово обговорювались у процесі виконання роботи під час консультацій із науковим керівником. Одержані напрацювання мають прикладне значення та можуть бути використані для підготовки наукових тез, участі в студентських конференціях і публікацій у фахових виданнях. Робота має потенціал до подальшої апробації в академічному середовищі, зокрема в контексті розвитку мобільних технологій для автоматизації фінансового планування.

– ***Структура та обсяг роботи:*** кваліфікаційна робота складається з вступу, анотації, трьох розділів, висновків, списку використаних джерел і додатків. Загальний обсяг роботи становить 59 сторінок. Робота містить 3 таблиці, 13 рисунків, 3 додатки. Список використаних джерел налічує 30 найменувань.

# РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ

## 1.1 Поняття особистого бюджету та фінансового планування

Особистий бюджет є основою фінансової стабільності кожної людини. Його грамотне планування дозволяє не лише контролювати поточні витрати, але й досягати довгострокових цілей, уникати фінансових труднощів у майбутньому та ефективно розпоряджатися наявними ресурсами. Управління особистим бюджетом є не просто фінансовим інструментом, а важливою складовою життєвої стратегії, що відображає цінності, пріоритети та стиль життя конкретної особи.

Загалом, **особистий бюджет складається з трьох ключових елементів: доходів, витрат і заощаджень**. Доходи включають усі фінансові надходження, які особа отримує протягом певного періоду часу — це може бути заробітна плата, прибуток від підприємницької діяльності, пасивний дохід з інвестицій або державна допомога. Витрати, у свою чергу, охоплюють усі щоденні й періодичні фінансові зобов’язання: оплата житла, харчування, медичні витрати, освіта, транспорт, дозвілля тощо. Заощадження та інвестиції становлять ту частину доходів, яка відкладається або спрямовується на довгострокові фінансові цілі, як-от купівля нерухомості, пенсійне забезпечення чи навчання.

Правильне балансування між цими компонентами дозволяє уникати перевитрат, підтримувати позитивний грошовий потік і створювати фінансову подушку безпеки. Така подушка є необхідною умовою для впевненості у завтрашньому дні: згідно з рекомендаціями фінансових експертів, вона має становити щонайменше 3–6 місячних витрат родини. Її наявність дозволяє зберігати стабільність навіть у разі втрати доходу чи форс-мажорних ситуацій.

Окрім того, ефективне управління бюджетом сприяє зниженню рівня стресу, підвищенню впевненості у завтрашньому дні та формуванню звички раціонального споживання. У сучасному світі, де рівень споживчого тиску надзвичайно високий, здатність контролювати свої витрати стає ключовим елементом емоційного благополуччя.

Для досягнення ефективності у фінансовому плануванні необхідно дотримуватися певних **принципів**:

- **Регулярність** — важливо систематично переглядати бюджет та вносити до нього зміни відповідно до життєвих обставин: зростання або зменшення доходів, нові потреби, зовнішні економічні фактори (інфляція, зміни курсу валюти тощо).
- **Реалістичність** — бюджет має базуватися на точних і обґрунтованих даних. Переоцінка можливостей призводить до нездійсненних очікувань і подальших розчарувань.
- **Гнучкість** — у випадку непередбачених витрат або змін у пріоритетах бюджет повинен легко адаптуватися.
- **Точний облік усіх витрат** — включаючи дрібні покупки, які непомітно можуть перетворитися на значну статтю витрат. Наприклад, щоденна покупка кави за 50 грн у підсумку за місяць коштує понад 1 500 грн.
- **Чіткість фінансових цілей** — короткострокові (купівля техніки, відпустка) та довгострокові (накопичення на житло, пенсія).

Планування бюджету має низку переваг. Воно сприяє підвищенню фінансової грамотності, формує усвідомлений підхід до витрат, допомагає уникнути імпульсивних покупок і, зрештою, забезпечує досягнення поставлених фінансових цілей. Наприклад, за даними опитування Національного банку України (2023), 72% респондентів, які ведуть бюджет хоча б у спрощеній формі, відзначили зниження випадкових витрат.

Наявність чіткої стратегії також дозволяє створити резервний фонд, який може стати критично важливим у разі непередбачених ситуацій, таких як втрата роботи, хвороба, криза на ринку чи військовий конфлікт. Особливо це актуально в умовах сучасної нестабільності — зокрема, повномасштабної війни в Україні та спричинених нею соціально-економічних потрясінь.

У контексті стрімкого розвитку цифрових технологій особливої актуальності набувають мобільні застосунки для фінансового планування. Вони автоматизують процес обліку, забезпечують зручну візуалізацію даних, інтегруються з банківськими сервісами та дозволяють у режимі реального часу аналізувати поточний фінансовий стан користувача. Деякі сучасні сервіси навіть використовують елементи штучного інтелекту для прогнозування витрат та рекомендацій щодо заощаджень.

Прикладом таких рішень є застосунки **MINT**, **YANB** і **Pocket Guard**, що дають змогу класифікувати витрати, встановлювати ліміти, отримувати сповіщення про перевищення бюджету, а також бачити динаміку змін у вигляді графіків.

Варто зазначити, що особистий бюджет є не лише індивідуальним інструментом, а й відображенням загального рівня економічної культури населення. Масове впровадження цифрових інструментів для планування бюджету може сприяти не лише зростанню фінансової грамотності, а й загальній стабільності на макроекономічному рівні. За оцінками Світового банку, країни з високим рівнем особистої фінансової відповідальності демонструють вищу стійкість до економічних криз.

Саме тому вивчення, розробка та впровадження ефективних засобів управління особистими фінансами, зокрема **мобільних застосунків**, є надзвичайно актуальним і затребуваним напрямом у сучасному інформаційному суспільстві. Вони допомагають не лише спростити планування бюджету, а й стимулюють зміну фінансової поведінки населення, роблячи її більш структурованою, відповідальною та орієнтованою на довгостроковий добробут.

### 1.2 Існуючі методики та інструменти управління особистими фінансами

Управління особистими фінансами є багатогранним процесом, що охоплює низку методик та підходів, спрямованих на ефективне планування доходів, витрат і заощаджень. У сучасному світі, де фінансова грамотність стає невід’ємною

складовою повсякденного життя, існує чимало перевірених стратегій ведення особистого бюджету, а також цифрових інструментів, що автоматизують цей процес.

### Традиційні методи фінансового планування

Серед найпоширеніших класичних методів вирізняються кілька підходів, які здобули популярність завдяки своїй простоті та ефективності.

Один із найвідоміших і практичних підходів до управління особистими фінансами — це **метод 50/30/20**, який передбачає раціональний розподіл щомісячного доходу на три основні категорії. Згідно з цим методом, **50% доходу** рекомендовано спрямовувати на **обов'язкові витрати**, до яких належать комунальні платежі, оренда житла, продукти харчування, транспорт та інші базові потреби. **30% коштів** пропонується використовувати на **особисті потреби та розваги**, такі як походи в кіно, кафе, хобі або інші непершочергові, але важливі для психологічного комфорту витрати. Нарешті, **20% доходу** доцільно **відкладати у вигляді заощаджень або інвестувати** - наприклад, у депозит, пенсійний фонд, цінні папери чи інші інструменти довгострокового збереження капіталу.

*Рис. 1.1 Ілюстрація правила розподілу бюджету 50/30/20; Джерело:ucap.io*

Такий підхід сприяє формуванню **фінансової дисципліни**, дозволяє уникати імпульсивних витрат і поступово створювати фінансову «подушку безпеки». Крім того, він є універсальним і може адаптуватися до різного рівня доходів, оскільки головне — це дотримання пропорцій, а не абсолютних сум. Метод 50/30/20 допомагає зберігати **баланс між поточним споживанням та довгостроковим накопиченням**, що є важливим чинником фінансової стабільності, особливо в умовах економічної нестабільності чи інфляційних коливань.

Інший ефективний підхід до управління особистими фінансами — це **метод “конвертів”**, який передбачає **жорсткий поділ бюджету на окремі категорії витрат із чітко визначеними лімітами**. Суть методу полягає в тому, що кожна категорія витрат (наприклад, харчування, транспорт, одяг, розваги тощо) отримує свою **встановлену суму грошей**, яка фізично або цифрово розміщується у відповідному “конверті”. У традиційному варіанті використовують паперові

конверти з готівкою, тоді як у сучасних умовах - окремі рахунки, мобільні додатки або підкатегорії у фінансових трекерах.

Основна перевага цього підходу полягає у візуалізації меж доступних коштів: як тільки вміст певного конверта вичерпано, витрати за відповідною категорією припиняються або відкладаються до наступного місяця. Це сприяє формуванню фінансової дисципліни, допомагає уникати перевитрат і дозволяє краще усвідомлювати, куди саме спрямовуються кошти. Метод "конвертів" особливо корисний для людей, схильних до імпульсивних покупок або тих, хто лише починає формувати навички бюджетування. У цифрову епоху ця методика зберігає свою актуальність завдяки адаптації у численних мобільних застосунках, які автоматизують розподіл та контроль витрат за категоріями.

Більш структурованим є **метод "нульового бюджету"**, де кожна одиниця доходу має своє призначення - витрати, заощадження чи інвестиції. Мета такого підходу - забезпечити максимальну ефективність використання кожної гривні, уникаючи необґрунтованих витрат.

### Сучасні цифрові інструменти для управління бюджетом

З розвитком цифрових технологій дедалі більше людей звертаються до мобільних застосунків та програмного забезпечення, які дозволяють автоматизувати процес планування фінансів і значно спрощують аналіз бюджету.

**Мобільні застосунки для фінансового планування** стали зручним інструментом для щоденного обліку. Серед найпопулярніших:

- **Mint** - один з найвідоміших сервісів, що автоматично синхронізується з банківськими рахунками, дозволяє аналізувати витрати та створювати персоналізовані бюджети.
- **YNAB (You Need a Budget)** - застосунок, побудований на принципі "призначення завдання кожному долару", спрямований на уникнення боргів та поступове досягнення фінансових цілей.

- **PocketGuard** - платформа, що допомагає визначити, скільки коштів залишилось для витрат, аналізуючи поточні витрати і надаючи поради щодо оптимізації бюджету.

| Метод | Основна ідея | Переваги | Недоліки |
|---|---|---|---|
| Метод 50/30/20 | Розподіл бюджету: 50% – на потреби, 30% – на бажання, 20% – на заощадження | Простий у використанні, підходить для початківців | Не враховує індивідуальних обставин, жорстке співвідношення |
| Метод «Конвертів» | Розподіл готівки або бюджету по категоріях у вигляді “конвертів” | Візуальний контроль витрат, формує фінансову дисципліну | постійного обліку та оновлення, менш ефективний у цифровому середовищі |
| Нульовий бюджет | Кожен долар має конкретне призначення | Висока ефективність, максимальний контроль над фінансами | Вимагає багато часу для ведення та планування |

*Рис. 1.1; Джерело: сформовано автором*

Крім мобільних додатків, користувачі активно використовують **фінансові планувальники та калькулятори**, які дозволяють оцінювати обсяг необхідних заощаджень, планувати великі покупки або розраховувати пенсійні внески. Наприклад, **пенсійні калькулятори** враховують інфляцію, очікувану дохідність та рівень доходу для формування довгострокового фінансового плану.

Ще одним поширеним інструментом залишаються **електронні таблиці (Excel, Google Sheets)**. Вони дозволяють вручну створювати адаптовані під індивідуальні потреби бюджети, здійснювати моніторинг динаміки витрат, будувати графіки та фінансові звіти на основі історичних даних.

### Переваги цифрових рішень у фінансовому плануванні

Цифрові інструменти мають низку переваг, які роблять процес управління фінансами більш точним, зручним і гнучким. Насамперед, це **автоматизація** - сучасні застосунки здатні самостійно фіксувати транзакції, аналізувати витрати та

формувати звіти без участі користувача. Важливою функцією є **інтеграція з банківськими рахунками**, що забезпечує актуальність даних у реальному часі.

Також варто відзначити **доступність** - мобільні додатки дають змогу керувати бюджетом у будь-який час і з будь-якого місця. Окрім цього, вони пропонують **інтерактивні візуалізації** - графіки, діаграми, повідомлення про перевищення лімітів, що сприяє кращому розумінню фінансових звичок і мотивує до змін.

Загалом, комбінація класичних методик та цифрових рішень створює гнучку та ефективну систему управління особистими фінансами, яку кожен користувач може адаптувати до власних потреб та життєвих обставин.

### 1.3 Аналіз наявних програмних рішень (огляд аналогів)

У сучасному цифровому середовищі автоматизація особистого фінансового планування стала ключовим чинником досягнення стабільності та фінансового добробуту. З метою розробки ефективного мобільного додатку для управління особистим бюджетом доцільно здійснити аналіз існуючих рішень на ринку. Це дозволить не лише виявити переваги та недоліки конкурентних продуктів, але й визначити функціонал, який варто інтегрувати в майбутній додаток.

Один із найпопулярніших інструментів у сфері особистих фінансів - мобільний додаток **Mint**, розроблений компанією Intuit. Цей застосунок здобув широку популярність серед користувачів завдяки поєднанню зручного інтерфейсу, глибокої інтеграції з банківськими сервісами та наявності функцій для детального аналізу особистих фінансів. Mint дозволяє не лише контролювати поточний стан рахунків, а й здійснювати комплексний моніторинг витрат, доходів, заощаджень та фінансових зобов’язань.

Серед ключових функціональних можливостей додатку -автоматична синхронізація з банківськими рахунками, кредитними картками, позиками, інвестиційними рахунками та іншими фінансовими джерелами. Завдяки цьому

користувач отримує актуальну інформацію в режимі реального часу. Додаток також пропонує створення персоналізованих бюджетів з урахуванням індивідуальних витрат, автоматичну категоризацію транзакцій, візуалізацію даних у вигляді діаграм і графіків, а також повідомлення про перевищення лімітів або майбутні платежі.

Серед переваг Mint -безкоштовне використання основного функціоналу, автоматичне оновлення фінансової інформації, можливість налаштування сповіщень та зручна аналітика витрат за категоріями. Окрім того, інтеграція з різними банківськими установами дозволяє централізувати управління фінансами в одному інтерфейсі, що особливо зручно для користувачів з кількома рахунками.

Проте, незважаючи на свою функціональність, Mint має і певні недоліки. До них можна віднести обмежені можливості персоналізації бюджету, відсутність повної підтримки користувачів за межами США та Канади, а також потенційні ризики, пов’язані із зберіганням та передачею конфіденційної фінансової інформації через сторонні сервери.

**YNAB (You Need A Budget)** — це популярний фінансовий додаток, який позиціонує себе не просто як інструмент обліку витрат, а як повноцінну методологію фінансового мислення. Основна ідея, яку просуває платформа, полягає в принципі "дай кожному долару роботу" (*Give every dollar a job*), тобто в необхідності розподіляти кожен отриманий дохід між конкретними категоріями витрат ще до здійснення реальних транзакцій. Такий підхід дозволяє формувати чіткий план управління фінансами, підвищити фінансову дисципліну та уникати імпульсивних витрат.

Однією з ключових переваг YNAB є його навчальна спрямованість: додаток супроводжується великою кількістю покрокових інструкцій, відеоуроків та порад, що дозволяють навіть новачкам поступово зануритися у процес планування бюджету. Крім того, платформа підтримує створення коротко- і довгострокових фінансових цілей, а також передбачає гнучке налаштування категорій бюджету

відповідно до індивідуальних потреб користувача. Кожна витрата відстежується в реальному часі, що забезпечує високий рівень контролю та прозорості.

Водночас, YNAB має і певні недоліки. По-перше, додаток є повністю платним - вартість щорічної підписки становить близько 99 доларів США, що може стати суттєвим бар'єром для користувачів з обмеженим бюджетом. По-друге, початкове налаштування сервісу може здатися досить складним і заплутаним для нових користувачів, оскільки вимагає ретельного аналізу всіх фінансових потоків та системного підходу. Також інтерфейс YNAB менш інтуїтивний порівняно з конкурентами, що потребує певного часу для адаптації.

**PocketGuard** — це зручний мобільний додаток для управління особистими фінансами, який орієнтований насамперед на швидкий та легкий контроль за поточним фінансовим станом користувача. Головна концепція сервісу полягає в тому, щоб допомогти людині в будь-який момент часу точно знати, скільки грошей вона може витратити без шкоди для обов'язкових платежів і заощаджень. Цей підхід реалізується через ключову функцію додатку -**"In My Pocket"**, яка відображає доступну для витрат суму з урахуванням регулярних доходів, фіксованих витрат, підписок, боргових зобов'язань та запланованих заощаджень.

Однією з головних переваг PocketGuard є його простий, інтуїтивно зрозумілий інтерфейс, який робить процес користування додатком максимально комфортним навіть для тих, хто не має досвіду у фінансовому плануванні. Автоматичне підключення до банківських рахунків дозволяє додатку в реальному часі аналізувати транзакції, розподіляти витрати по категоріях і створювати візуалізації витрат. Також користувач може встановлювати ліміти на окремі категорії витрат, що допомагає уникати перевитрат і тримати бюджет під контролем.

Попри низку переваг, PocketGuard має й певні обмеження. Зокрема, функціонал додатку більше орієнтований на поточний облік, тому відсутні розширені інструменти для довгострокового планування бюджету чи фінансових

цілей. Крім того, можливості кастомізації -тобто створення власних категорій, підкатегорій або цілей є обмеженими, що може не задовольнити користувачів з більш складною фінансовою структурою.

**Порівняння основних функцій**

| Функція | Mint | YNAB | PocketGuard |
|---|---|---|---|
| Синхронізація з банками | Так | Так | Так |
| Можливість створення бюджету | Так | Так | Так |
| Аналіз витрат по категоріях | Так | Обмежений | Так |
| Створення фінансових цілей | Так | Так | Так |
| Рекомендації по заощадженням | Безкоштовно | Обмежено | Так |
| Платність | Безкоштовно | Платно | Безкоштовно |

*Рис. 1.3; Джерело: сформовано автором*

**Висновки**

Проведений аналіз популярних мобільних додатків для управління особистими фінансами дозволяє зробити низку важливих висновків. Зокрема, Mint і YNAB надають користувачам широкий набір інструментів для глибокого занурення у процес фінансового планування. Ці сервіси дозволяють не лише фіксувати витрати й доходи, але й прогнозувати фінансові потоки, формувати довгострокові цілі, аналізувати динаміку змін у бюджеті. Такий функціонал особливо корисний для користувачів, які прагнуть повного контролю над своїми фінансами та готові витратити час на налаштування й вивчення можливостей системи. Водночас, варто зазначити, що YNAB має вищий поріг входу через складність первинної конфігурації, а також обмежений доступ через платну підписку.

PocketGuard, навпаки, орієнтований на користувачів, які шукають прості рішення для швидкої оцінки фінансового стану. Його функціонал забезпечує базовий рівень обліку та автоматичного аналізу витрат, що робить його

доступним навіть для новачків. Проте обмежена кастомізація та відсутність інструментів для довгострокового планування звужують сферу його застосування.

Загалом, проведений аналіз дає змогу чітко зрозуміти потреби сучасних користувачів у сфері особистого фінансового менеджменту. При розробці власного додатку доцільно поєднати зручний інтерфейс, характерний для PocketGuard, із розширеними можливостями планування, притаманними Mint і YNAB. Такий підхід забезпечить баланс між функціональністю та простотою використання.

# РОЗДІЛ 2 ПРОЄКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ АВТОМАТИЗАЦІЇ ОСОБИСТОГО БЮДЖЕТУ

## 2.1 Вибір інструментів розробки (мови програмування, фреймворки, середовище)

Розробка мобільного додатку передбачає ухвалення низки важливих технічних рішень, які безпосередньо впливають на ефективність, стабільність, швидкість розробки та зручність подальшої підтримки проєкту. Одним із ключових аспектів на початковому етапі є вибір відповідних інструментів: мов програмування, фреймворків, середовищ розробки, а також технологій для збереження й обробки даних. До основних критеріїв, якими слід керуватись при прийнятті таких рішень, належать кросплатформеність, продуктивність, зручність роботи з інтерфейсом, швидкість розробки, доступність документації, активність спільноти розробників і легкість масштабування.

### Мови програмування та фреймворки

Серед основних мов програмування, які використовуються у сфері мобільної розробки, слід відзначити **Kotlin**, **Swift**, **Dart** та **JavaScript/TypeScript**.

**Kotlin** — це сучасна та зручна мова програмування, рекомендована Google для розробки AnPodroid-додатків. Розроблена компанією JetBrains, вона вперше з’явилася у 2011 році, а вже у 2017 отримала офіційну підтримку від Google, ставши ключовим інструментом у створенні сучасних Android-додатків.Kotlin повністю сумісна з Java, що дозволяє легко інтегрувати існуючі Java-бібліотеки та фреймворки, а також поступово мігрувати існуючі проєкти на Kotlin без необхідності повного переписування коду. Це створює комфортні умови для команд розробників і забезпечує плавний перехід на нову технологію.

Однією з головних переваг Kotlin є її висока продуктивність та безпека. Мова включає розширені можливості контролю типів, які допомагають уникнути типових помилок, наприклад, NullPointerException, що часто зустрічаються у Java.

Kotlin має підтримку корутин - спеціального механізму для роботи з асинхронним кодом, який спрощує обробку фонового виконання задач без блокування основного потоку, що особливо важливо для мобільних додатків.

Kotlin тісно інтегрується з Android Studio - основним середовищем розробки для Android. Android Studio забезпечує підтримку автоматичного доповнення коду, рефакторингу, налагодження, UI-інспекції та багатьох інших інструментів, які значно полегшують процес створення, тестування і підтримки застосунків. Крім того, Kotlin підтримує мультиплатформену розробку - можливість створення кросплатформених застосунків, де бізнес-логіка кодується один раз і працює як на Android, так і на iOS, що дозволяє суттєво зекономити ресурси.

**Swift - основна мова програмування для розробки під платформу iOS.** Вона була представлена компанією Apple у 2014 році як сучасна, безпечна та продуктивна альтернатива Objective-C. Swift вирізняється високою швидкодією завдяки сучасним механізмам оптимізації коду, що дозволяє створювати швидкі та ефективні додатки для пристроїв Apple.

Однією з важливих переваг Swift є її безпечний і зрозумілий синтаксис, що спрощує розробку і знижує кількість типових помилок, зокрема пов’язаних з роботою з пам’яттю. Мова також має потужне типобезпечне середовище та систему опціональних значень (optionals), яка допомагає уникати помилок, пов’язаних із null-значеннями.

Swift має добре документоване та постійно оновлюване API, що надає розробникам широкі можливості для створення інтерфейсів користувача, роботи з мережею, базами даних, графікою та іншими аспектами мобільних додатків. Swift підтримує функціональне, об'єктно-орієнтоване і протокольно-орієнтоване програмування, що робить її гнучкою для розробки як невеликих, так і великих складних проєктів.

Розробка на Swift здійснюється переважно у середовищі **Xcode**, яке є офіційним інструментом Apple для створення iOS, iPadOS, watchOS та macOS

застосунків. Проте, використання Swift та Xcode вимагає операційної системи **macOS**, що накладає певні обмеження для розробників, які працюють на інших платформах. Це ускладнює одночасну розробку та підтримку Android-версії застосунку, адже для Android зазвичай використовується Kotlin або Java в Android Studio на Windows чи Linux.

Через цю особливість розробники часто стикаються з викликами під час реалізації кросплатформених проєктів, коли потрібно підтримувати і iOS, і Android, що потребує або окремих команд, або використання додаткових кросплатформених фреймворків (наприклад, Flutter або React Native), які підтримують Swift чи Kotlin як частину свого стека.

Для реалізації кросплатформених мобільних додатків набагато ефективнішим є використання мови програмування **Dart** у поєднанні з фреймворком **Flutter** від компанії **Google. Flutter** дозволяє створювати нативоподібні застосунки одночасно під Android і iOS за допомогою єдиного базового коду, що значно скорочує час розробки та знижує витрати на підтримку проєкту. Це особливо важливо для стартапів та команд із обмеженими ресурсами, адже немає необхідності розробляти окремі версії для кожної платформи.

Однією з ключових переваг **Flutter** є підтримка функції **«гарячого перезапуску» (hot reload)** - швидкого оновлення інтерфейсу користувача без повного перезапуску застосунку. Це дозволяє розробникам оперативно вносити зміни, тестувати їх і миттєво бачити результат, що значно підвищує ефективність роботи.

Flutter пропонує велику кількість готових **віджетів** (widgets), що реалізують стандартні компоненти інтерфейсу, включно з підтримкою **Google Material Design** та **Apple Cupertino** стилю. Окрім цього, фреймворк дозволяє повністю кастомізувати зовнішній вигляд додатків, створюючи унікальні та привабливі UI/UX-рішення.

Крім технічних можливостей, **Flutter** відзначається активним розвитком і має потужну глобальну спільноту розробників. Це забезпечує регулярне оновлення інструментів, появу нових бібліотек і плагінів, а також багату документацію, що спрощує навчання та впровадження технології.

Завдяки таким властивостям **Dart** і **Flutter** стали одним із найпопулярніших виборів для кросплатформеної мобільної розробки, поєднуючи високу продуктивність, гнучкість та зручність використання.

Ще одним популярним підходом до кросплатформеної розробки є використання мов програмування **JavaScript** або **TypeScript** у поєднанні з фреймворком **React Native.** React Native базується на бібліотеці React, яка широко застосовується для створення веб-інтерфейсів, і дозволяє розробляти мобільні додатки з використанням знайомих веб-технологій. Це відкриває можливості для веб-розробників швидко адаптуватися до мобільної розробки без необхідності вивчати нові мови програмування.

Однією з ключових переваг React Native є **низький поріг входу** для розробників, які вже володіють JavaScript або TypeScript, що дозволяє швидко запускати проєкти та скорочує час навчання команди. Крім того, React Native має розвинену екосистему плагінів та модулів, що розширюють функціональність додатків, а також дозволяють взаємодіяти з нативними API операційних систем.

Проте, цей підхід має й певні обмеження. Однією з проблем є **складність реалізації складних анімацій** та графічних ефектів, які часто потребують додаткової оптимізації або написання частини коду на нативних мовах (Java/Kotlin для Android, Swift/Objective-C для iOS). Також React Native іноді потребує значних ресурсів для підтримки високої продуктивності, що може бути викликом для складних або ресурсомістких додатків.

Ще одним фактором є **залежність від зовнішніх бібліотек**, які не завжди повністю сумісні між платформами або регулярно оновлюються. Це може

призводити до технічних боргів, збільшувати складність підтримки проєкту та потребувати частого оновлення коду.

В цілому, React Native залишається популярним вибором для швидкої розробки кросплатформених мобільних додатків, особливо коли важлива швидкість виходу на ринок і наявність команди з веб-експертизою.

| Категорія | Інструмент | Переваги | Недоліки |
|---|---|---|---|
| Мови програмування | Kotlin / Swift | Висока продуктивність, повна підтримка платформи | Вимагає розробки окремо для кожної платформи |
| Мови програмування | Flutter (Dart) | Один код для Android та iOS, потужний UI, велика спільнота | Великі розміри додатку, потреба в адаптації |
| Мови програмування | React Native | Швидка розробка, багато плагінів | Низька продуктивність у складних проєктах |
| Бази даних | SQLite | Швидкий доступ офлайн, легко інтегрується | Локальне збереження, не підтримує хмарну синхронізацію |
| Бази даних | Firebase | Хмарне збереження, автоматична синхронізація, аналітика | Залежність від підключення до Інтернету |
| IDE / Середовище | Android Studio | Повна підтримка Android, інтеграція з Firebase | Важка IDE, потребує багато ресурсів |
| IDE / Середовище | VS Code | Легка IDE, підтримка Flutter / React Native | Потребує налаштування для емуляції |

*Рис. 2.1;Джерело: сформовано автором*

**Середовище розробки**

Середовище розробки відіграє ключову роль у забезпеченні зручності створення, налагодження та тестування мобільного додатку. Воно об’єднує всі необхідні інструменти для написання коду, перевірки його коректності, оптимізації продуктивності та подальшої публікації застосунку.

Для платформи **Android** найпопулярнішим і найбільш функціональним інтегрованим середовищем розробки (IDE) є **Android Studio** - офіційний продукт компанії Google. Android Studio базується на платформі IntelliJ IDEA і підтримує потужну систему збірки **Gradle**, що дозволяє гнучко керувати процесом компіляції, залежностями та конфігураціями проєкту. Серед важливих інструментів Android Studio - зручний **візуальний редактор інтерфейсу**, який допомагає швидко створювати UI-компоненти, **UI-інспектор** для аналізу та

оптимізації графічного інтерфейсу, а також **емулятор пристроїв** з широкими можливостями налаштувань, що дозволяє тестувати додатки на різних віртуальних смартфонах та планшетах. Крім того, Android Studio пропонує **інструменти профілювання продуктивності**, що допомагають відстежувати споживання пам'яті, використання процесора та енергії, що особливо важливо для мобільних застосунків.

Для розробки з використанням кросплатформених фреймворків, таких як **Flutter** або **React Native**, часто обирають **Visual Studio Code (VS Code)** - легке, швидке та гнучке середовище розробки, яке підтримує численні **плагіни та розширення**. VS Code сумісний з багатьма мовами програмування і має вбудовану інтеграцію з популярними системами контролю версій, такими як **Git**, що полегшує спільну роботу над проєктами. Завдяки можливостям роботи з терміналом, налагодженням (debugging) і автодоповненням коду VS Code стає зручним інструментом для мобільних розробників, особливо в умовах кросплатформеної розробки.

Для створення додатків під **iOS** обов'язковим є використання середовища **Xcode** - офіційного IDE від Apple. Xcode включає всі необхідні інструменти для написання коду, компіляції, налагодження, тестування на реальних пристроях та емуляторах, а також публікації додатків у **App Store**. Серед ключових можливостей Xcode - підтримка Swift і Objective-C, потужний графічний редактор інтерфейсів (Interface Builder), інтеграція з системою контролю версій, а також інструменти для аналізу продуктивності та автоматизованого тестування.

Таким чином, правильний вибір середовища розробки значно впливає на ефективність роботи розробників і якість кінцевого продукту, а також забезпечує комфортний робочий процес від початкового написання коду до його запуску і підтримки.

**Бази даних і хмарні сервіси**

Окрему увагу під час розробки мобільного додатку слід приділити вибору та реалізації технологій зберігання даних, які забезпечують надійність, швидкодію і зручність доступу до інформації.

Для підтримки **офлайн-доступу** до даних оптимальним рішенням є використання **SQLite** - легкої вбудованої реляційної бази даних, що інтегрується безпосередньо у мобільний додаток. SQLite не потребує запуску окремого серверу чи складної налаштувальної інфраструктури, оскільки всі дані зберігаються локально у вигляді файлу на пристрої користувача. Це робить SQLite ідеальним вибором для додатків із невеликою або середньою кількістю даних, де потрібно швидко виконувати операції з читання, запису, оновлення та видалення інформації. Завдяки своїй легкості та широкій підтримці SQLite забезпечує стабільну і продуктивну роботу навіть на пристроях з обмеженими ресурсами.

Для реалізації **хмарної синхронізації даних**, а також для забезпечення комплексного функціоналу, що включає **авторизацію користувачів, обробку push-повідомлень, аналітику та багатокористувацьку взаємодію**, доцільно застосовувати сервіс **Firebase** від компанії Google. Firebase представляє собою набір інтегрованих хмарних сервісів, які значно спрощують розробку сучасних мобільних і веб-застосунків.

Зокрема, для зберігання та синхронізації даних у реальному часі Firebase пропонує два основні варіанти баз даних:

- **Realtime Database** — це хмарна NoSQL база даних, що забезпечує миттєве оновлення даних у всіх клієнтів, які підключені до проєкту. Вона добре підходить для застосунків із постійною комунікацією між користувачами, таких як чати, ігри або системи сповіщень.
- **Cloud Firestore** — більш сучасна NoSQL база даних із покращеною масштабованістю, гнучкішими запитами та кращою інтеграцією з іншими сервісами Firebase.

Firebase також підтримує широкі можливості **аутентифікації** користувачів - за допомогою email та пароля, соціальних мереж (Google, Facebook, Twitter), а також анонімного входу. Це дозволяє швидко інтегрувати безпечну систему авторизації без необхідності розробляти власний бекенд.

Додатково Firebase надає інструменти для **відправлення push-повідомлень** (Firebase Cloud Messaging), аналітики використання застосунку (Firebase Analytics), тестування продуктивності, управління функціями додатку і багато іншого.

Завдяки своїй комплексності, Firebase є потужним і гнучким рішенням для розробників, які прагнуть створити сучасний, інтерактивний і масштабований мобільний додаток із мінімальними затратами на інфраструктуру.

### 2.2 Функціональні та нефункціональні вимоги до додатку

У процесі проєктування мобільного додатку для автоматизації особистого бюджету одним із ключових етапів є чітке та всебічне формулювання вимог до майбутнього програмного продукту. Від правильності, деталізації та повноти цього етапу залежить не лише якість та стабільність реалізації системи, а й рівень задоволеності кінцевих користувачів. Вони очікують, що додаток буде інтуїтивно зрозумілим, зручним у користуванні, функціонально насиченим, а також стабільним та безпечним у роботі. Вимоги виступають як фундаментальні орієнтири для розробників, окреслюючи основні цілі проєкту, визначаючи межі функціональності, а також встановлюючи стандарти щодо продуктивності, безпеки, сумісності, надійності та масштабованості системи. У загальному вигляді вимоги поділяються на дві основні категорії: функціональні, що визначають, що саме має робити додаток, та нефункціональні, які описують, яким чином система повинна виконувати ці функції.

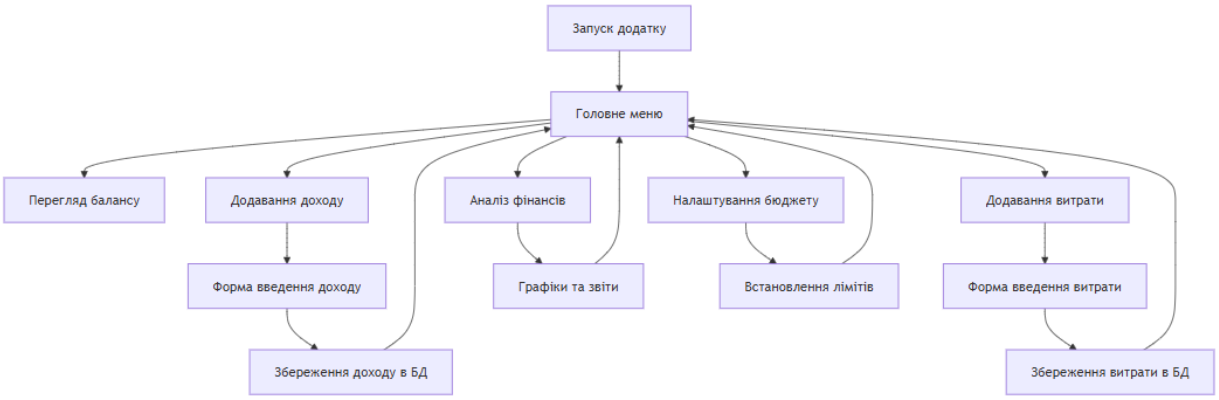


*Рис.2.2.1 Блок-схема взаємодії основних функцій мобільного додатку для планування бюджету; Джерело: сформовано автором*

**Функціональні вимоги**

Функціональні вимоги охоплюють ключові сценарії взаємодії користувача з додатком і є основою для формування архітектури програмного продукту. В першу чергу, користувач повинен мати змогу легко та швидко додавати записи про свої фінансові операції-витрати або доходи. Для кожного запису необхідно передбачити вказання суми, дати проведення операції, вибір категорії, а також можливість додати текстовий опис для детальнішої інформації. Всі записи зберігаються локально в базі даних пристрою, що дозволяє користувачу отримувати доступ до історії фінансових операцій у будь-який момент навіть без підключення до інтернету. Це підвищує автономність та надійність додатку. Важливою функцією є система категорій, що дозволяє обирати зі стандартного набору або створювати власні категорії витрат і доходів. Така гнучкість сприяє персоналізації обліку фінансів, адаптуючи додаток під індивідуальні звички користувача.

Додатково передбачено можливість редагування і видалення будь-якої транзакції, що підвищує зручність користування та допомагає уникати помилок у даних. Особливу увагу приділено функціоналу перегляду статистики. На основі

зібраних даних користувач може оцінювати динаміку своїх фінансових потоків у різні часові проміжки. Реалізовано розвинену систему фільтрації і сортування за датою, категорією чи типом операції. Результати подаються у вигляді графіків, гістограм, кругових діаграм і таблиць, що значно спрощує аналіз та прийняття усвідомлених фінансових рішень. Не менш важливою є функція планування бюджету: користувач може встановлювати ліміти витрат по окремих категоріях, наприклад, на харчування, транспорт чи розваги, на певний період часу. Додаток інформує про досягнення або перевищення встановлених лімітів за допомогою сповіщень або повідомлень у інтерфейсі, що допомагає уникати непередбачених фінансових труднощів.

Окрему увагу приділено можливості експорту та імпорту даних, що дозволяє зберегти інформацію на зовнішніх носіях або перенести її на інші пристрої. Такий функціонал підвищує мобільність і гнучкість у використанні додатку. Для полегшення ознайомлення нових користувачів передбачена інтерактивна інструкція та розділ “Питання та відповіді”, що допомагають швидко освоїтися у програмі.

### Нефункціональні вимоги

Нефункціональні вимоги визначають якість роботи системи, комфортність взаємодії користувача з додатком та надійність роботи під час різних умов експлуатації. Однією з найважливіших вимог є продуктивність: інтерфейс повинен залишатися швидким, чуйним і стабільним незалежно від обсягу даних, які обробляє додаток. Запити до бази даних, відображення графіків і оновлення інформації мають виконуватися миттєво, без видимих затримок або зависань. Для цього особлива увага приділяється оптимізації алгоритмів обробки даних, раціональному використанню оперативної пам’яті та завантаженню UI-компонентів. Надійність системи також є критичною, додаток має гарантувати збереження даних навіть у разі раптового вимкнення живлення, збою пристрою або примусового закриття програми. Всі дані повинні бути коректно збережені і відновлені при наступному запуску.

Безпека є одним із пріоритетних напрямів. Хоча додаток не використовує хмарні сервіси і не вимагає підключення до облікових записів, важливо захистити локальні дані від несанкціонованого доступу. Для цього використовуються стандартні механізми Android: обмеження доступу до внутрішнього сховища, шифрування локальної бази даних, а також у перспективі -реалізація додаткових засобів автентифікації, таких як PIN-код, біометричні дані (відбиток пальця, розпізнавання обличчя) або двофакторна автентифікація. Для безпечного відновлення інформації при перевстановленні програми передбачена функція резервного копіювання з використанням вбудованих можливостей ОС.

Інтерфейс користувача розробляється з акцентом на інтуїтивність та доступність. Навігація має бути логічною і зрозумілою, з послідовністю дій, що не викликає плутанини навіть у новачків. Використання зрозумілих піктограм, збалансованої кольорової палітри та адаптивного дизайну забезпечує комфортну роботу на пристроях з різними розмірами екранів -від компактних смартфонів до великих планшетів. Особливу увагу приділено сумісності з поширеними версіями Android, починаючи з 8-ї (Oreo) і вище, що дозволяє охопити максимально широку аудиторію.

Ще одним важливим аспектом є енергоефективність. Додаток не повинен надмірно навантажувати акумулятор пристрою або споживати ресурси у фоновому режимі без потреби, що особливо важливо для користувачів із невеликим запасом батареї. Завдяки цьому користувач може довго працювати з додатком без зайвих переживань щодо розряду пристрою.

З точки зору масштабування і розвитку, на етапі проєктування враховано можливість майбутнього розширення функціоналу без порушення архітектурної цілісності. Можливе додавання модулів для ведення спільного бюджету (для родини чи пари), створення фінансових цілей і накопичень, а також інтеграція з банківськими сервісами чи автоматичне розпізнавання регулярних платежів.

У підсумку, ретельно продумані функціональні та нефункціональні вимоги забезпечують не лише якісне технічне виконання додатку, а й формують зручний, безпечний та ефективний інструмент для керування особистими фінансами, який сприяє покращенню фінансової грамотності та дисципліни користувача.

### 2.3 UI/UX-дизайн інтерфейсу користувача

Інтерфейс користувача -один із найважливіших чинників, що визначають успіх мобільного додатку. Саме через нього користувачі формують своє перше враження, а також оцінюють зручність і ефективність взаємодії з системою. Особливо це важливо для додатків, призначених для планування особистого бюджету, де інтерфейс має поєднувати високу функціональність з простотою використання.

Розробка UI/UX-дизайну вимагає особливої уваги до створення інтуїтивної навігації, комфортного візуального сприйняття і зрозумілої логіки взаємодії. Метою дизайну є не лише привабливе оформлення, а й формування такої структури, що дозволяє користувачеві максимально швидко і без зайвих зусиль досягати поставлених завдань -вводити дані, контролювати бюджет, аналізувати статистику.

Зручність інтерфейсу підвищує лояльність користувачів, мінімізує час навчання та знижує ймовірність помилок під час роботи з додатком. Важливо, щоб дизайн відповідав сучасним стандартам і був адаптивним до різних розмірів екрану, забезпечуючи комфортну роботу як на смартфонах, так і на планшетах.

Розробка інтерфейсу має ґрунтуватися на принципах мінімалізму, зрозумілої візуальної ієрархії, єдиної стилістики, а також адаптивності до різних пристроїв. Простота дизайну дозволяє уникнути зайвого інформаційного перевантаження та допомагає користувачам зосередитися на виконанні основних дій - введенні витрат, перегляді бюджету, аналізі статистики. Інтуїтивність інтерфейсу забезпечує легке орієнтування у функціоналі навіть для користувачів, які вперше відкривають додаток, без потреби в додаткових інструкціях. Усі

елементи управління повинні мати чітке призначення, однаковий стиль та передбачувану поведінку, що формує відчуття цілісності і довіри до продукту.

Головний екран додатку виконує функцію навігаційного центру, з якого користувач отримує доступ до ключових розділів. Його структура має бути максимально лаконічною та зрозумілою. На цьому екрані доречно розміщувати поточний стан бюджету, короткий огляд витрат, кнопку для швидкого додавання нової транзакції та доступ до розділу статистики. Усі дії мають бути доступні у два-три натискання, що значно покращує користувацький досвід.

*Рис.2.3.1 Схема інтерфейсу головного екрану додатку; Джерело: Dribbble*

Особливу увагу слід приділити екрану введення витрат. Він має бути оптимізований для максимально швидкого і комфортного введення даних. У структурі цього екрана доцільно реалізувати просту форму з полями для введення суми, вибору категорії та дати. Випадаючі списки з підказками, автоматичне

збереження останніх використаних категорій, а також візуальні іконки для кожної категорії сприятимуть більш зручній і швидкій взаємодії. Варто уникати перевантаження цього екрана зайвими елементами, аби не відволікати увагу користувача від основної дії.

Інший важливий розділ — це статистика. Для аналізу фінансових звичок користувача слід реалізувати можливість перегляду звітів за різні періоди часу у вигляді кругових діаграм, стовпчикових графіків або лінійних схем. Така візуалізація дозволяє швидко оцінити, на що саме витрачаються кошти, які категорії є пріоритетними, а також які тенденції спостерігаються у фінансовому житті користувача. Дизайн графіків має бути простим, контрастним і легко читабельним. Важливо, щоб навіть при великій кількості категорій інформація залишалася доступною і не перевантаженою.

Крім основних функцій, інтерфейс має враховувати взаємодію з підказками та оповіщеннями. Наприклад, у разі перевищення встановленого ліміту витрат додаток може надсилати відповідне повідомлення. Такі повідомлення мають бути стилізовані таким чином, щоб вони були помітними, але не нав’язливими. Їхній дизайн повинен відповідати загальному стилю додатку, з використанням приємних кольорів та помірної анімації. Користувач має мати змогу налаштовувати частоту, тип і форму сповіщень, аби уникнути інформаційного перенавантаження.

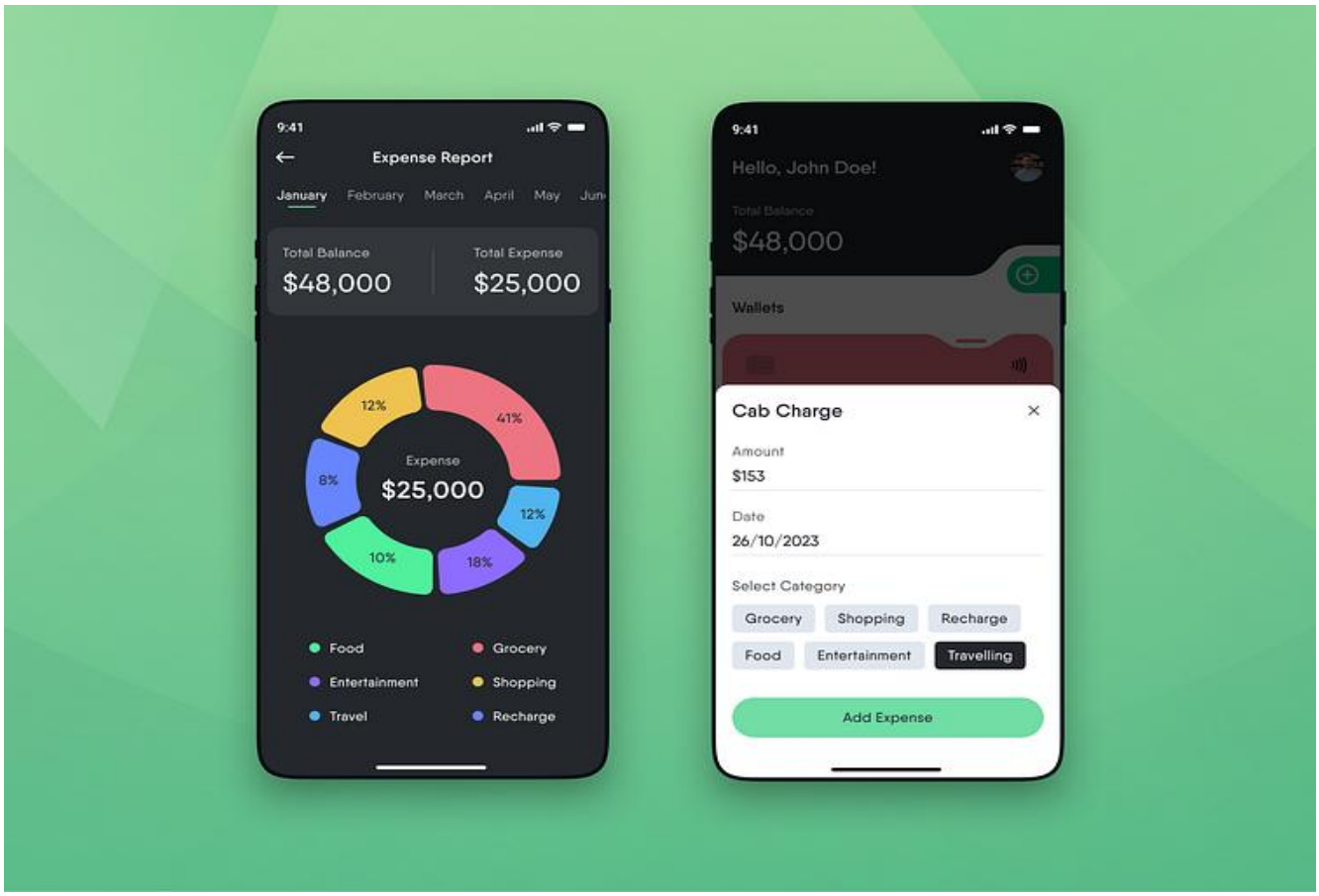


*Рис.2.3.2 Графік статистики витрат;Джерело: Dribbble*

Розділ налаштувань повинен бути зручним і зрозумілим, дозволяючи користувачеві змінювати мову інтерфейсу, валюту обліку, формат дати, а також темний або світлий режим. Наявність таких параметрів створює враження персоналізованого продукту, що адаптується до індивідуальних уподобань користувача. У перспективі, можливість змінювати колірну палітру, активувати додаткові функції або синхронізувати дані із зовнішніми сервісами може стати важливою перевагою.

Ще одним значущим елементом дизайну є вибір кольорів і шрифтів. Колірна палітра має бути спокійною та виваженою. Вона повинна викликати довіру і відповідати тематиці фінансового обліку. Рекомендується використовувати приглушені сині, зелені або сірі відтінки, які сприяють концентрації та не подразнюють очі при тривалому використанні. Шрифти повинні бути чіткими, легко читабельними, без зайвих декоративних елементів.

Розмір тексту має бути достатнім для зручного читання на екранах різної діагоналі.

Загалом, UI/UX-дизайн мобільного додатку повинен забезпечувати приємний досвід використання, сприяти ефективній взаємодії, мінімізувати кількість дій для досягнення результату та відповідати очікуванням сучасного користувача. Узгоджене поєднання візуальної естетики, функціональної логіки та технічної реалізації створює додаток, який не лише виконує свою основну задачу, а й формує позитивне емоційне ставлення, що стимулює користувача до регулярного використання.

**Висновки**

У розділі було розглянуто ключові етапи проєктування мобільного додатку для автоматизації особистого бюджету. На початку розділу проведено аналіз актуальних інструментів розробки, серед яких обрано **Android Studio** як основне середовище розробки, мову програмування **Kotlin** за її зручність і сучасність, а також бібліотеку **Jetpack Compose**, яка значно спрощує створення гнучкого й привабливого користувацького інтерфейсу.

Після вибору технологій було визначено структуру майбутнього додатку: його основні модулі, функціональні компоненти та логіку взаємодії між ними. Було приділено увагу розробці архітектури, що відповідає принципам модульності, масштабованості та розширюваності. Такий підхід дозволяє ефективно підтримувати застосунок у майбутньому та легко додавати новий функціонал без порушення цілісності системи.

Особливу увагу приділено інтерфейсу користувача, який має бути простим, інтуїтивно зрозумілим і привабливим. З цією метою спроєктовано основні екрани додатку: головна сторінка, сторінки доходів і витрат, статистика та налаштування. Також передбачено елементи зручної навігації, що сприятимуть позитивному користувацькому досвіду.

Загалом, у цьому розділі закладено чітку концептуальну та технічну основу для реалізації мобільного додатку. Обрані інструменти й архітектурні рішення забезпечують стабільність, гнучкість і зручність майбутнього програмного продукту. Усе це створює сприятливі умови для переходу до наступного етапу – реалізації та тестування функціоналу застосунку.

# РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ “МОЇ ФІНАНСИ”

## 3.1. Вибір середовища розробки та постановка задачі

Початковим етапом реалізації проєкту розробки мобільного додатку для автоматизації обліку особистих фінансів стало визначення оптимального інструментарію. Від цього вибору залежить не лише швидкість і зручність розробки, а й можливість подальшої підтримки, масштабування та інтеграції нових функцій. В умовах стрімких технологічних змін та зростання вимог користувачів це завдання набуває особливого значення.

Під час аналізу варіантів увага була зосереджена на ключових критеріях: доступності ліцензії, відкритості вихідного коду, підтримці широкого спектру мобільних пристроїв різних виробників, розвиненій екосистемі бібліотек і фреймворків, а також інтеграції інструментів для тестування і налагодження. Не менш важливими стали рівень документації і підтримка спільноти розробників, що суттєво скорочує час опанування нових технологій.

Порівнявши середовища Visual Studio з Xamarin, Flutter та Android Studio, було прийнято рішення обрати **Android Studio** -офіційне інтегроване середовище розробки мобільних додатків для платформи Android. Основні переваги цього вибору: безкоштовність і відкритий код, що важливо для навчальних і дослідницьких проєктів з обмеженим бюджетом, а також підтримка широкого спектра мобільних пристроїв -від бюджетних до флагманських моделей.

Однією з ключових переваг Android Studio є її багатофункціональність. Середовище забезпечує зручний редактор коду, потужний візуальний конструктор інтерфейсів, інтегровані засоби налагодження, емулятор пристроїв із різними конфігураціями, а також підтримку сучасних архітектурних патернів і бібліотек. Особливо корисною є можливість легкої інтеграції таких популярних бібліотек, як Room для локальної бази даних, Retrofit для мережевих запитів та Jetpack-компонентів, які спрощують управління станом додатку та його життєвим циклом.

Після визначення середовища розробки була поставлена основна задача: створити мобільний додаток, що дозволить користувачам вносити записи про доходи та витрати, переглядати історію фінансових операцій, отримувати узагальнену інформацію за різні часові проміжки, а також працювати в автономному режимі без необхідності постійного підключення до Інтернету. Таким чином, додаток повинен забезпечити зручний і ефективний інструмент для ведення особистого фінансового обліку, сприяючи підвищенню фінансової грамотності та плануванню бюджету.

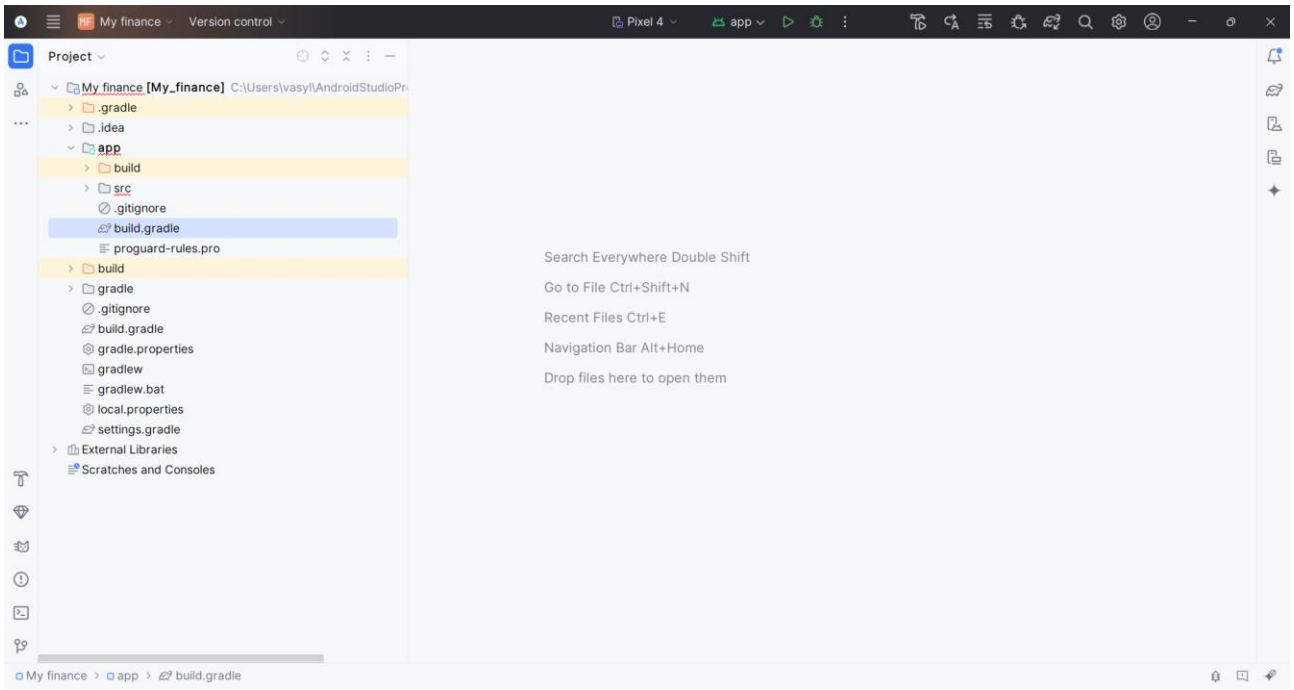


*Рис. 3.1.1 - Робоче середовище Android Studio під час створення проєкту "Мої фінанси"*

Для реалізації поставленої мети була розроблена початкова архітектура продукту, що передбачає тісну взаємодію клієнтської частини з локальною базою даних. Такий підхід гарантує незалежність роботи від зовнішніх сервісів, що забезпечує стабільність функціонування у різних умовах. Принципи модульності та розширюваності системи дозволяють у майбутньому легко додавати нові функції без суттєвих змін у структурі додатку.

На початковому етапі було створено базовий прототип інтерфейсу користувача у вигляді макетів, що відтворювали основні сценарії взаємодії. Це дозволило своєчасно виявити потенційні проблеми з навігацією та організацією інформації, що суттєво спростило подальшу розробку і тестування було обрано середовище Android Studio. Структура головного вікна додатку, що визначає контейнер для відображення окремих екранів (фрагментів) та містить панель нижньої навігації, описана у файлі activity_main.xml (див. Додаток А).

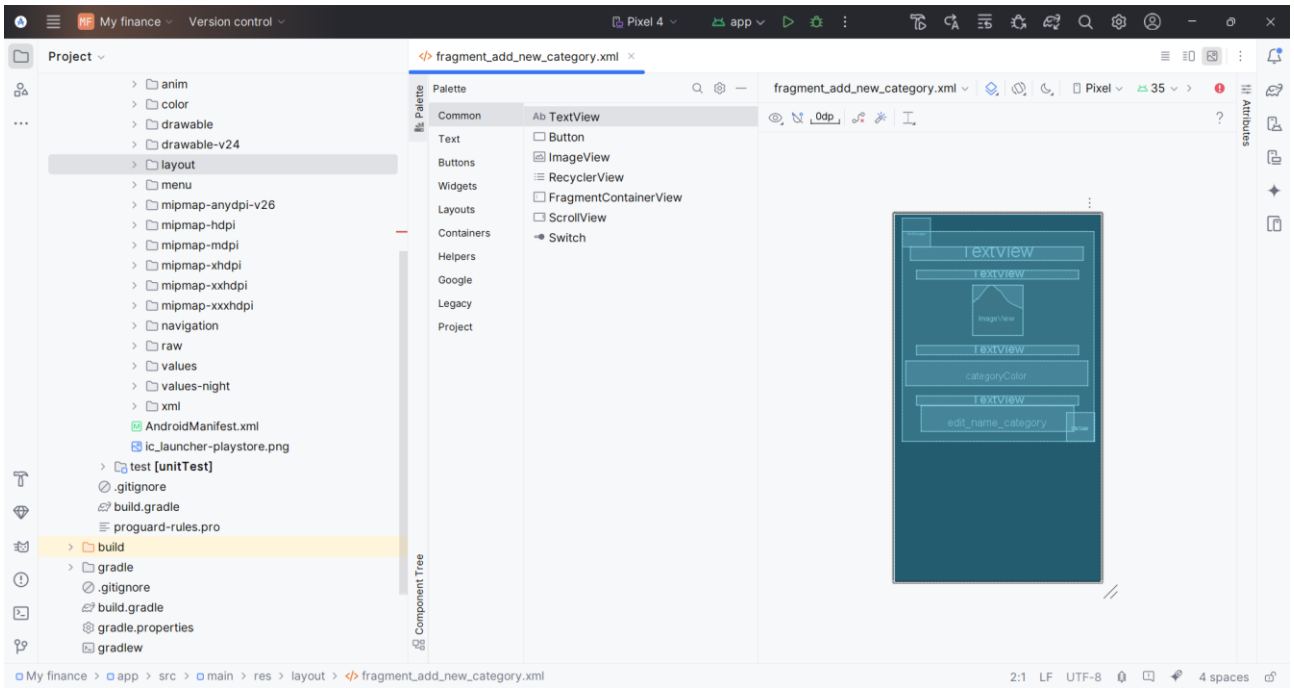


*Рис. 3.1.2 - Початкова архітектура додатку "Мої фінанси" у вигляді схеми взаємодії компонентів*

Отже, вибір Android Studio, чітка постановка задачі та продумане планування архітектури створили міцний фундамент для успішної реалізації мобільного додатку «Мої фінанси». Це забезпечить створення ефективного й зручного інструменту, який задовольнить потреби користувачів у веденні особистого фінансового обліку.

### 3.2. Розробка функціоналу додатку

Розробка функціональної частини мобільного додатку "Мої фінанси" стала ключовим етапом у реалізації проєкту та вимагала комплексного підходу до

побудови логіки роботи системи, моделювання інтерфейсу користувача та забезпечення стабільної взаємодії між окремими її компонентами. Основним завданням на цьому етапі було не лише втілення всіх попередньо визначених функціональних вимог, але й створення продукту, який би гармонійно поєднував у собі високу продуктивність, інтуїтивність використання, надійність збереження даних і сучасний адаптивний дизайн.

На початковій стадії розробки особлива увага приділялася проєктуванню інтерфейсу користувача. Головною концепцією стало створення мінімалістичного та водночас функціонального середовища, яке дозволяло б користувачеві миттєво орієнтуватися в додатку та виконувати ключові операції за мінімальну кількість дій. Головний екран був спроєктований із розрахунком на те, щоб одразу відображати загальний баланс користувача, наявні доходи та витрати, а також надавати прямий доступ до функцій додавання нової операції чи перегляду історії.

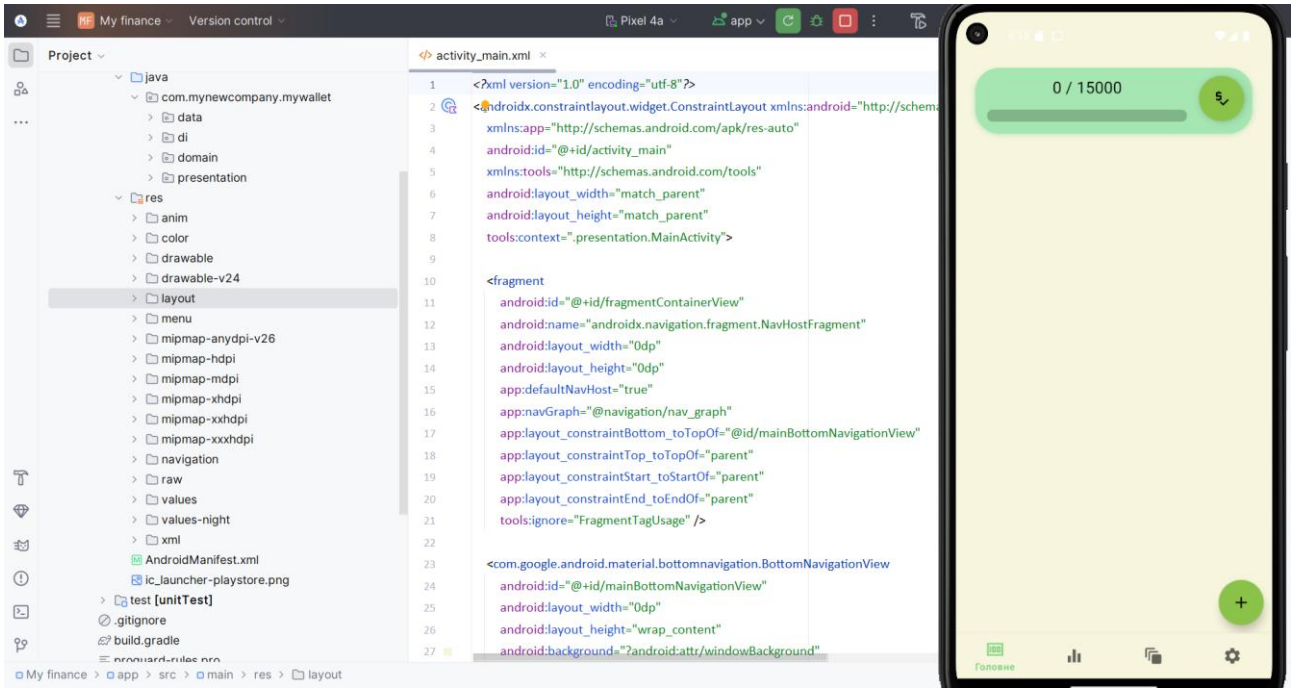


*Рис. 3.2.1 - Головний екран додатку "Мої фінанси", що демонструє загальний баланс і основні функціональні кнопки*

Основна навігація між фрагментами реалізована за допомогою **Navigation Component**, який дозволяє описати всі переходи в окремому XML-файлі. Це дало змогу централізовано керувати логікою переходів, а також передавати параметри між фрагментами без додаткової обробки у коді. (див. Додаток Б)

Далі було розроблено окремий екран для додавання нової фінансової операції. Користувач мав можливість обрати категорію витрат або доходів зі списку або створити власну, ввести суму операції, додати за потреби текстовий опис і вказати дату транзакції. Створення власних категорій стало важливою складовою персоналізації додатку, адже дозволяло кожному користувачеві адаптувати систему обліку під свої фінансові звички та потреби.

Окрім основного обліку, для користувачів, що прагнуть досягати конкретних фінансових результатів, було розроблено функціонал встановлення та відстеження цілей. Макет екрану, призначеного для введення та відображення таких цілей (див. Додаток В).

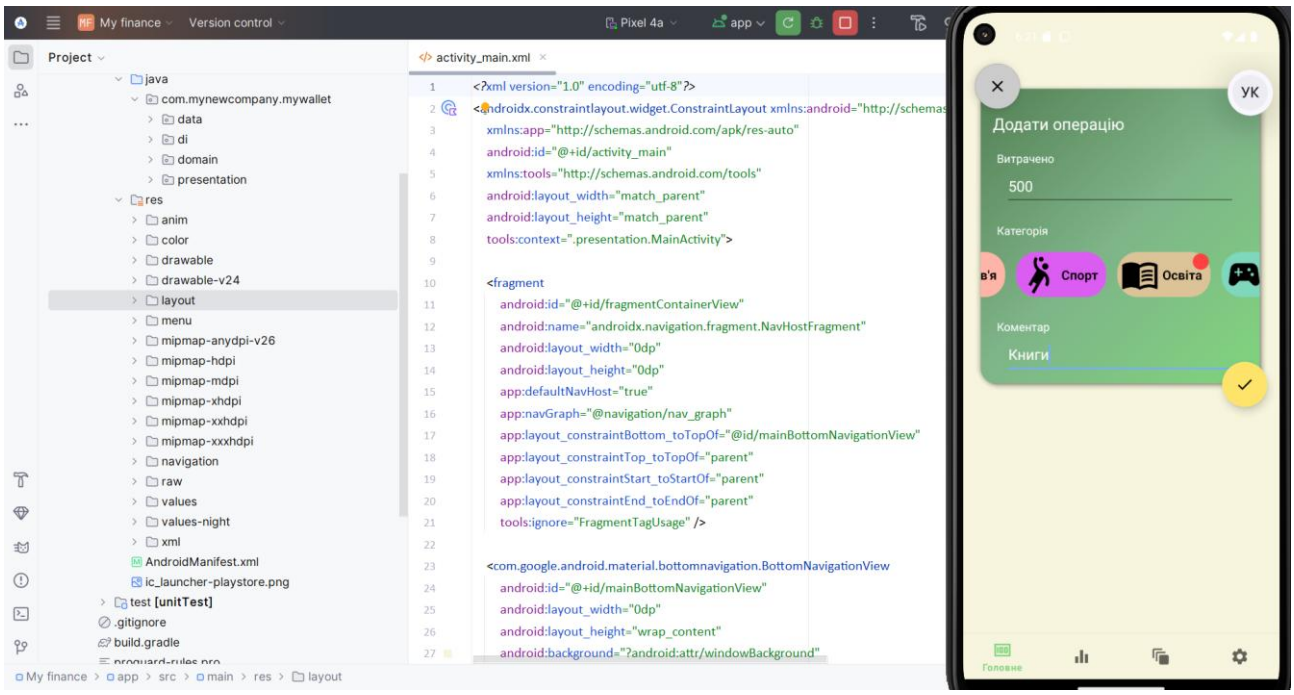


*Рис. 3.2.2 - Екран додавання нового доходу або витрати з вибором категорії та введенням суми*

Додатково було реалізовано окремий функціональний екран для редагування вже створених фінансових операцій. Користувач отримував можливість змінювати раніше введені дані: категорію витрати або доходу, суму, опис та дату транзакції. Така функціональність дозволяє виправляти помилки у внесених даних або актуалізувати інформацію у разі зміни обставин. Зокрема, збереження гнучкості в управлінні фінансовими записами сприяє більш точному обліку та формуванню достовірної статистики. Завдяки зручному інтерфейсу редагування, користувач може швидко оновити необхідну інформацію без необхідності видалення і повторного створення операції, що значно підвищує зручність користування додатком.

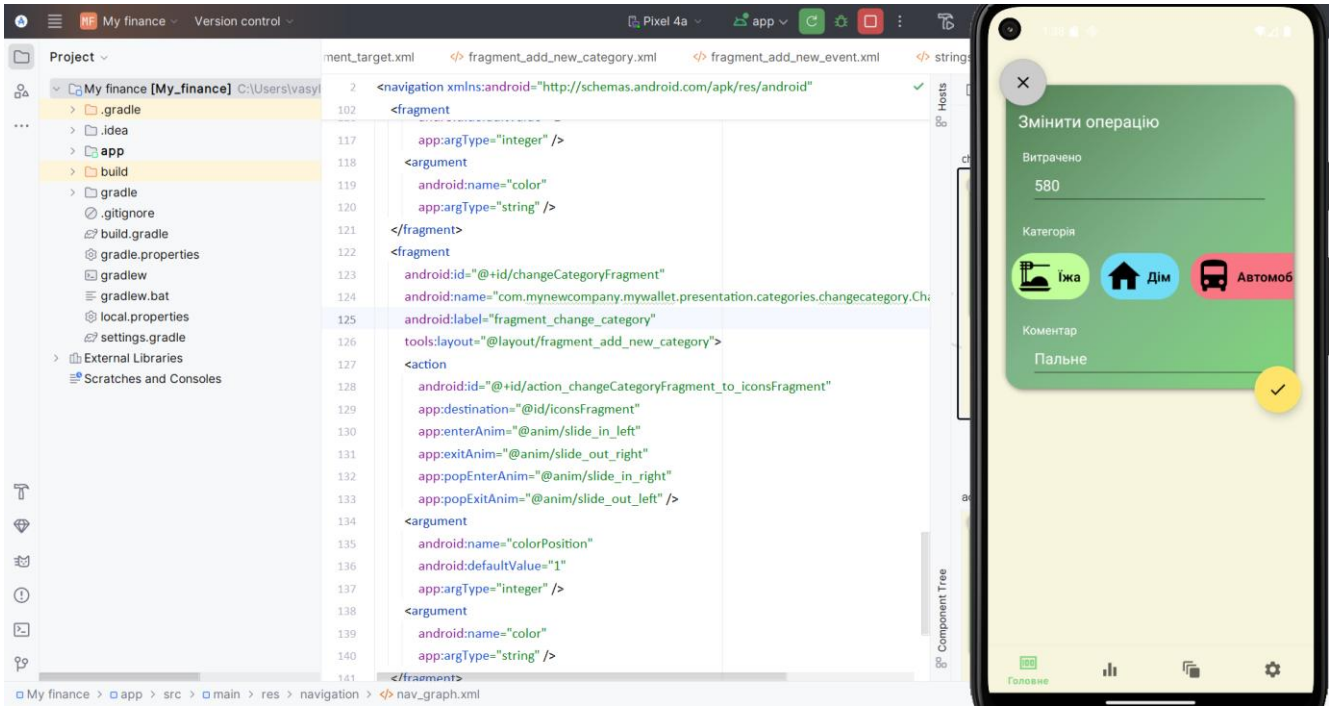


*Рис. 3.2.3 - Екран зміни доходу або витрати з вибором категорії та введенням суми*

Реалізація процесу збереження даних здійснювалася за допомогою інтеграції локальної бази даних Room. Використання саме цього інструменту було зумовлене його надійністю, високою швидкістю обробки запитів, а також можливістю організовувати збереження даних відповідно до принципів

нормалізації. Це забезпечувало ефективну структуру бази даних, мінімізуючи дублювання записів і полегшуючи подальший аналіз даних.

Щоб зробити аналіз фінансової активності більш доступним і наочним, було впроваджено модуль візуалізації статистики. Він дозволяв будувати графічні діаграми витрат і доходів за обраний період, що значно полегшувало користувачеві розуміння фінансової картини і сприяло прийняттю обґрунтованих бюджетних рішень.

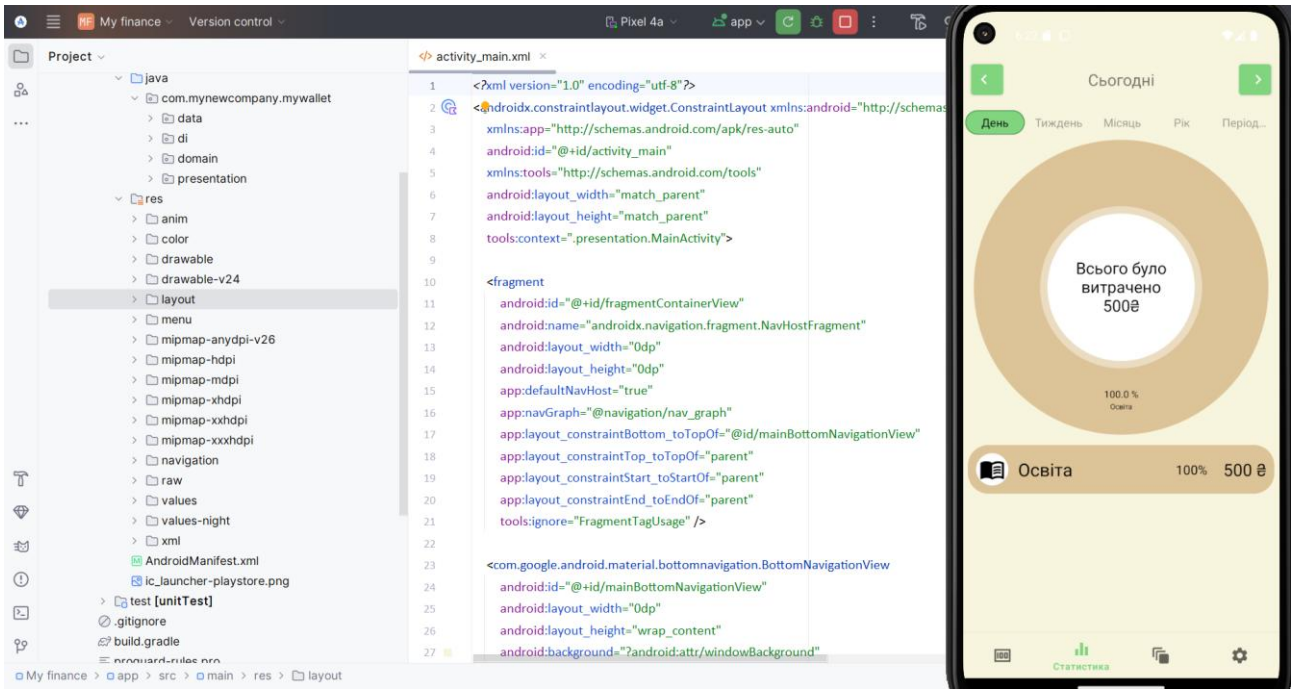


*Рис. 3.2.4 - Графічна статистика витрат та доходів за певний період часу*

Одним із викликів при розробці додатку стала необхідність забезпечення стабільної роботи додатку навіть при великій кількості збережених транзакцій. Для вирішення цієї задачі було впроваджено механізм пагінації історії операцій, що дозволяло відображати інформацію порційно та уникати перевантаження інтерфейсу. Також було оптимізовано виконання запитів до бази даних, що позитивно вплинуло на швидкість відкриття великих обсягів історичних записів.

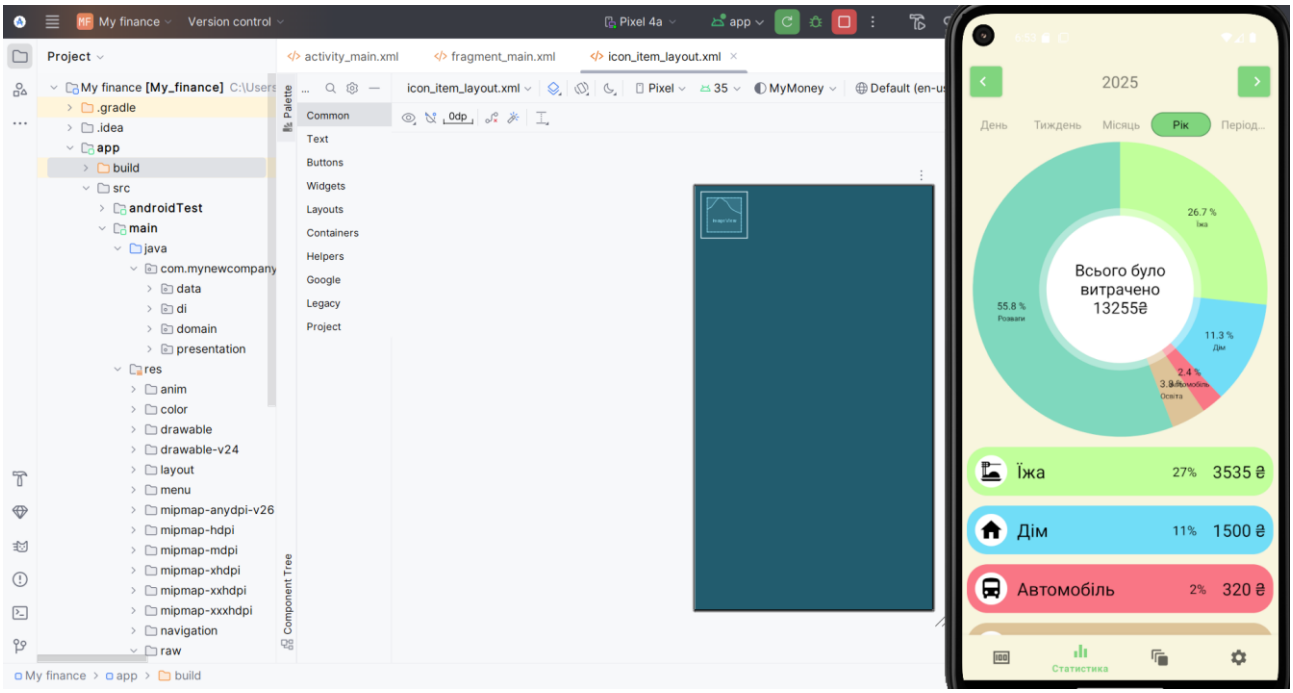


*Рис. 3.2.5 - Екран історії фінансових операцій користувача з можливістю перегляду записів*

Розробка функціоналу передбачала також врахування різноманітних характеристик мобільних пристроїв, тому інтерфейс додатку було реалізовано за принципами адаптивного дизайну. Це дозволило забезпечити його коректне відображення та стабільну роботу як на пристроях із малими екранами, так і на великих планшетах із високою роздільною здатністю.

На фінальному етапі розробки проводилося тестування юзабіліті додатку за участю реальних користувачів. Це дозволило зібрати зворотній зв'язок щодо зручності інтерфейсу, зрозумілості логіки навігації та загального враження від використання додатку. За результатами тестування були внесені деякі покращення, серед яких - оптимізація переходів між екранами, уточнення підписів кнопок і підвищення контрастності візуальних елементів для кращої читабельності.

У підсумку розроблений функціонал мобільного додатку "**Мої фінанси**" повністю відповідає поставленим завданням. Він об'єднує сучасний, легкий у використанні інтерфейс, потужний набір інструментів для обліку та аналізу

фінансів, а також стабільну та надійну архітектуру. Успішна реалізація даного етапу відкриває широкі можливості для подальшого розвитку продукту та його вдосконалення.

### 3.3. Тестування, оцінка ефективності та перспективи розвитку додатку

Після завершення основного етапу розробки мобільного додатку "**Мої фінанси**" розпочався процес всебічного тестування, що мало на меті перевірку коректності реалізації функціоналу, стабільності роботи додатку в різних умовах експлуатації, а також виявлення можливих недоліків і шляхів їх усунення. Тестування додатку відбувалося у кілька етапів, кожен з яких мав чітко окреслені цілі та завдання.

Першим етапом стало функціональне тестування, в межах якого перевірялися всі основні сценарії використання додатку. Зокрема, здійснювалася перевірка правильності виконання операцій додавання доходів та витрат, редагування записів, видалення транзакцій та формування загальної фінансової статистики. Тестування виконувалося як у нормальних умовах, так і в ситуаціях підвищеного навантаження, наприклад, при внесенні великої кількості фінансових записів за короткий проміжок часу. Усі тести показали стабільність роботи додатку та відсутність критичних помилок.

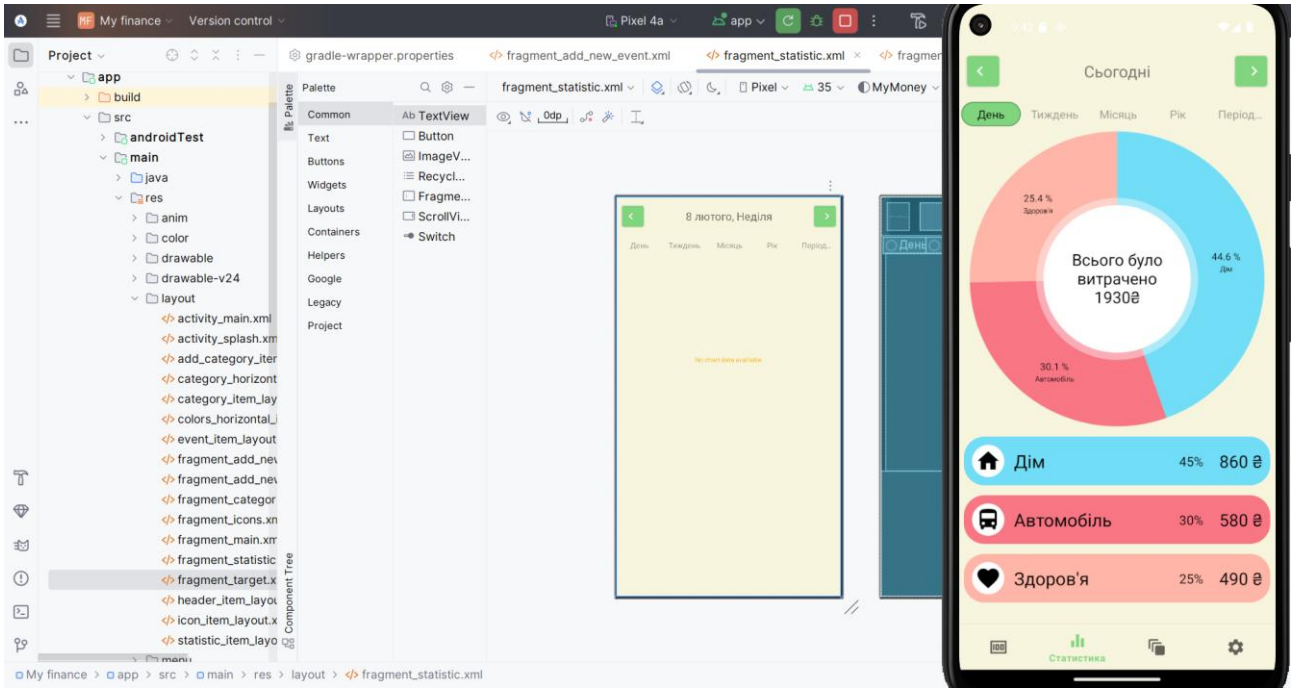

*Рис. 3.3.1 - Підсумковий інтерфейс додатку після внесення декількох фінансових записів*

Наступним етапом стало **модульне тестування**, яке передбачало окрему перевірку логіки роботи кожного з основних функціональних модулів: додавання нової транзакції, обробки даних у базі Room, формування діаграм та графіків. Кожен модуль тестувався автономно з використанням відповідних юніт-тестів, що дозволяло точно виявляти можливі недоліки на рівні окремих компонентів системи без впливу зовнішніх факторів. Для цього використовувалася бібліотека JUnit у поєднанні з інструментами перевірки бази даних (наприклад, InstantTaskExecutorRule для LiveData). Такі тести дозволили, зокрема, перевірити коректність збереження та вибірки транзакцій, правильну категоризацію витрат, а також точність розрахунків у підсумкових блоках. За результатами тестування усі основні модулі відповідали заданим технічним характеристикам, демонструючи стабільність та передбачувану поведінку в межах визначених сценаріїв.

Після цього було проведено **інтеграційне тестування**, яке дозволило оцінити якість взаємодії між різними частинами додатку, зокрема -між інтерфейсною частиною, логікою обробки запитів та базою даних. У процесі перевірки використовувалися як інструменти автоматичного тестування (наприклад, Espresso), так і ручні тести на емуляторі та реальному пристрої. Особливу увагу було приділено коректності відображення змін у даних після виконання дій користувача: додавання, редагування або видалення фінансових операцій. Наприклад, після видалення певної операції історія транзакцій автоматично оновлювалася без потреби перезапуску додатку або ручного оновлення вікна. Аналогічно, після редагування операції відповідні зміни одразу відображалися в списку транзакцій, підсумкових розрахунках та графічних діаграмах, що свідчило про правильну побудову логіки оновлення даних і передачі стану між компонентами інтерфейсу.

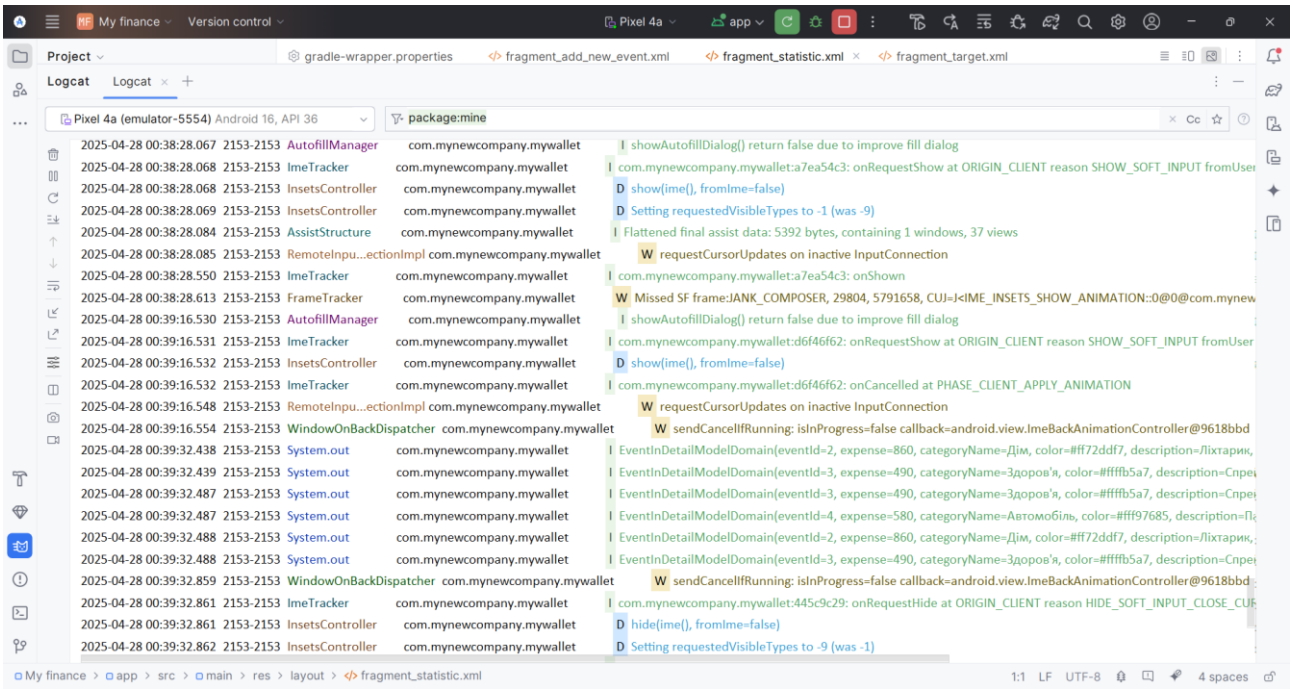


*Рисунок 3.3.2 - Лог роботи додатку під час тестового запуску на реальному пристрої*

У межах порівняльного аналізу мобільного додатку «**Мої фінанси**» було проведено детальне зіставлення з популярними на ринку рішеннями, такими як **YNAB**, **Mint** та **PocketGuard**. Незважаючи на те, що ці продукти мають широку аудиторію та багатий функціонал, розроблений додаток вирізняється кількома ключовими перевагами, які роблять його унікальним і не мають прямих аналогів серед існуючих сервісів.

Перш за все він має орієнтований на максимальну простоту використання, що є важливим чинником для широкого кола користувачів, особливо тих, хто не хоче витрачати багато часу на навчання або розбиратися з надмірно складними функціями. Водночас, завдяки оптимізованій внутрішній логіці та ефективній роботі з базою даних, усі основні операції - від додавання нових записів до побудови графіків витрат - виконуються миттєво, навіть на пристроях із середнім технічним рівнем. Це забезпечує відчутно кращу швидкодію порівняно з більшістю конкурентів, де затримки та "важкість" додатку часто викликають незручності у користувачів.

Другий важливий аспект — це зручність та інтуїтивність інтерфейсу. Під час тестування користувачі відзначали високий рівень комфорту при роботі з додатком, що значно покращує загальний користувацький досвід. У той час як інші програми часто надають величезний спектр функцій, які не завжди є потрібними кожному користувачу, додаток зосереджується саме на базових, але найнеобхідніших інструментах. Це дозволяє уникнути перевантаження та спрощує навігацію.

Щодо надійності, результати тестування підтвердили стабільність роботи додатку навіть при тривалому використанні і великій кількості транзакцій. Відсутність критичних збоїв, втрати даних чи помилок у функціоналі свідчить про високий рівень розробки архітектури та обґрунтованість вибору технологічних рішень. Надійність є одним із ключових факторів для користувачів, адже фінансові дані мають бути завжди під захистом, а додаток - працювати безперебійно.

Важливо також відзначити, що додаток «**Мої фінанси**» споживає мінімальні системні ресурси, що робить його доступним навіть на бюджетних смартфонах - цей аспект суттєво відрізняє його від багатьох сучасних рішень, які часто вимагають потужнішого «заліза». Таким чином, додаток розширює аудиторію користувачів і задовольняє потреби тих, хто не готовий або не може інвестувати у дорогі пристрої.

Аналіз результатів тестування і оцінка ефективності роботи додатку дозволили визначити перспективні напрямки для подальшого розвитку. Зокрема, у планах передбачено впровадження функції синхронізації даних із хмарними сервісами, що значно підвищить безпеку збереження інформації та дозволить користувачам легко переносити дані між пристроями без ризику їх втрати. Також планується розробка розширеного аналітичного модуля, який дасть змогу не лише відслідковувати поточні доходи та витрати, а й будувати прогнози на основі історичних даних, аналізувати фінансові тенденції, що є важливою перевагою для користувачів, які прагнуть глибшого контролю над своїм бюджетом.

Крім того, розробка підтримки мультивалютності дозволить ефективно працювати з кількома валютами одночасно, що актуально для користувачів, які ведуть міжнародні фінансові операції або часто подорожують. Запровадження системи нагадувань про регулярні платежі та заплановані витрати підвищить рівень персоналізації та допоможе уникнути пропущених платежів і непередбачених фінансових втрат.

Отже, проведене тестування, аналіз ефективності і сформульовані напрями розвитку свідчать про високий потенціал мобільного додатку «**Мої фінанси**». Його унікальне поєднання простоти, швидкості роботи та надійності забезпечує значні переваги порівняно з існуючими аналогами. Це підтверджує доцільність подальшого вдосконалення і масштабування додатку, що дозволить задовольнити зростаючі потреби користувачів та підвищити рівень їхнього фінансового комфорту.

## ВИСНОВКИ

У межах даної дипломної роботи було реалізовано повноцінний проєкт створення мобільного застосунку для автоматизації планування особистого бюджету. Розробка такого програмного продукту є актуальною відповіддю на сучасні потреби суспільства, яке дедалі більше звертається до цифрових інструментів з метою ефективного керування фінансами. У процесі виконання роботи було досягнуто поставлену мету та вирішено основні завдання, що охоплювали як теоретичний, так і практичний аспекти.

На першому етапі дослідження було проаналізовано сутність процесу планування особистого бюджету, вивчено роль фінансової грамотності у повсякденному житті людини. Особистий бюджет — це інструмент, що дозволяє людині не лише контролювати свої доходи та витрати, а й планувати майбутні фінансові дії, накопичення, інвестиції. У цьому контексті було розглянуто функціональність і недоліки існуючих мобільних застосунків для управління фінансами. Проведений аналіз дав змогу виявити ряд спільних проблем таких

програм, серед яких складний інтерфейс, обмежена функціональність, відсутність локалізації українською мовою, а також недостатня персоналізація під потреби користувача.

На основі отриманих результатів було сформовано вимоги до власного програмного продукту. Основними принципами, які лягли в основу створення застосунку, стали простота, інтуїтивність, адаптивність і функціональність. Програму реалізовано на платформі Android з використанням Android Studio - одного з найпопулярніших середовищ для створення мобільних застосунків. У ході реалізації було застосовано сучасні підходи до архітектури програмного забезпечення, включаючи розділення на шари логіки, використання бази даних Room, шаблони MVVM, а також ефективну взаємодію між інтерфейсом та бізнес-логікою.

У процесі реалізації було впроваджено такі можливості, як ведення доходів і витрат із категоріями, перегляд статистики у вигляді графіків, редагування та видалення записів, локалізація українською мовою, а також збереження даних у локальній базі. Було створено зручний і сучасний інтерфейс, що відповідає вимогам користувача щодо візуального сприйняття, використано гармонійні кольори та структурований підхід до подачі інформації. Застосунок протестовано на реальних пристроях, перевірено на відповідність заявленим вимогам і відсутність критичних помилок.

У процесі тестування було застосовано основні види тестування, зокрема функціональне, модульне, інтеграційне та ручне. Це дозволило виявити незначні недоліки на ранніх етапах розробки, що в результаті забезпечило стабільну роботу застосунку. Також було сформовано баг-репорти з прикладами опису типових помилок відповідно до принципів оформлення в теорії тестування.

Результати дипломної роботи засвідчили ефективність обраних методів розробки. Особливу увагу було приділено відповідності програми потребам цільової аудиторії, що досягається за рахунок простоти використання, зручного

введення даних та зрозумілої статистики. Крім того, продукт може бути основою для подальшого розвитку. Зокрема, можлива реалізація хмарного збереження, синхронізації між пристроями, додавання нагадувань про платежі.

Також важливо зазначити, що створений застосунок відповідає сучасним вимогам безпеки та конфіденційності, що особливо актуально у сфері фінансів. Використання локального шифрування та можливість додаткової автентифікації гарантують захист особистих даних користувачів. Цей аспект має велике значення для формування довіри та стимулювання активного використання мобільного рішення.

Перспективи подальшого розвитку проєкту пов'язані не лише з технічним удосконаленням, але й із розширенням функціоналу. Планується впровадження аналітичних інструментів, що дозволять прогнозувати фінансові тенденції, а також інструментів для автоматичного розпізнавання транзакцій із банківських виписок. Це значно підвищить зручність і точність обліку. Крім того, розробка мультивалютної підтримки та адаптація під різні регіональні стандарти розширить потенційну аудиторію застосунку.

Таким чином, дипломна робота підтвердила, що створення мобільного застосунку для планування особистого бюджету є не лише технічно можливим, а й соціально значущим завданням. Реалізований продукт є прикладом того, як інформаційні технології можуть сприяти підвищенню фінансової грамотності, організованості та відповідальності користувачів. Робота має практичну цінність, може бути використана в реальному житті та слугувати базою для подальших досліджень у сфері мобільної розробки й фінансових сервісів.

## СПИСОК ЛІТЕРАТУРИ


1. Алессандрія С. *Flutter Cookbook*. Birmingham: Packt Publishing, 2020. 470 с.
2. Білецький В. О. *Фінансове планування: теорія та практика*. Київ: КНЕУ, 2018. 276 с.
3. Блох Дж., Мерфі А. *Kotlin в дії*. 2-ге вид. К.: Манн, Іванов і Фербер, 2021. 392 с.
4. Богл Дж. К. *Bogle on Mutual Funds: New Perspectives for the Intelligent Investor*. New York: McGraw-Hill, 2010. 320 p.
5. Гауст Л. *Google Flutter Mobile Development Quick Start Guide*. Addison-Wesley, 2019. 210 p.
6. Гордон П. *Beginning App Development with Flutter*. Pearson, 2019. 250 p.
7. Гуревич І. М. *Особисті фінанси: управління та інвестиції*. Харків: Фінанси і статистика, 2020. 312 с.
8. Дарн Дж. *Personal Financial Management*. New York: Springer, 2017. 280 p.
9. Еванс Ч. *Android Programming: The Big Nerd Ranch Guide*. 4-е вид. Berkeley: Big Nerd Ranch, 2019. 624 p.
10. З'єра К., Бейтс Б. *Head First Kotlin: A Brain-Friendly Guide*. O'Reilly Media, 2019. 360 p.
11. Кійосакі Р. *Багатий тато, бідний тато*. Київ: Наш Формат, 2017. 352 с.
12. Колінз Дж. Л. *The Simple Path to Wealth: Your Road Map to Financial Independence and a Rich, Free Life*. JL Collins NH, 2016. 286 p.
13. Корнійчук О. П. *Мобільні технології в управлінні фінансами*. Львів: Видавництво ЛНУ, 2019. 224 с.
14. Кокоріна Н. О. *UI/UX Design Fundamentals*. Київ: Loop, 2021. 196 с.
15. Класон Д. *Найбагатший чоловік у Вавилоні*. Харків: Фоліо, 2020. 144 с.
16. Мороні Л. *Flutter in Action*. Manning Publications, 2021. 440 p.
17. Наполі М. Л. *Beginning Flutter: A Hands On Guide to App Development*. Packt Publishing, 2019. 420 p.
18. Нільсен Я. *Usability Engineering*. San Francisco: Morgan Kaufmann, 1994. 362 p.

19.Орман С. *Women & Money: Owning the Power to Control Your Destiny*. Free Press, 2007. 304 p.
20.Пархоменко О. Ю. *Методика тестування програмного забезпечення*. Миколаїв: МНАУ, 2022. 160 с.
21.Пейн Р. *Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2*. Birmingham: Packt, 2020. 394 p.
22.Прессман Р. С. *Інженерія програмного забезпечення: практичний підхід*. 9-е вид. К.: Видавництво «Вільямс», 2020. 960 с.
23.Рамсі Д. *Фінансова свобода: Повне перетворення вашого бюджету*. Київ: Ніка-Центр, 2020. 328 с.
24.Рамсі Д. *The Total Money Makeover Workbook*. Ramsey Press, 2013. 256 p.
25.Рубін Д., Чіснелл Д. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. 2-е вид. Wiley, 2008. 384 p.
26.Сієрра К., Бейтс Б. *Head First Kotlin*. O'Reilly Media, 2019. 360 p.
27.Стенлі Т., Данко В. Д. *The Millionaire Next Door: The Surprising Secrets of America's Wealthy*. Continuum, 1996. 272 p.
28.Талер Р. Х., Сансейн К. Р. *Nudge: Improving Decisions About Health, Wealth, and Happiness*. Yale University Press, 2008. 312 p.
29.Хаусел М. *The Psychology of Money: Timeless Lessons on Wealth, Greed, and Happiness*. Harriman House, 2020. 252 p.
30.Хінден С. *How to Retire Happy, Wild, and Free: Retirement Wisdom That You Won't Get from Your Financial Advisor*. Wiley, 2008. 240 p.

# ДОДАТКИ

## Код розробленого додатку

## ДОДАТОК А

*Activitymain.xml:*

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/activity_main"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".presentation.MainActivity">

    <fragment
        android:id="@+id/fragmentContainerView"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:defaultNavHost="true"
        app:navGraph="@navigation/nav_graph"
        app:layout_constraintBottom_toTopOf="@id/mainBottomNavigationView"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        tools:ignore="FragmentTagUsage" />

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/mainBottomNavigationView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="?android:attr/windowBackground"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:menu="@menu/menu_bottom_main"
        tools:ignore="VisualLintBottomNav" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

# ДОДАТОК Б

Navgraph.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/nav_graph"
    app:startDestination="@id/mainFragment">

    <fragment
        android:id="@+id/mainFragment"
        android:name="com.mynewcompany.mywallet.presentation.main.MainFragment"
        android:label="@string/home"
        tools:layout="@layout/fragment_main">
        <action
            android:id="@+id/action_mainFragment_to_addNewEventFragment"
            app:destination="@id/addNewEventFragment"
            app:enterAnim="@anim/slide_in_left"
            app:exitAnim="@anim/slide_out_right"
            app:popEnterAnim="@anim/slide_in_right"
            app:popExitAnim="@anim/slide_out_left" />
        <action
            android:id="@+id/action_mainFragment_to_detailsFragment"
            app:destination="@id/detailsFragment"
            app:enterAnim="@anim/slide_in_left"
            app:exitAnim="@anim/slide_out_right"
            app:popEnterAnim="@anim/slide_in_right"
            app:popExitAnim="@anim/slide_out_left" />
        <action
            android:id="@+id/action_mainFragment_to_targetFragment"
            app:destination="@id/targetFragment"
            app:enterAnim="@anim/slide_in_left"
            app:exitAnim="@anim/slide_out_right"
            app:popEnterAnim="@anim/slide_in_right"
            app:popExitAnim="@anim/slide_out_left"/>
    </fragment>
    <fragment
        android:id="@+id/plansFragment"
        android:name="com.mynewcompany.mywallet.presentation.statistics.StatisticFragment"
        android:label="@string/statistic"
        tools:layout="@layout/fragment_statistic" />
    <fragment
        android:id="@+id/categoriesFragment"

android:name="com.mynewcompany.mywallet.presentation.categories.CategoriesFragment"
        android:label="My categories"
        tools:layout="@layout/fragment_categories">
        <action
            android:id="@+id/action_categoriesFragment_to_addNewCategoryFragment"
            app:destination="@id/addNewCategoryFragment"
            app:enterAnim="@anim/slide_in_left"
            app:exitAnim="@anim/slide_out_right"
            app:popEnterAnim="@anim/slide_in_right"
            app:popExitAnim="@anim/slide_out_left" />
        <action
            android:id="@+id/action_categoriesFragment_to_changeCategoryFragment"
            app:destination="@id/changeCategoryFragment"
            app:enterAnim="@anim/slide_in_left"
            app:exitAnim="@anim/slide_out_right"
            app:popEnterAnim="@anim/slide_in_right"
```

```
            app:popExitAnim="@anim/slide_out_left" />
        <argument
            android:name="categoryModel"
            app:argType="com.mynewcompany.mywallet.domain.model.CategoryModelDomain" />
    </fragment>
    <fragment
        android:id="@+id/settingsFragment"
        android:name="com.mynewcompany.mywallet.presentation.settings.SettingsFragment"
        android:label="SettingsFragment"
        tools:layout="@xml/root_preferences" />
    <fragment
        android:id="@+id/addNewEventFragment"
        android:name="com.mynewcompany.mywallet.presentation.addevent.AddNewEventFragment"
        android:label="fragment_add_new_event"
        tools:layout="@layout/fragment_add_new_event" />
    <fragment
        android:id="@+id/detailsFragment"
        android:name="com.mynewcompany.mywallet.presentation.details.DetailsFragment"
        android:label="fragment_details"
        tools:layout="@layout/fragment_add_new_event">
        <argument
            android:name="eventToChange"
            app:argType="com.mynewcompany.mywallet.domain.model.EventInDetailModelDomain"
/>
    </fragment>
    <fragment
        android:id="@+id/iconsFragment"

android:name="com.mynewcompany.mywallet.presentation.categories.chooseicons.IconsFragment"
        android:label="fragment_icons"
        tools:layout="@layout/fragment_icons">
        <action
            android:id="@+id/action_iconsFragment_to_addNewCategoryFragment"
            app:destination="@id/addNewCategoryFragment" />
        <argument
            android:name="icon"
            android:defaultValue="2131230857"
            app:argType="integer" />
        <action
            android:id="@+id/action_iconsFragment_to_changeCategoryFragment"
            app:destination="@id/changeCategoryFragment" />
        <argument
            android:name="icon"
            android:defaultValue="2131230857"
            app:argType="integer" />
    </fragment>
    <fragment
        android:id="@+id/addNewCategoryFragment"

android:name="com.mynewcompany.mywallet.presentation.categories.addnewcategory.AddNewCateg
oryFragment"
        android:label="fragment_add_new_category"
        tools:layout="@layout/fragment_add_new_category">
        <action
            android:id="@+id/action_addNewCategoryFragment_to_iconsFragment"
            app:destination="@id/iconsFragment"
            app:enterAnim="@anim/slide_in_left"
            app:exitAnim="@anim/slide_out_right"
            app:popEnterAnim="@anim/slide_in_right"
            app:popExitAnim="@anim/slide_out_left" />
        <argument
            android:name="colorPosition"
            android:defaultValue="1"
```

```
            app:argType="integer" />
        <argument
            android:name="color"
            app:argType="string" />
    </fragment>
    <fragment
        android:id="@+id/changeCategoryFragment"

android:name="com.mynewcompany.mywallet.presentation.categories.changecategory.ChangeCateg
oryFragment"
        android:label="fragment_change_category"
        tools:layout="@layout/fragment_add_new_category">
        <action
            android:id="@+id/action_changeCategoryFragment_to_iconsFragment"
            app:destination="@id/iconsFragment"
            app:enterAnim="@anim/slide_in_left"
            app:exitAnim="@anim/slide_out_right"
            app:popEnterAnim="@anim/slide_in_right"
            app:popExitAnim="@anim/slide_out_left" />
        <argument
            android:name="colorPosition"
            android:defaultValue="1"
            app:argType="integer" />
        <argument
            android:name="color"
            app:argType="string" />
    </fragment>
    <fragment
        android:id="@+id/targetFragment"
        android:name="com.mynewcompany.mywallet.presentation.target.TargetFragment"
        android:label="fragment_target"
        tools:layout="@layout/fragment_target" />
</navigation>
```

# ДОДАТОК В

Fragment_target.xml:

```
 <?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

    tools:context=".presentation.target.TargetFragment">

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/btnEscape"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="-30dp"
        android:backgroundTint="@color/lightgray"
        android:src="@drawable/baseline_clear_24"
        app:backgroundTint="@color/gray"
        app:layout_constraintBottom_toTopOf="@id/internalConstrainLayout_target"
        app:layout_constraintStart_toStartOf="@id/internalConstrainLayout_target"
        tools:ignore="ContentDescription" />


    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/internalConstrainLayout_target"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="15dp"
        android:layout_marginTop="35dp"
        android:layout_marginEnd="15dp"
        android:background="@drawable/add_my_custom_shape_soft_recycler_light_green"
        android:paddingTop="20dp"
        android:paddingBottom="20dp"
        app:circularflow_angles="10dp"
        app:circularflow_radiusInDP="12dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <TextView
            android:id="@+id/helperText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="20dp"
            android:text="@string/helper"
            android:textSize="20sp"
            android:textColor="@color/white"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <EditText
            android:id="@+id/editTarget"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="30dp"
            android:layout_marginEnd="30dp"
            android:textColor="@color/white"
            android:textSize="34sp"
            android:autofillHints="target"
```

```
            android:inputType="number"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@id/helperText"
            tools:ignore="LabelFor" />

        <TextView
            android:id="@+id/helperText1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="20dp"
            android:textColor="@color/white"
            android:text="@string/helper1"
            android:textSize="20sp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@id/editTarget" />

    </androidx.constraintlayout.widget.ConstraintLayout>
    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/btnSave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="-20dp"
        app:layout_constraintBottom_toBottomOf="@id/internalConstrainLayout_target"
        app:layout_constraintEnd_toEndOf="@id/internalConstrainLayout_target"
        android:backgroundTint="@color/pastel_yellow"
        app:backgroundTint="@color/pastel_light_yellow"
        android:src="@drawable/baseline_check_24"
        tools:ignore="ContentDescription" />
</androidx.constraintlayout.widget.ConstraintLayout>
```