

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СІРЕНКО

(підпис)

“ _ ” _____ 2022 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою “Комп’ютерні системи та мережі”

спеціальності 123 “Комп’ютерна інженерія”

на тему: Мобільний додаток для обліку особистих витрат

Виконав студент 4 курсу, групи ІВ-82

Сірокомський Микола Сергійович

(прізвище, ім’я, по батькові)

(підпис)

Керівник доц., к.т.н.Павлов В.Г.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль) Сімоненко А.В.

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент д.т.н., проф. Корнієнко Б.Я.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальність 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

Сергій СІРЕНКО

(підпис)

“ ___ ” _____ 2022 р.

ЗАВДАННЯ

на бакалаврський дипломний проєкт студента

Сірокомського Миколи Сергійовича

1. Тема проєкту Мобільний додаток для обліку особистих витрат
2. Керівник проєкту доц., к.т.н.Павлов В.Г.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від _____ 2022 року № _____
3. Термін здачі студентом закінченого проєкту 16 червня 2022 р.
4. Вихідні дані до проєкту
5. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)
Аналіз предметної області, Аналіз засобів реалізації, Деталі реалізації.

6. Консультанта проекту

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Сімоненко А.В.		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів дипломного проекту (роботи)	Строк Виконання етапів проекту	Примітки
1.	<i>Затвердження теми роботи</i>	<i>10.12.2021-15.12.2021</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>15.12.2021-15.03.2022</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>15.03.2022-25.03.2022</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>25.03.2022-5.04.2022</i>	
5.	<i>Програмна реалізація системи</i>	<i>5.04.2022-15.04.2022</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>15.04.2022-20.05.2022</i>	
7.	<i>Захист програмного продукту</i>	<i>30.04.2022</i>	
8.	<i>Передзахист</i>	<i>30.05.2022</i>	
9.	<i>Захист</i>	<i>23.06.2022</i>	

Студент-дипломник _____

(підпис)

Керівник проекту _____

(підпис)

Анотація

У цій роботі розглядаються різні мобільні додатки для персонального фінансового обліку для Android. Зроблено порівняльний опис і обрано функції, які найкраще підходять для зменшення особистих витрат, планування бюджету та розподілення коштів для особистих потреб для найбільш ефективного використання. Інструменти архітектури, платформи були обрані для оптимальної продуктивності та простого інтерфейсу. На цій основі програмна реалізація програми виконується самостійно.

Annotation

This work discusses various mobile applications for personal financial accounting for Android. A comparative description is made and the functions that are best suited for reducing personal costs, budget planning and allocating funds for personal needs for the most efficient use are selected. Architecture tools, platforms have been chosen for optimal performance and a simple interface. On this basis, the software implementation of the program is performed independently.

Опис альбому

До дипломного проекту

На тему: Мобільний додаток для обліку особистих витрат

справки	Формат	Значення	Найменування	Кіл. листів	№ екземпляр	Додаток
			Документація загальна			
			Знову розроблена			
	<i>A4</i>	<i>ІАЛЦ.466538.002 ТЗ</i>	Мобільний додаток для обліку особистих витрат	1		
			Технічне завдання			
	<i>A4</i>	<i>ІАЛЦ.466538.003 ТП</i>	Мобільний додаток для обліку особистих витрат	2		
			Відомість технічного проекту			
	<i>A4</i>	<i>ІАЛЦ.466538.004 ПЗ</i>	Мобільний додаток для обліку особистих витрат	64		
			Пояснювальна записка			
	<i>A4</i>	<i>ІАЛЦ.466538.005 Д1</i>	Мобільний додаток для обліку особистих витрат	1		
			Принципова схема			
	<i>A4</i>	<i>ІАЛЦ.466538.006 Д2</i>	Мобільний додаток для обліку особистих витрат	1		
			Функціональна схема			
	<i>A4</i>	<i>ІАЛЦ.466538.007 Д3</i>	Мобільний додаток для обліку особистих витрат	1		
			Структурна схема			

					ІАЛЦ.466538.001 АО			
<i>Зм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>				
<i>Розробив</i>		Сірокомський М.			Мобільний додаток для обліку особистих витрат	Літ.	Аркуш	Аркушів
<i>Перевірів</i>		Павлов В.Г.					1	1
<i>Консультант з нормоконтролю.</i>		Сімоненко А.В.			НТУУ "КПІ" ФІОТ ІВ-82			

ТЕХНІЧНЕ ЗАВДАННЯ

**до дипломного проєкту
освітньо-кваліфікаційного рівня бакалавр**

на тему: «Мобільний додаток для обліку особистих витрат»

Технічне завдання до дипломного проекту

Зміст

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	2
2. ПРИЧИНИ ДЛЯ РОЗРОБКИ.....	2
3. ЦІЛЬ ТА ПРИЗНАЧЕННЯ РОЗРОБКИ	2
4. ДЖЕРЕЛА РОЗРОБКИ	2
5. ТЕХНІЧНІ ВИМОГИ	3
5.1 ВИМОГИ ДО ПРОДУКТУ, ЩО РОЗРОБЛЯЄТЬСЯ.....	3
5.2 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	3
5.3 ВИМОГИ ДО АПАРАТНОЇ ЧАСТИНИ.....	3
6. ЕТАПИ РОЗРОБКИ.....	4

					ІАЛЦ.466538.002 ТЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Сірокомський М.С.			Мобільний додаток для обліку особистих витрат	Лит.	Арк	Аркушів
Перев.		Павлов В.Г.					1	2
Реценз.						НТУУ «КПІ»		
Н. Контр.		Сімоненко А. В.						
Затверд.								

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

Це технічне завдання передбачає розробку мобільного додатка для персонального фінансового обліку. Сфера дії всієї системи – це автоматизація процесу бюджетування та сортування витрат по категоріям.

2. ПРИЧИНИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання бакалаврського проекту по освітньо-професійної програми — спеціальності 123 — «Комп'ютерна інженерія», затверджене кафедрою Обчислювальної техніки Національного технічного Університету України — «Київський Політехнічний інститут імені Ігоря Сікорського».

3. ЦІЛЬ ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою розробки є програмна реалізація мобільного додатка персонального фінансового обліку.

4. ДЖЕРЕЛА РОЗРОБКИ

У сучасному світі існує багато аналогів цього додатка, зокрема Spendeo, Monefy тощо, які стали джерелом натхнення та дали нові ідеї для створення додатків. Іншими джерелами розвитку є науково-технічна література з теорії та практики програмування, алгоритми, шаблони, публікації в Інтернеті з цих питань.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленого продукту

- Надає можливість перегляду всіх витрат за місяць по конкретним категоріям;
- Надати можливість перегляду звіту про бюджет за вибраний час;
- Забезпечити вибору категорій витрат та встановлення ліміту на них
- Можливість ставити бюджетні цілі та досягати їх;

5.2. Вимоги до програмного забезпечення

- Операційна система MS Windows 7 SP1 або вище
- Android Studio
- Android SDK
- Windows Powershell 5.0 або вище

5.3. Вимоги до апаратної частини

- Операційна система Android 4.1 або вище
- Оперативної пам'яті не менше 1 Гбайт
- Вільний простір жорсткого диска не менше 200 Мбайт
- Підключення до Інтернету

6. ЕТАПИ РОЗРОБКИ

	Дата
6.1 Вивчення літератури	20.12.2021
6.2 Складання і узгодження технічного завдання	15.01.2022
6.3 Аналіз структури розробленого курсу навчання	27.01.2022
6.4 Створення модулів розробленого курсу	14.02.2022
6.5 Тестування дистанційного доступу до курсу	01.05.2022
6.6 Відлагодження і виправлення помилок	15.05.2022
6.7 Оформлення документації дипломного проєкту	03.06.2022

					<i>ІАЛЦ.466538.003 ТП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		2

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проєкту

на тему: «Мобільний додаток для обліку особистих витрат»

Київ 2022-року

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1. Особистий фінансовий облік	6
1.2. Основні принципи при розробці мобільних додатків	9
1.3. Аналіз та порівняння аналогічних додатків	14
1.3.1 Expense Manager	15
1.3.2. Додаток Andro Money	16
1.3.3. Додаток Money Pro	18
1.3.4. Monify	20
Висновки до розділу 1	22
РОЗДІЛ 2. АНАЛІЗ ЗАСОБІВ РЕАЛІЗАЦІЇ.....	23
2.1 Про Android	23
2.2 Архітектура Android.....	25
2.3 Розробка Android-застосунків	28
2.4 Android SDK.....	29
2.5 Фреймворки Розробки	30
2.5.1 Flutter	30

					ІАЛЦ.467200.003 ПЗ			
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Сірокомський М.			Мобільний додаток для обліку особистих витрат	Літ.	Аркуш	Аркушів
Перевірив		Павлов В.Г.					1	64
						НТУУ «КПІ»		
Н. контр.		Сімоненко А. В.						
Затверд.								

2.5.2 React Native.....	33
2.5.3 Xamarin	34
2.5.4 JavaScript.....	35
2.5.5 Dart.....	36
2.5.6 C#	37
2.5.7 Порівняння фреймворків для розробки	39
2.6 Бази даних	40
Висновки до розділу 2.....	43
РОЗДІЛ 3. ДЕТАЛІ РЕАЛІЗАЦІЇ ДОДАТКУ	44
3.1 Загальні відомості по розробці	44
3.2 Розробка дизайну	45
3.3 Розробка функціоналу	51
3.4 Робота з додатком	54
Висновки до розділу 3.....	61
ВИСНОВКИ	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	63

ПЕРЕЛІК СКОРОЧЕНЬ

SQL Structured query language (Мова Структурованих Запитів)

SDK Software development kit

(Комплект для розробки програмного забезпечення)

JSX JavaScript XML

JiT Just-in-time (Вчасно)

HTML Hypertext markup language (Мова гіпертекстової розмітки)

UX User experience (Досвід користувача)

АoT Ahead-of-time (Достроково)

API Application programming interface

(Інтерфейс прикладного програмування)

DX DirectX

APK Android Package (Пакет Android)

UI User interface (Інтерфейс користувача)

XML Extensible Markup Language

(Розширювана мова розмітки)

OS Operation System (Операційна система)

I/O Input/Output (Ввід/вивід)

GPS Global Positioning System (Глобальна система позиціонування)

RAM Random Access Memory (Оперативна пам'ять)

SMS Short Message Service (Служба коротких повідомлень)

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		3

ВСТУП

Сьогодні дуже актуальним є питання особистого бюджетного контролю. Як розрізняють успішну людину серед інших? Вона завжди є впевненою в собі і своєму накопиченому капіталі, тому що такі люди знають як контролювати свої кошти. Вони дуже уважно стежать за рухом грошей: доходів і витрат. Крім того, контролюють сфери, де гроші використовуються.

Процес відстеження бюджету може бути складним і тривалим. Але зараз проблем бути не повинно. Розроблений додаток у цьому дипломному проєкті допоможе користувачам заощадити гроші та час. Користувачам потрібно лише слідкувати за своїми витратами та аналізувати їх.

У сучасному світі розроблено велику кількість систем персонального обліку. Розробники програмного забезпечення мають завдання впровадити прості у використанні, але ефективні програми, які дозволяють їм легко відстежувати бюджети.

Метою даного проєкту є розробка програми для Android, ціллю якої є полегшення персонального фінансового обліку.

Щоб цього досягти, необхідно вирішити наступні завдання:

- ❖ Оглянути літературу на вибрані теми;
- ❖ Зрозуміти архітектуру, інструменти та можливості розробки додатків
- ❖ Розробляти програму відповідно до вимог;
- ❖ Розробити інтерфейс користувача;
- ❖ Протестувати розроблений додаток.

Під час виконання завдання будуть проаналізовані існуючі рішення та вибрана відповідна практична основа, на якій можна буде перейти до впровадження програмного забезпечення та тестування мобільного додатка персонального фінансового обліку.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		4

Перший розділ складається із огляду існуючих рішень на ринку, що передбачають реалізацію мобільного додатку. В цей розділ буде входити опис розроблених систем персонального обліку.

Крім того, він буде включати аналіз ефективності відомих додатків, економічна оцінка застосування того, чи іншого забезпечення, а також виконає завдання та мету розробки власного додатка.

У другій частині представлений огляд методів, які використовуються для створення мобільних додатків. Він також забезпечує порівняння різних технологій, аналіз ефективності тих вибраних продуктів, які будуть ключовими для готовності програмного забезпечення.

Третя частина описує розробку та впровадження програми та інтерфейсу користувача вибраного макета, інструкцію для користувача.

Наприкінці роботи всі розділи дають загальні висновки щодо того, чому в кінцевому цей продукт дійсно відповідає заявленим вимогам та має користь для пересічного користувача.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		5

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Особистий фінансовий облік

Переважно усі компанії ведуть особистий облік фінансів. Компанії потребують цього, бо створюються для розвитку та розширення у подальшому, а для цього потребують прибутку. Сім'ї також можуть вести власні фінансові записи, щоб контролювати та створювати певні цілі. Вести записи не складно. Як і з будь-якою звичкою, особистими фінансами слід займатися відповідно до власної волі та власної внутрішньої мотивації. Спочатку справа виглядає важкою і не покращується, але насправді на день витрачається 5-7 хвилин, лише півтори години на місяць. Щодня приділяйте час на перевірку своїх фінансів, а потім на основі зібраної статистики можна коригувати марні або надто дорогі звички чи роботу, щоб збільшити потік доходу.

Особисті фінанси не можуть відповісти на питання «звідки берете гроші?», проте здатні допомогти та краще зрозуміти, звідки надходять гроші, куди вони витрачаються, і на основі цієї інформації кожен може зробити власні висновки. Корисність фінансового менеджменту, перш за все, полягає в тому, щоб зрозуміти, як і для чого ми живемо, зробити певні висновки з фінансового менеджменту – жити щодо своїх можливостей. Якщо людина хоче жити краще більше витратити чи більше заощадити – їй доводиться більше заробляти або накопичувати на майбутнє, контролюючи «марні» витрати.

Основні кроки до управління власним бюджетом.[6]

Крок 1. Найпростіший спосіб розпочати роботу – це запам'ятати, де зберігаються гроші. Практично у кожного є гаманець, яким користуються часто або хоча б зрідка. Тому першим записом у списку місць для заощадження буде «Готівка».

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		6

Хтось завершує, хтось продовжує, записуючи, наприклад, «банківський депозит», платіжну картку тощо. У програмі обліку особистих фінансів цей крок буде називатися «your wallet», тобто твій гаманець. Насправді, у сучасних людей все менше готівки і все більше грошей на банківських рахунках. Очевидно, що кредитні картки є основою бюджету у більшості, з тією лише різницею, що вони будуть мати від'ємний баланс. Ще одним видом рахунків можна вважати борг.

Крок 2. Створити власний список категорій витрат. Якщо рахунок дозволяє просто побачити, де знаходиться та чи інша сума, то категорії дають уявлення про те, куди витрачаються гроші. Багато планів пропонують готові списки категорій витрат і доходів. Але найкраще зробити це самостійно, адже у кожного своя ситуація і не існує єдиного списку для всіх. Можна створювати власні категорії, додаючи, видаляючи або змінюючи категорії, використовуючи списки категорій інших людей як підказки. Діаграму категорій таких витрат та доходів додано у кінці проекту.

Усі види витрат важко передбачити та відсортувати в категорії, для деяких видів витрат просто недоцільно створювати категорію, якщо такі витрати зустрічається нечасто і мають невеликі розміри. У цьому випадку вартість входить до категорії «непередбачені». Усі помилки в обліку, якщо такі є, також підпадають під цю категорію.

Крок 3. Далі потрібно почати відстежувати витрати і доходи. Акт витрачання, отримання або переказ коштів між рахунками також відомий як транзакція. Кожна операція повинна відповідати певному категорії та даті.

Підхід різний, і всі виграють від особистих фінансів. Деякі переваги реєстрації власних доходів і витрат можуть мотивувати сім'ї: коли людина починає вести фінансовий облік, її фінансове становище може стабілізуватися або покращитися. Людина багато знає про потік власних грошей. Вона знає, скільки витрачає на місяць. Вона також дізналася, що насправді витрачає не так багато грошей, як думала

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		7

Проте маєте чітке уявлення про те, куди йдуть гроші. Вона знає, які предмети споживання споживають найбільше грошей, а які – найменше. Вона почне контролювати власні фінанси. Багато людей цього не знають і тому не знають, куди йдуть їхні гроші. Дивлячись на свій фінансовий звіт, починаєш замислюватися про цінність грошей: чи витратити їх на чергову непотрібну імпульсивну покупку. Вона зрозуміє своє фінансове становище і знатиме, як воно зміниться з часом.

Основні характеристики управління домашнім бюджетом:

1) Створення звіту. Це дуже важлива частина ведення сімейного бюджету. Комп'ютер дозволяє швидко сформувати необхідні звіти по господарству подальшого аналізу. Аналізуючи свій бюджет, можливо розробити стратегію та плани дій або внести корективи в існуючі плани.

2) Ведення кількох типів рахунків. Людина може мати кілька облікових записів. Такі як бюджет окремої сім'ї, банківські депозити тощо. Вся інформація відображається на екрані гаджету, що дуже зручно.

3) Особисте фінансове планування. Чудова можливість здійснити майбутні платежі та оцінити витрати. Планування дозволить людині прорахувати власні дії на кілька кроків вперед.

4) Розрахунок кредитів і депозитів. Багато програм мають вбудовані калькулятори для розрахунку кредиту/депозитів. Особливо зручними є ті сервіси, які дозволяють внести депозит. Дуже зручна і мотивуюча річ.

5) Контроль за боргом. Боржників можна додати, щоб точно відобразити особисте фінансове становище. Ця функція також допомагає не забути, кого і скільки позичала людина, адже іноді забуває, хто позичив гроші. Боржники, у свою чергу, забувають, у кого позичили.

6) Захист даних. Людина зможе за допомогою надійного пароля приховати всі свої дані від сторонніх очей.

7) Нагадування про необхідну оплату. Виплатити квартиру, погасити кредит, погасити борг, купити подарунок – щоб нічого не забути, про це нагадає План сімейного бюджету. Дивовижна особливість - процесори мають набагато кращу пам'ять, ніж люди.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		8

Як вести сімейний бюджет та його облік? Почнемо з того, що домашні бюджети потрібно контролювати якомога частіше. Бажано щодня. Це займає дуже мало часу. Все, що потрібно зробити на початку, це врахувати всі власні доходи і витрати, а потім занести їх у комірницю або спеціальну програму.

Контролювати бюджет треба напротязі довгого часу. Найкраще витратити не менше 2-3 місяців на збір даних для аналізу доходів і витрат. Ймовірно, що дані, зібрані цього місяця, об'єктивно не відобразатимуть домашні сімейні бюджети.

Отже, якщо не вести власний бюджет або робити це нерегулярно, результату не буде, краще змусити проводити цей експеримент протягом 2-3 місяців. Найважче - це витратити. Високу вартість легко запам'ятати. Але саме невеликі витрати до кінця місяця становитимуть 10-30%, що спричинить так звану «діру в кишені».

1.2 Основні принципи при розробці мобільних додатків

У цьому розділі порівняємо мобільні пристрої та їхні програми з їхніми настільними та серверними аналогами. Тож настав час перерахувати, що робить мобільний додаток дійсно якісним.

Важливою характеристикою мобільних додатків є час їх відкриття. Через високу частоту та коротку тривалість використання мобільних пристроїв можливість швидкого запуску мобільних додатків є обов'язковою умовою. Можливо, 6 секунд роздумів про те, щоб дочекатися заставки, енциклопедії чи середовища розробки, щоб з'явилося головне вікно програми, не дуже приносять задоволення, але, враховуючи загальну тривалість типового сеансу для такої програми, цей фактор можна вважати незначним. У випадку з мобільним додатком користувач витрачає 20 секунд, щоб використати або оновити невелику програму частку інформації, втрата 6 секунд - незаконна розкіш.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		9

Розумно дотримуватися мнемонічного правила, згідно з яким тривалість сесії користувача з програмою має бути значно довшою за тривалість запуску цієї програми. Для настільних додатків робочі сеанси тривають довше, тому користувачі готові миритися з довшим часом очікування.

У випадку з мобільним додатком тривалість робочого сеансу є короткою, тому тривалість циклу запуску програми повинна бути відповідно короткою. Тривалість циклу запуску програми також є важливим фактором для настільних комп'ютерів, а критична вона для мобільних пристроїв із нерегулярним і коротким робочим часом.

Ставлячись до мобільних пристроїв як до слухняних механічних іграшок, очікується, що вони поводитимуться відповідним чином. Коли користувач впливає на пристрій, натискає кнопку або маніпулює елементами керування девайса будь-яким іншим способом, він очікує, що його інтерфейс та система відреагує. Не отримавши негайної відповіді від пристрою, нетерплячий користувач починає нервувати і відразу ж повторює спробу виконати потрібну дію.

Це може спричинити проблеми, якщо маніпулювати елементами керування, клацати чи подібні дії будуть оброблені програмою помилково. Тому дуже важливо стежити за тим, щоб після виконання будь-яких дій з пристроєм користувач отримав від нього, в тій чи іншій формі, негайно підтвердження реакції на ці дії.

Ідеальним варіантом такого підтвердження є виконання запитаної дії; краще від цього не буде. Другим у цьому плані є підтвердження того, що запити надходять і обробляються у фоновому режимі, а програма зберігає можливість приймати інші запити. Третє місце належить відображенню курсора очікування, щоб інформувати користувача про те, що запит обробляється; програма не відповідає на жодні подальші запити, але користувач знає, що робота по обслуговуванню його запиту почалася і виконується.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		10

Найгірший випадок — коли користувач не бачить належної відповіді пристрою, він може лише здогадуватися, чи було прийнято його запит. Як і інші функції, які ми розглянемо, ці вимоги не є унікальними для мобільних пристроїв, але в даному випадку вони особливо важливі через особливості використання цих пристроїв і очікування, які люди мають від них.

Ще однією характеристикою мобільного додатка, що визначає його успіх, є ступінь його спеціалізації. План повинен мати чітко визначений набір завдань, які він повинен ефективно вирішувати. Дії, що виконуються в цьому випадку, повинні виконуватися швидко і вимагати від користувача мінімальної кількості натискань або інших дій.

З цієї причини пристрої часто мають спеціальні кнопки для конкретних завдань, які може виконувати пристрій (наприклад, кнопку для прямого переходу до бази даних, яка зберігає контактну інформацію, кнопку для перегляду календаря зустрічей або кнопку для зчитування штрих-коду та його передача для конкретної програми).

Потрібно докласти усіх зусиль, щоб якомога точніше охопити загальні завдання, які вимагають мінімальних дій користувача. Ця порада повинна керувати як високорівневими функціональними можливостями програми, так і завданнями низького рівня, які користувач зазвичай повинен вирішувати під час використання цієї програми. Наприклад, якщо мобільний додаток має забезпечити швидке введення даних, то потрібно звернути увагу на максимальне спрощення (і прискорення) цього процесу та вимірювати його ефективність. Якщо у мобільному додатку повинна бути можливість вказати конкретне місце за допомогою карти міських вулиць, щоб користувач міг вибрати маршрут до кінцевого пункту призначення, а потім переконатися, що це можна зробити якомога швидше. Важливо також встановити взаємодію із зовнішніми джерелами інформації.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		11

Треба розуміти, що створення чудової мобільної програми – це більше, ніж просто написання коду, який правильно працює на пристрої. Також не забувайте про зовнішнє програмне забезпечення, яке взаємодіє із мобільним додатком. Програми повинні враховувати особливості джерела інформації, що обслуговує мобільний пристрій, і перевіряти, що інформація надається у формі, яка відповідає цьому пристрою.

Прикладом є мобільний сервіс електронної пошти. Універсальна програма електронної пошти вимагає відповідного програмного забезпечення як на стороні сервера, так і на стороні клієнта. Клієнт звертається до сервера, щоб отримати інформацію про вхідні повідомлення та завантажити відповідний вміст на локальний пристрій. Оскільки мережеві з'єднання, які використовуються мобільними пристроями, мають нерегулярний доступ до них, знижену пропускну здатність і в багатьох випадках коштують дорожче, ніж мережеві з'єднання, які використовуються персональними комп'ютерами, поштова служба сервера повинна адаптуватися до цих вимог. Зазвичай це відбувається тому, що сервер надає інструменти для обмеження кількості одержуваного вмісту або фільтри, які передають лише ту інформацію, яка насправді може знадобитися користувачам мобільних пристроїв.

Можливо, знадобиться розширити функціональність серверних служб, спочатку розроблених для обслуговування персональних комп'ютерів, щоб підтримувати ефективні сценарії під час обслуговування мобільних пристроїв. Крім того, повинні існувати механізми для зміни параметрів конфігурації програми для роботи на серверах, настільних комп'ютерах і на самому мобільному пристрої, за допомогою яких користувачі можуть визначати свої запити на інформацію та налаштовувати інформаційні фільтри, щоб відповідати вимогам. Дизайн мобільних додатків далеко не обмежений фізичними межами самого мобільного пристрою.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		12

З огляду на те, що будь-який мобільний пристрій компактний і автономний, здається природним, що користувач захоче працювати з інтерфейсом, який не залежить від характеру завдання, яке вирішує пристрій. Кожен мобільний пристрій має свій колір. Типовий добре розроблений додаток розглядається не як набір окремих інструментів, а як гармонійне розширення самого пристрою.

Тому при розробці мобільних додатків дуже важливо дотримуватися єдиного стилю. Як запустити і зупинити програму, переміщатися в інтерфейсі та організувати розмову між користувачем і програмою, необхідно ретельно розглянути окремо для кожного типу пристрою. Користувачі мобільних пристроїв підсвідомо пристосовуються до метафор інтерфейсу користувача, і будь-яке відхилення від звичного шаблону може бути для них дуже незручним. Для настільних додатків створення звичного середовища робочого столу має великий сенс, але через різноманітність функцій, які надають такі програми, вони дозволяють досягти тієї ж мети кількома способами (наприклад, за допомогою комбінацій клавіш, натискань мишкою, меню та панелі інструментів). У випадку з мобільними пристроями, як правило, існує лише один спосіб вирішення цієї проблеми, і користувачі не можуть не звикнути до того чи іншого способу. Краще коли є чотири різні версії програми, кожна з яких відповідає певній метафорі інтерфейсу користувача.

Використовуючи аналогію зі сфери будівництва, настільні комп'ютери та ноутбуки можна порівняти з великими садибами в сільській місцевості. Тоді про мобільні пристрої можна сказати, що вони схожі на індивідуальні чи комунальні міські квартири. Кожен тип житла розроблений з урахуванням конкретних потреб і має специфічні вимоги для найбільш ефективного використання.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		13

Присадибні ділянки зазвичай великі за розміром і мають багато місць для зберігання всіляких речей. Деякі з цих місць знаходяться неподалік, а до інших важче дістатися. У присадибній ділянці речі, які рідко використовуються, зазвичай зберігаються на горищі або в підвалі, звідки їх можна періодично виносити за потреби.

З іншого боку, в умовах міської квартири придатних для зберігання речей місць набагато менше. У квартирі знаходяться речі, які повинні бути під рукою. Не зберігаються однакові речі, які нечасто використовуються в квартирі. Замість того, щоб купувати рідко використовувані речі, часто дешевше взяти їх напрокат.

Те ж саме стосується мобільних пристроїв. Як оперативна пам'ять, так і довготривала пам'ять зосереджені на зберіганні найбільш часто використовуваної інформації. Рідко використовувані дані слід завантажувати на сервер, щоб їх можна було запитувати в разі потреби.

У настільних комп'ютерах оперативна і довготривала пам'ять розділені. Між тим, пристрої RAM часто використовуються як робоча пам'ять для програм, так і як проміжне або довготривале сховище. Флеш-пам'ять все частіше використовується для тривалого зберігання даних, як правило, у вигляді карт пам'яті.

1.3 Аналіз та порівняння аналогічних додатків

Деякі користувачі почали використовувати програмне забезпечення для домашнього обліку ще довго після появи сучасних мобільних пристроїв. Тоді і зараз на ринку було багато різних програм, куди можна вносити свої доходи та витрати, а також сортувати всі свої транзакції за категоріями.

Але існують і великі фінансові «портфелі». Їх функціонал набагато ширший. За допомогою таких додатків користувачі зможуть відстежувати коливання ринкових цін на акції, відстежувати витрати на складні плани, контролювати свої інвестиції тощо. Кожен навіть зможе переглянути докладний звіт про витрати для певного аспекту, якщо захоче.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		14

Такий великий інструмент має багато недоліків:

- ❖ Немає безкоштовної версії
- ❖ Вони рідко націлені на звичайних людей
- ❖ Функція перенасичення
- ❖ Незручно використовувати на мобільних платформах

Та такі портфелі призначені більше для професіоналів та знавців фінансової справи, а для пересічного користувача вони не підходять, та занадто складні. Тож порівнювати будемо одні з найпоширеніших та найпопулярніших звичайних додатків для обліку фінансів.

1.3.1 Expense manager

Менеджер витрат (рисунок 1.1,1.2). Незважаючи на величезну кількість завантажень (близько мільйона), додаток є досить простим фінансовим планувальником. Розробник заздалегідь встановив деякі основні категорії витрат (можна змінити в налаштуваннях). Крім того, користувачі можуть встановлювати регулярні платежі або переглядати свої щотижневі, місячні та річні розклади. На жаль, повна версія програми платна. Ще один мінус програми – не дуже зручний інтерфейс та відсутність перекладів українською, англійською мовами.

До недоліків додатку можна віднести:

- ❖ розраховуються тільки витрати;
- ❖ немає перекладу англійською або українською мовами;
- ❖ застарілий інтерфейс;
- ❖ не простий у використанні;
- ❖ Розробник не оновлював додаток більше року;
- ❖ - дуже дрібний шрифт у пункті «Статистика».

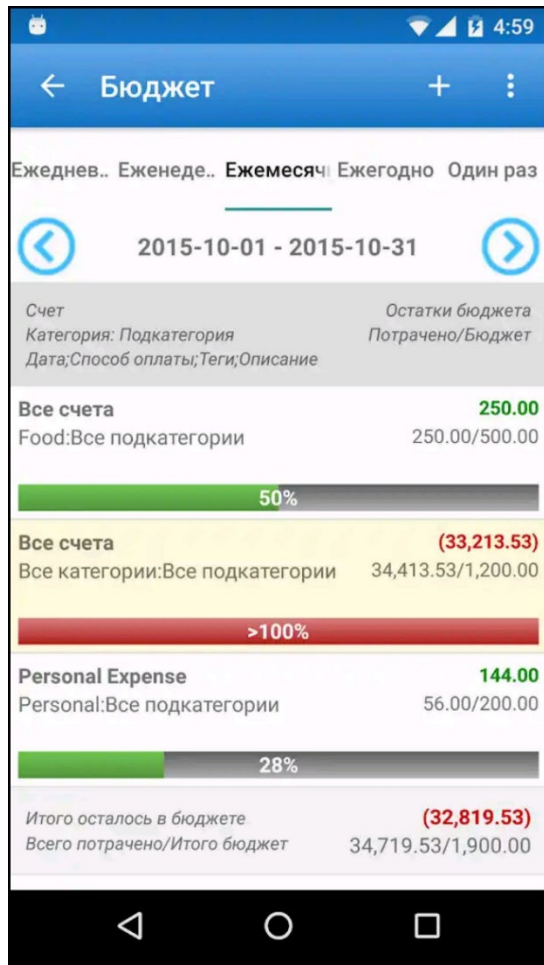


Рисунок 1.1 – Интерфейс додатку Expense Manager

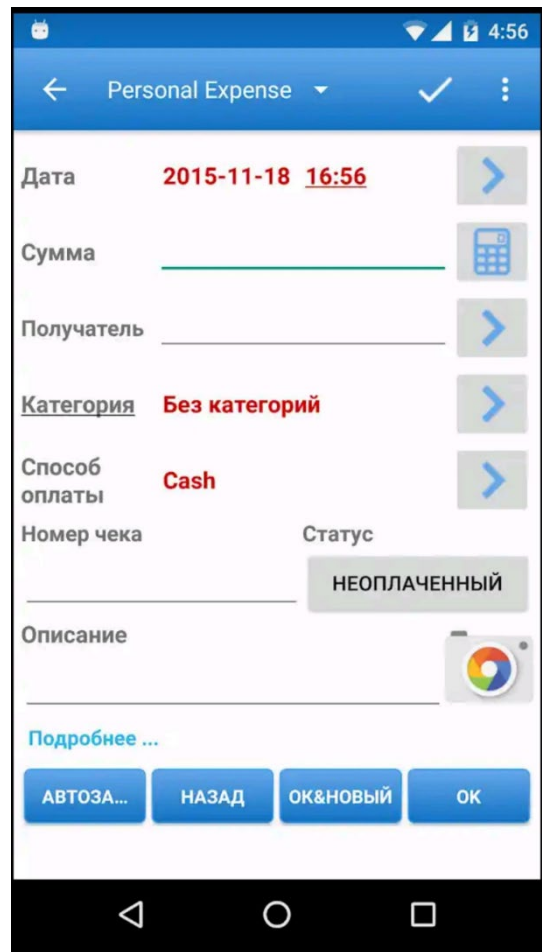


Рисунок 1.2 – меню витрат додатку Expense Manager

1.3.2 Додаток Andro Money

AndroMoney – розширений інструмент для персонального фінансового обліку. Додаток дозволяє записувати свої власні доходи, витрати та перекази в різних валютах, відстежувати борги та доступні ліміти кредитних карток, створювати бюджети тощо. Великою перевагою цієї програми є можливість створювати необмежену кількість гаманців, які називаються обліковими записами в AndroMoney. Вони можуть бути трьох видів: готівка, кредитна та банківська картки. Однак для рахунків карток, підключених до гаманця, немає функцій. Корисні функції базової версії програми включають збереження бази даних програми на Google Drive. Дані легко відновлюються у разі втрати даних або зміни пристрою.

Зм.	Арк.	№ документа	Підпис	Дата

Для початку роботи у додатку треба для кожної категорії обрати бюджет, тобто максимальну суму на витрати. Після того, як введете місячний бюджет, власний річний, тижневий і щоденний бюджети з'являться одразу. Існує функція перевищення бюджету, за допомогою якої можна встановити поріг бюджету (відсоток), який також відобразатиметься у спливаючому повідомленні. Додаток підвищує рівень безпеки фінансового обліку за допомогою пароля, який можна включити в налаштування. Також можете змінити тему калькулятора, колір фону програми, встановити в налаштуваннях початок місяця потрібне число, дату та формат числа тощо.

Недоліком цієї програми є те, що функціонал програми все ж таки досить обмежений в будь-якій версії. До недоліків також відноситься інтерфейс, який важко зрозуміти. У додаток можна авторизуватись за допомогою облікового запису Google, який був синхронізований з телефоном під час першого входу, але не зрозуміло як його змінити згодом.

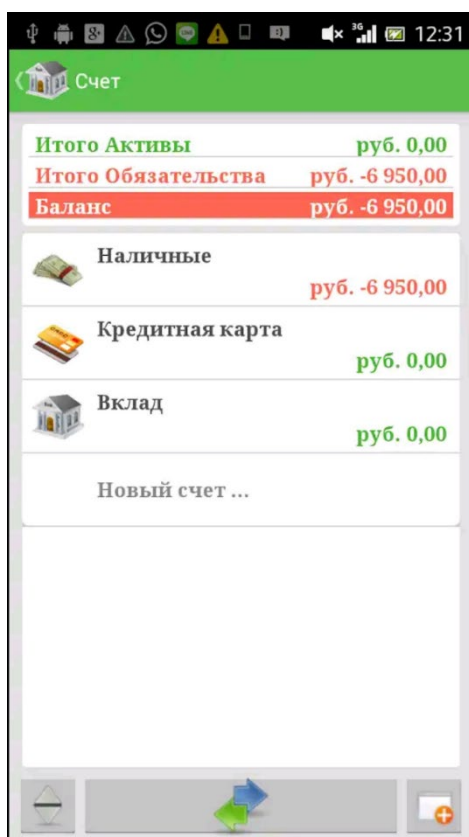


Рисунок 1.3 – Интерфейс додатку
Android Money

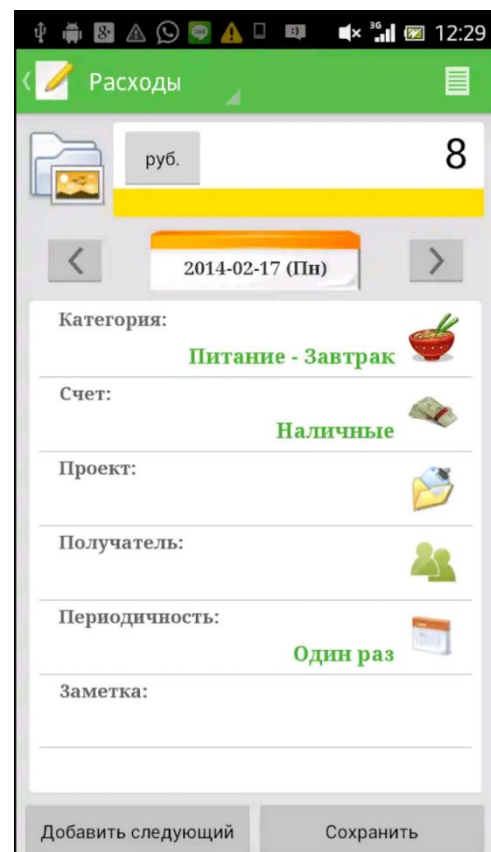


Рисунок 1.4 – меню витрат додатку
Android Money

Зм.	Арк.	№ документа	Підпис	Дата

1.3.3 Money pro

Money Pro з'явився в середині 2018 року і був розроблений компанією iBear LLC. Найважливішим фактором та те що виділяє додаток серед інших є унікальна логіка обліку валютних операцій. Програма частково безкоштовна, але є дві платні підписки - Pro і Gold.

Базові функції:

- ❖ Сортування витрат по певній класифікації
- ❖ Нагадування про оплату
- ❖ Зміна профілю у версії Family Budget Pro:
- ❖ 1500 іконок
- ❖ Будь-який часовий інтервал для фінансового звіту
- ❖ Можливість переносу залишку на наступний місяць
- ❖ Особливості депозиту/позики у версії додатку Gold
- ❖ Можливість підв'язки банківської картки
- ❖ Широкий інструмент для налізу витрат
- ❖ Певна роздільність усього капіталу на категорії (тобто готівка, картки і так далі)

Що стосується економічної оцінки додатків CoinKeeper на ринку, то версія Gold є найдорожчою з усіх програм такого роду. Профі-версія має обмежені можливості, але конкурентоспроможну ціну на ринку.

Додаток Money Pro взагалі був розроблений достатньо давно й набрав певну аудиторію користувачів й оновлюється дуже часто, що звісно дійсно є великою перевагою, це доволі рідко серед такого типу додатків у Play Market. Користувачі ж пишуть відгуки та роблять свій вклад у функціонал.

Інтерфейс цієї програми сильно відрізняється від своїх аналогів, складний і включає:

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		18

- ❖ Головний екран (бюджет)
- ❖ сьогодні
- ❖ рівновага
- ❖ Звіт
- ❖ налаштувати

Головний екран «Бюджет» являє собою статичну таблицю з трьох клітинок. Перший повністю складається з таблиці виплат, другий — це динамічний масив полів виплат, а третій — депозит.

Таблиця виплат - звичайний екран із календарем та областю, що показує надходження до гаманця для вибраної дати.

Залишки – це динамічна таблиця, де кожна клітинка є певний залишок конкретної категорії.

Звіт — це звичайний екран із круговою діаграмою витрат і кнопкою вибору періоду витрат.

Екран налаштувань — це динамічна таблиця з великою кількістю налаштувань, таких як валюта за замовчуванням, експорт, сповіщення, паролі. Щодо інтерфейсу програми він дещо перевантажений, та й дизайн не можна назвати сучасним. На рисунку 1.5 показаний головний екран та реалізований розподіл капіталу. Функціонал програми Money Pro має іншу логіку, ніж всі аналоги для запису грошових операцій, тут є автоматичне підтягування транзакцій з банківського рахунку, що є дійсно корисно. З економічної точки зору ця програма є дуже дорогою в порівнянні з іншими програмами, які будуть розглянуті в наступних підпунктах. Але у програмі знову ж відсутня або українська або англійська мови, лише російська, яка зрозуміла далеко не кожному користувачу. Загальна оцінка цієї програми по відгукам та функціоналу достатньо висока, що робить цю програму дуже популярною серед інших аналогів.

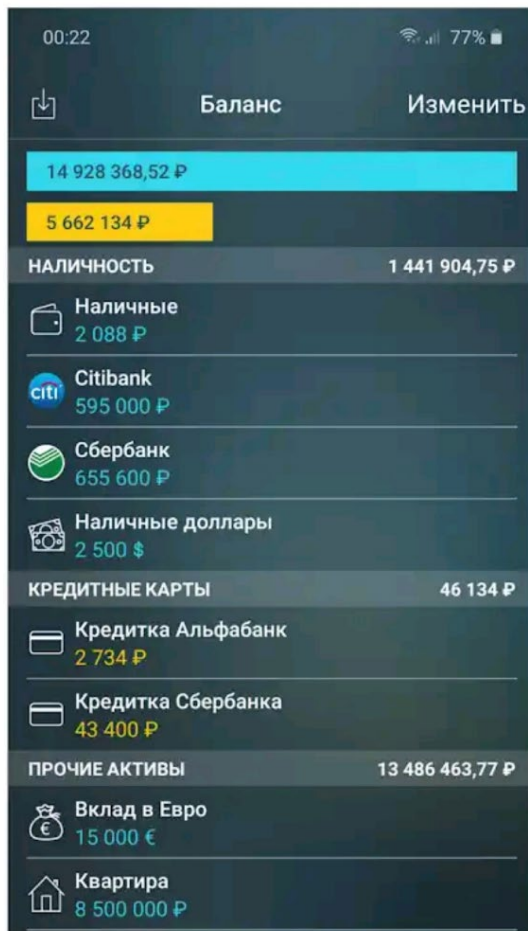


Рисунок 1.5 – Интерфейс додатку Money Pro



Рисунок 1.6 – меню рахунків додатку Money Pro

1.3.4 Monify

Робота в додатку починається з домашньої сторінки. Вона доволі інформативна та зручна. На цій сторінці є дві кнопки «Дохід» і «Витрати», тут же можна й додавати їх, сортуючи. Надалі ці кнопки будуть позначені «+» і «-». Також можна натиснути на одну із відображених категорій, щоб подивитись усі витрати по ній. Кожна категорія має наглядно свою іконку. У додатку присутня зручна різнокольорова діаграма і взагалі інтерфейс є різнобарвним та цікавим. У додатку показуються два рахунки одночасно, за ними можна ділити усі витрати, це - готівка та платіжна картка. Профіль не може мати синхронізації банківського рахунку, тож автоматика немає.

Коли створюєте новий обліковий запис, для нього обирається спеціальний значок, тобто іконка профілю.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		20

Спочатку зазначалося, що цей значок певним чином вплине на тип облікового запису, але це не так. Можливо, це є перевагою саме платної версії, бо у звичайні версії аккаунти користувачів мають ідентичні функції. Відбуваються лише між обліковими записами в додатку. Доходи та витрати поділяються на певні зазначені самою програмою категорії, які взагалі не можна видалити або змінити якимось на більш зручні індивідуально. Є функція створення капіталу. Капітали відображаються на спеціальній вкладці, яка об'єднує всі операції по рахунку. Суть полягає в тому, що користувач встановлює суму витрат щомісяця, а у вкладці буде наявно видно якщо бюджет перевищено.

Загалом інтерфейс програми зручний та сучасний, щодо функціоналу, він за замовчуванням обмежений самою програмою, як, наприклад, ті ж самі категорії.

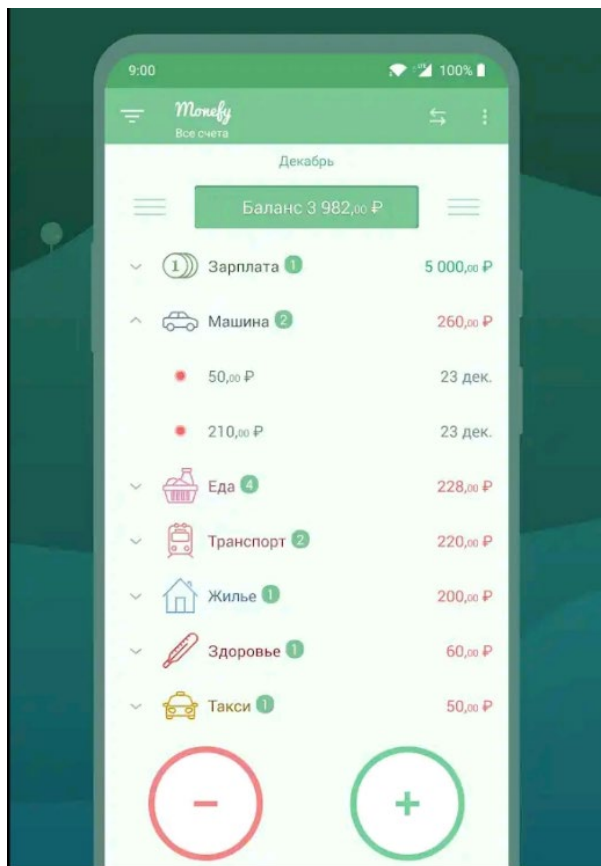


Рисунок 1.7 – Інтерфейс додатку Monify

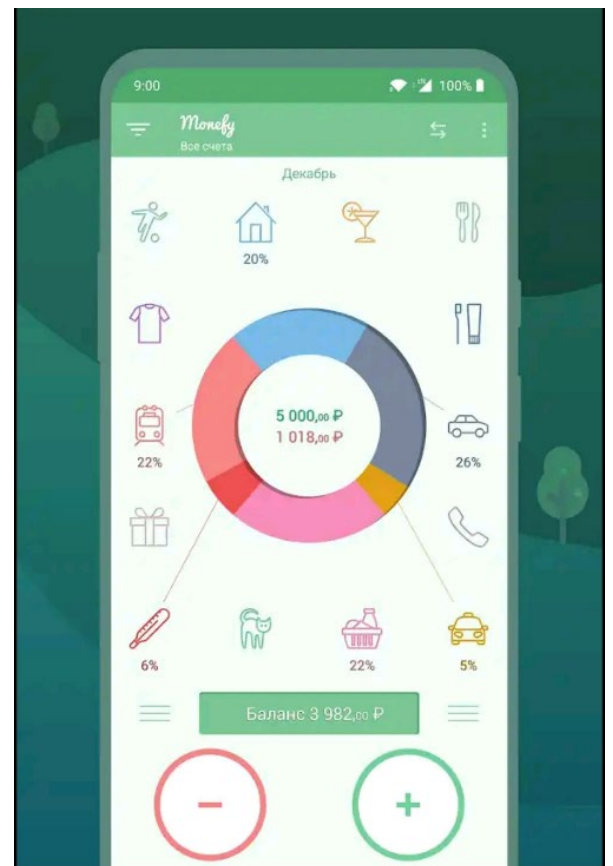


Рисунок 1.8 – меню рахунків додатку Monify

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		21

Висновки до розділу 1

У цьому розділі описується предметна область, а її головна мета — запровадити інструмент, що полегшує та допомагає користувачам керувати своїм бюджетом. Також описано потребу в мобільних програмах для особистого фінансового обліку як засобу заощадження та саморегулювання. Зроблено опис аналогів, визначені їх функції, розказано про інтерфейси. Визначено недоліки існуючих рішень, що й буде взято до уваги при розробці.

Можна підсумувати, що розроблено досить багато різних додатків, які надають змогу контролювати доходи та витрати. Більшість з них дозволяють відмічати свої грошові надходження та трати, деякі з них навіть мають певну автоматизацію через банківські рахунки.

В загальному, проаналізовані програми мають як недоліки, так і переваги. Однак слід зазначити, що більшість переваг з точки зору інструментів пов'язані з функціями платних програм, а деякі програми одразу ж за замовчування обмежують користувача в базових функціях.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		22

РОЗДІЛ 2

АНАЛІЗ ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1. Про Android

У цьому розділі розглянемо можливі засоби для реалізації додатку, його інтерфейсу, опишемо ОС Android, її поширеність та як з нею працювати. Більшість людей на планеті мають смартфони та девайси, на яких встановлена операційна система Android. Пристрої Android не просто працюють, а й роблять життя користувачів легшим. Особливості операційної системи дозволяють уникати заторів за допомогою GPS-навігатора, відправляти текстові повідомлення на годинник і отримувати відповіді на свої запитання від помічника. На базі Android працює близько двох з половиною мільярдів різних пристроїв, від телефонів із підтримкою 5G до просунутих планшетів. [13]

Android — це операційна система, відкрита для всіх: розробників, дизайнерів і виробників пристроїв. Це означає, що велика кількість людей має можливість експериментувати, придумувати революційні ідеї та втілювати їх у життя. [12]

Пристрої Android мають вбудовані інструменти безпеки. Усі програми перевіряються службою безпеки Google Play, а програмне забезпечення та платформа регулярно оновлюються та вдосконалюються. [1] Система безпеки Android завжди наготові – без перерв і вихідних. На своєму пристрої під управлінням Android ви можете вирішувати, до яких даних отримати доступ і коли. Якщо програма спробує отримати доступ до географічних даних, коли ви цього не знаєте, ви отримаєте повідомлення. Тепер роздільну здатність можна легко змінити, оскільки всі налаштування конфіденційності зосереджені на одній сторінці. У всіх пунктах конфіденційності остаточне рішення залишається власником цих даних.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		23

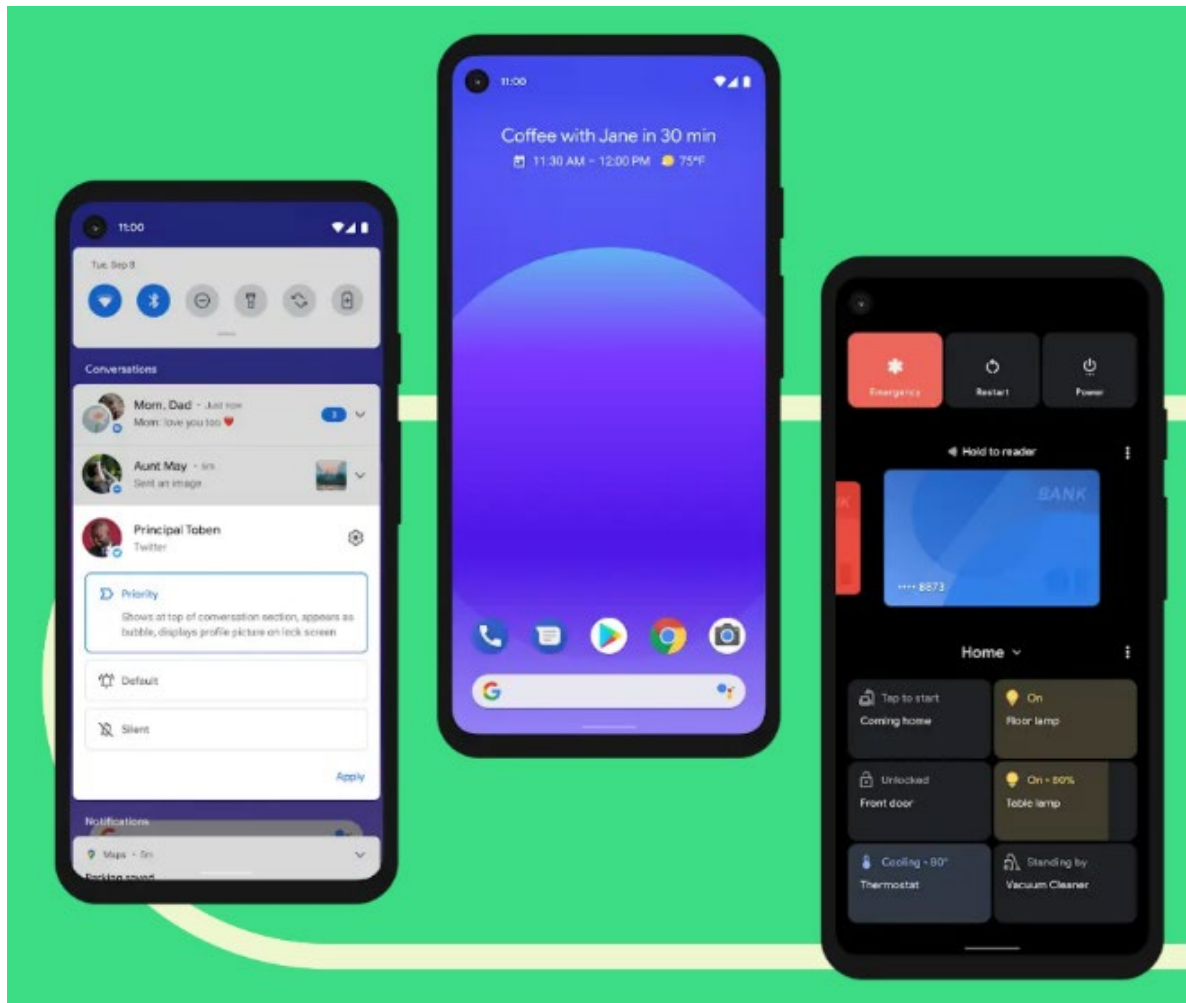


Рисунок 2.1 – інтерфейс операційної системи Android

Що стосується популярності цієї операційної системи, то вона вражає. У травні 2010 року було досягнуто ліміту в 100 000 активацій на день. Грудень 2010 року, 300 000 пристроїв активовано на день. У травні на конференції Google I/O оприлюднили статистику. За статистикою, щодня активується близько 400 тис. нових пристроїв на платформі Android. У липні 2011 року Енді Рубін, віце-президент Google з розробки платформи Android, оголосив, що перевищено новий ліміт у 500 000 активацій на день, при цьому платформа розповсюджується на 4,4% на тиждень. Статистика активації включає лише реєстрацію вперше для нових пристроїв, які мають пакет програмного забезпечення Google. Пристрої деяких азіатських виробників, чий програмне забезпечення не включає набір Google, не включаються в статистику.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		24

Станом на липень 2011 року понад 100 мільйонів пристроїв Android, було продано 36 виробниками та розповсюджено в мережах 215 операторів зв'язку. Загальна кількість моделей пристроїв на платформі Android досягла 310, а продано понад 200 мільйонів пристроїв Android. У каталозі Google Play понад 200 000 додатків було на той час. Загалом із Google Play встановлено близько 4,5 мільярдів додатків.

За даними Google, у вересні 2015 року було активовано 1,4 мільярда девайсів. Зараз ця цифра становить понад 2,5 мільярда різних пристроїв, що є дійсно вражаючим.

2.2 Архітектура Android

Android – це набір відкритого програмного забезпечення для мобільних пристроїв від компанії Google, до складу якого входить операційна система та комплект базових міжплатформних програм. Для розробки програм під платформу Android знадобиться набір інструментів та бібліотек API — Android SDK.

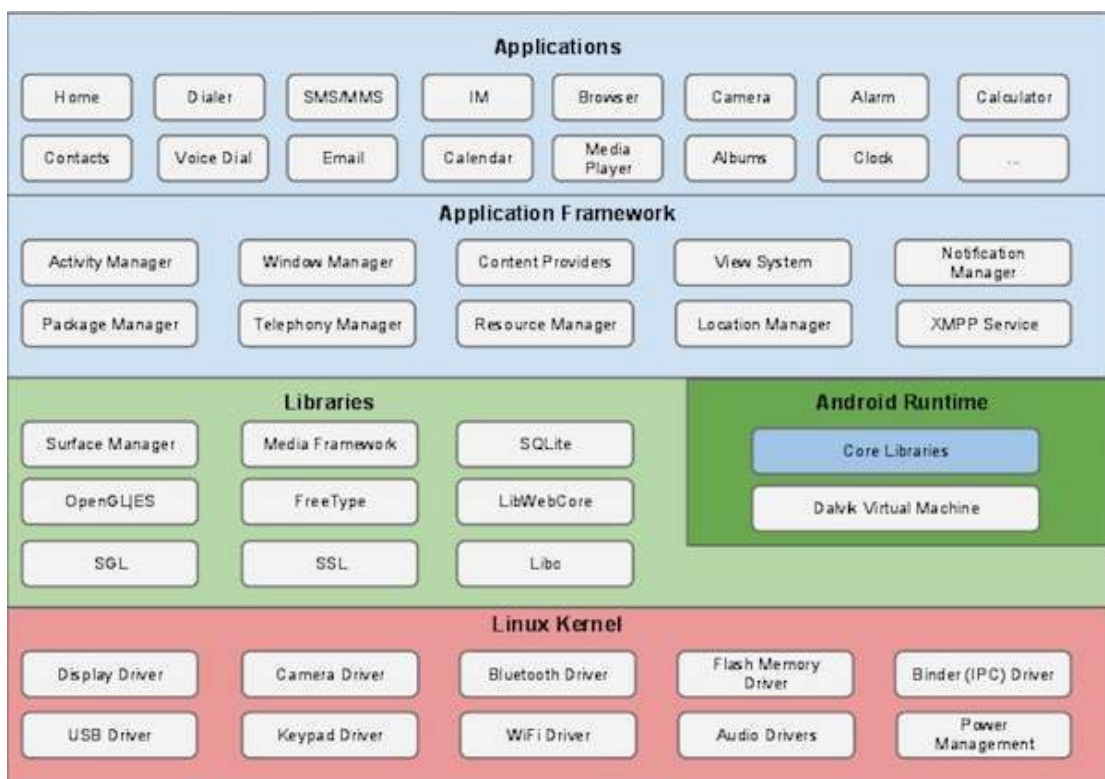


Рисунок 2.2 – Архітектура Android

Рівень додатків Android дозволяє використовувати всі функції API, які використовуються в програмах ядра. Архітектура побудована таким чином, що будь-яка програма може використовувати функціональні можливості, вже реалізовані іншою програмою, за умови, що вона має доступ до використання цих функціональних можливостей.

Таким чином, архітектура реалізує принцип багаторазового використання компонентів і додатків операційної системи. [9]

Основою всіх додатків є набір систем і сервісів:

1. Система перегляду (View System) – багата колекція візуалізацій з багатьма функціями, які можна розширити, для створення зовнішнього вигляду програми, включаючи такі компоненти, як списки, таблиці, поля введення, кнопки тощо.

2. Постачальники контенту (Content Providers) – це послуги, які дозволяють програмам отримувати доступ до даних інших програм і надають доступ до власних даних.

3. Менеджер ресурсів (Resource Manager) призначений для доступу до рядків, графіки та інших типів ресурсів.

4. Менеджер сповіщень (Notification Manager) дозволяє будь-якій програмі відображати повідомлення користувача в рядку стану.

5. Менеджер дій (Activity Manager) керує життєвим циклом програми та забезпечує систему навігацією по історії дій.

Платформа Android включає в себе набір бібліотек C/C++, які використовуються різними компонентами ОС.

Для розробників доступ до функціональних можливостей цих бібліотек досягається за допомогою фреймворків додатків.

Ось деякі з них:

1. Бібліотека System C – реалізація BSD стандартної системної бібліотеки C (libc) для вбудованих пристроїв на базі Linux.

2. Медіа-бібліотека – бібліотека на основі OpenCORE PacketVideo, розроблена для підтримки відтворення та запису популярних аудіо- та відеоформатів (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG тощо).
3. Surface Manager – менеджер поверхонь контролює доступ до шарів 2D і 3D графіки підсистеми відображення.
4. LibWebCore – сучасний механізм веб-браузера, який надає всі функції вбудованого браузера Android.
5. SGL - 2D графічний движок.
6. 3D бібліотека - движок для обробки 3D-графіки, заснований на OpenGL ES 1.0 API.
7. FreeType - бібліотека, призначена для роботи зі шрифтами.
8. SQLite - потужний движок для роботи з реляційними базами даних.

Android включає в себе набір бібліотек ядра, які забезпечують більшість функціональних можливостей бібліотеки ядра Java.

Платформа використовує оптимізовану, орієнтовану на реєстр віртуальну машину Dalvik, на відміну від стандартної віртуальної машини Java (орієнтованої на стек). Кожна програма виконується у своєму власному процесі з власним екземпляром віртуальної машини. Dalvik використовує формат Dalvik Executable (*.dex), який оптимізовано для мінімального використання пам'яті програмами. Це забезпечується основними функціями ядра Linux, такими як потокова передача та низькорівневе керування пам'яттю. Байт-код Java для написання програми компілюється у форматі dex за допомогою утиліти dx, що входить до складу SDK.

Android базується на Linux версії 2.6, що дозволяє платформі використовувати системні служби ядра, такі як управління пам'яттю та процесами, безпека, мережа та драйвери.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		27

2.3 Розробка Android-застосунків

Додатки для мобільних девайсів вже давно є основою дистанційної допомоги клієнтам і надання їм послуг. Вони активно замінюють повнофункціональні веб-додатки по багатьом критеріям. Це пов'язано з різноманітними чинниками, головний з яких – динамічність сучасності, коли все потрібно робити швидко, мобільні пристрої завжди під рукою, а ось повнорозмірні комп'ютери використовуються набагато рідше. Зараз розробка триває на двох основних платформах – Android та iOS. Кожна платформа має свої особливості, але загалом функціональність досить схожа. Написання додатків для цих платформ дозволяє надати найкращий можливий UX, що відповідає поточним рекомендаціям для кожної платформи, але доведеться кодувати програму двічі, щоб програма була доступна для Android та iOS. Щоб не писати двічі, існують популярні кросплатформні рішення.

При розробці програми необхідно враховувати багато факторів. У додатках повинні бути правильно оформлені елементи контролю з урахуванням правил використання та інших технічних характеристик, необхідних для швидкої та якісної роботи.[8] Дизайн мобільних додатків має багато функцій і досить сильно відрізняється від усіх інших сфер візуального дизайну. Щоб розробити якісний дизайн, необхідно знати основну мову, якою пишуться такі програми, і лише дотримуючись усіх правил, можна розраховувати на успішну розробку та якість свого мобільного додатка. Нативна розробка для Android означає, що програма доступна лише для Android. Є можливість користуватися як Android Studio так і Visual Studio для програмування в цьому середовищі, розробленому для системи Android. Продуктивність буде оптимізовано для пристроїв Android, Зовнішній вигляд користувальницького інтерфейсу відповідатиме іншим нативним програмам на пристрої, а всі функції пристрою користувача будуть перенаправлені для використання.[14]

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		28

Така розробка додатків відповідає всім шаблонам сумісності та стандартам користувача, встановленим спеціально для пристроїв Android.

2.4 Android SDK

Android SDK – це додатковий набір інструментів Android Studio, які допомагають написати код, запустити тестування та налагодження, перевірити роботу програми на різних версіях операційної системи та оцінити результат у реальному часі. Також пакет дозволяє користувачам отримувати інформацію про стан операційної системи, читати логи та виявляти помилки. Через SDK для Андроїд можна відновлювати програмну оболонку та встановлювати сторонні прошивки. Набір складається з пакетів, необхідних для створення програм. Ось основні, якими можуть користуватися розробники на Android .

Android SDK Platform Tools - до групи входять такі інструменти взаємодії з Android як Android Debugging Bridge (ADB), Fastboot, Systrace та інші. ADB допомагає знайти помилки у роботі додатків, встановити APK на смартфон. [7] Fastboot - активувати швидке завантаження для керування мобільним пристроєм з комп'ютера, перепрошувати гаджет, налаштувати доступ, параметри роботи операційної системи. Systrace — отримати інформацію про запущені процеси, простежити за активністю та обсягом даних, які надіслані через мережу.

Android SDK Build Tools - компоненти SDK використовуються для створення коду. Zipalign дозволяє оптимізувати файл APK, AAPT2 – проаналізувати, проіндексувати та скомпілювати ресурси у двійковий формат під платформу Android, Apksigner – підписати пакет APK за допомогою закритого ключа.

Емулятор Android - інструмент допомагає протестувати програми та випробувати функції останніх версій Android. Детальний опис Android SDK та необхідну документацію можна знайти на офіційному сайті в розділі User guide.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		29

Android Studio з набором Android SDK доступна на пристроях з операційними системами Windows, Mac, Linux та Chrome OS. Для запуску комп'ютер повинен відповідати мінімальним системним вимогам.

Таблиця 2.1 – Вимоги Android SDK

	Windows	Mac	Linux	Chrome OS
Версія	8/10 64 bit	10.14(Mojave)	Gnome, KDE,Unity DE	Версія 3.5
RAM	8 gb	8 gb	8 gb	8 gb
Місце на диску	8 gb (IDE + Android SDK+Emulator)	8 gb (IDE + Android SDK+Emulator)	8 gb (IDE + Android SDK+Emulator)	4 gb
Мінімальна роздільна здатність	1280 x 800	1280 x 800	1280 x 800	1280 x 800

2.5 Фреймворки розробки

Мало того, що Android домінує на ринку смартфонів, операційна система використовується ще й у планшетах, автомобілях, телевізійних приставках, розумному годиннику, побутовій техніці та у вигляді емуляторів для персональних комп'ютерів. Початківцям Android-розробникам часто доводиться гадати, який фреймворк краще для їхньої ідеї. Щоб зрозуміти функціонал, переваги та недоліки основних фреймворків для Android оглянемо їх.

2.5.1 Flutter

Flutter – це розроблений Google фреймворк з відкритим програмним кодом, який дозволяє просто і швидко створювати мобільні додатки для iOS і Android. [5]При цьому в роботі Flutter не використовує нативні компоненти.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		30

Натомість усі елементи інтерфейсу користувача у фреймворку створюються за допомогою власного графічного движка. Flutter дозволяє створювати всі елементи інтерфейсу користувача вашого додатка з готових віджетів. У цьому плані Flutter схожий на інші фреймворки - React і Vue, але має багато відмінностей від них. Так, він не використовує мову програмування Javascript, але Flutter використовує мову Dart.

Програми Flutter можуть працювати як на iOS, так і на Android, і інструмент нібито відхиляється від традиційного представлення в фреймворках інтерфейсу користувача для обох операційних систем. Flutter не компілюється безпосередньо в програми iOS та Android. Вони працюють на поєднанні механізмів візуалізації C++ і фреймворку Flutter на Dart. Згенеровані файли прикріплюються до кожної програми, а SDK збирає готові програми для певних платформ. [5]

Такий підхід називається розробкою програм для відеоігор. Подібно до того, як ігри майже ігнорують весь програмний фреймворк і перекладають основну роботу на свій движок, додатки на Google Framework замінюють елементи Flutter у всіх можливих частинах рідного фреймворка. Від цього страждає компактність програми, але така програма працює швидко.

Розробка для кількох платформ одразу за допомогою Flutter не означає падіння якості програмного продукту. Фреймворк надає всі власні віджети для інтерфейсів Android (Матеріальний дизайн) і iOS (Купертіно) і сам може змінювати поведінку окремих елементів, роблячи додаток більш придатним для звичайного користувача.

На додаток до міжплатформних можливостей, Flutter також надає окрему компіляцію файлів у режимі розробки. Компілятор JiT не є компілятором AoT для кінцевого продукту, але дозволяє швидше створювати та налагоджувати програми.

Крім того, Google Framework має підтримку більше 800 плагінів включаючи SQLite, Firebase, Facebook Connect і GraphQL. Він дозволяє поєднувати код Flutter з елементами стандартної мови розробки додатків, специфічної для платформи. Основним елементом програми Flutter є віджет. Загалом, це основний загальний елемент з якого складається інтерфейс.

Початок роботи Flutter розпочинається завжди з функції `main()`. Створення графічного інтерфейсу в цій функції називається іншою вбудованою функцією - `runApp (Widget app)`.

Він приєднує віджет до екрана. Це в основному те, що ми бачимо, коли запускаємо програму на екрані пристрою. Приєднаний віджет передається як аргумент функції `runApp()`.

Усі віджети успадковуються від класу `Widget`. Деякі віджети можуть включати інші віджети, утворюючи багатошарову систему у вигляді дерева. Віджети показують, які мають бути елементи інтерфейсу користувача, враховуючи їх поточну конфігурацію та стан.

Коли стан віджета змінюється, він змінює свій опис, що ще більше відрізняє кадр від попереднього опису. Під час розробки програмного забезпечення часто роблять нові віджети, які є підкласами `StatelessWidget` або `StatefulWidget`, залежно від того, чи контролює віджет будь-який стан.
[5]

Основна мета та задача віджета — зробити функцію `build()`. `StatelessWidget` рекомендується для закріплених віджетів. Це віджети без внутрішнього стану, які покладаються лише на параметри конфігурації та основний віджет. `StatefulWidget` необхідний, коли існує більше одного внутрішнього стану, і вони здатні змінювати один іншого.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		32

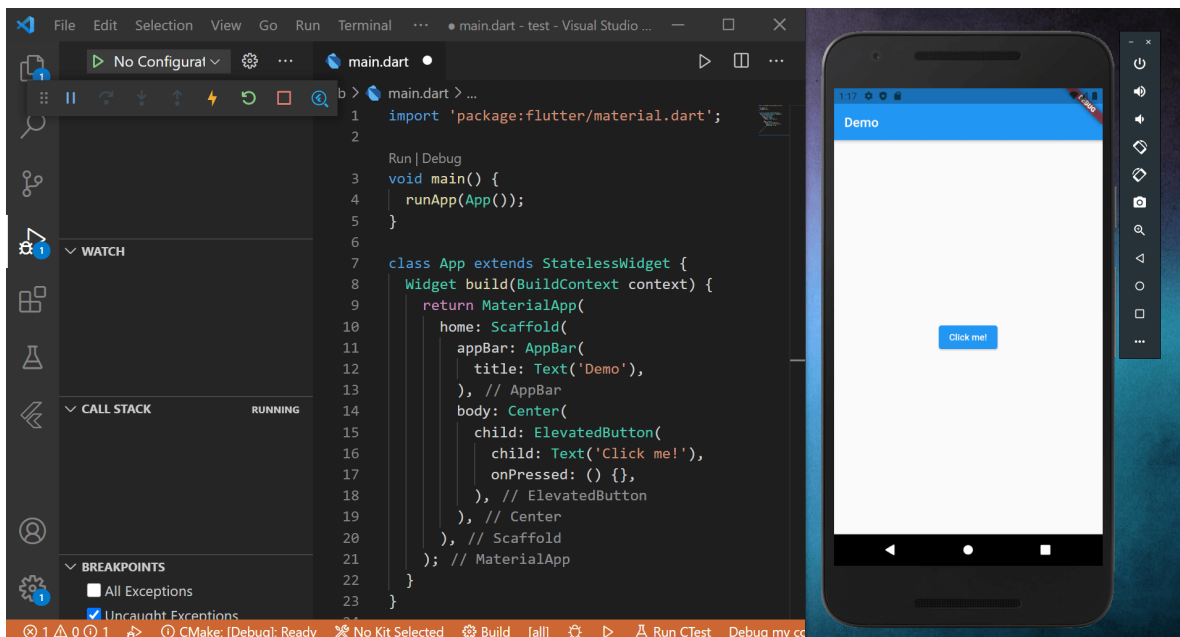


Рисунок 2.3 – Інтерфейс Flutter

2.5.2 React Native

React — це інструмент для створення інтерфейсів користувача. Його головне завдання — забезпечити, щоб на екрані відображалось те, що можна побачити на веб-сторінці. React дозволяє легко створювати інтерфейси, розбиваючи кожну сторінку на невеликі фрагменти та компоненти. React популярний з кількох причин. Він компактний і продуктивний, особливо коли має справу зі зміною даних. Програми, створені на основі React Native, мають саме такий вигляд завдяки власним компонентам інтерфейсу. Також, програмісти можуть швидко оновлювати створені сторінки, не чекаючи навіть фіналу власних збірок. React Native-фреймворк з доступним кодом, розроблений був у 2015, у розробці брала участь компанія Facebook. React Native розроблений на базі React, поширеної бібліотеки JavaScript для розробки у Web, яка використовує власний інтерфейс та вбудовані компоненти свого API для забезпечення високої продуктивності та швидкого розвитку.

						ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата			33

Програми React Native не є мобільними веб-програмами, бо React Native керується тими ж компонентами, як і повсякденні розробки під iOS та Android. Ці компоненти скомпільовані не на Swift, Kotlin чи Java, а на JavaScript та React Компоненти можуть посилатися на інші компоненти дерева компонентів. Це дозволяє використовувати одну і ту ж абстракцію - компоненти на будь-якому рівні.

Додатки, написані на React з нуля, зазвичай мають компонент App у верхній частині. Усі програми React побудовані на основі компонентів. Це зазвичай є прості функції та класи JavaScript, проте їх можна брати як редаговані частини HTML. Кнопки, розділи, меню та інші елементи інтерфейсу можуть бути створені як складові. Ці компоненти також зберігають інформацію про статус і розташування екрана. Компоненти зазвичай створюються за допомогою JSX. Це корисний посібник під час роботи з інтерфейсами користувача в коді JavaScript. JSX — це симбіоз мов HTML і JavaScript.

2.5.3 Xamarin

Xamarin є одним із найпопулярніших інструментів розробки кросплатформних додатків для iOS, Android, tvOS, watchOS, macOS та Microsoft, використовуючи C# та .NET.

На відміну від інших кросплатформних програмних рішень, Xamarin.Forms недоступний у веб-каналах. Натомість він створює власні високопродуктивні елементи керування за допомогою SDK для Android або iOS.

У проектах зі складними інтерфейсами вартість вдосконалення іноді може конкурувати з розробкою двох або трьох нативних рішень. Але бізнес-логіка та сама, і кожен нову версію програми буде легше розробляти.

					ІАЛІЦ.467200.003 ПЗ	Арк.
						34
Зм.	Арк.	№ документа	Підпис	Дата		

Алгоритми, розроблені на C# для однієї платформи, використовуються програмістами на всіх інших платформах.

Наприклад, до 75% коду можна використовувати повторно під час роботи на Xamarin.Android і Xamarin.iOS: додаток працює однаково на смартфонах iPhone і Android.

Якщо основний зміст у додатку - функціональність, а дизайн лише з кількома кнопками і приємною графікою - Xamarin ідеально підходить.

Інтерфейс програми Xamarin.Forms складається з об'єктів, які зіставляються з власними елементами керування кожної цільової платформи.

Це дозволяє програмам для платформи iOS, Android та універсальної платформи Windows використовувати код Xamarin.Forms, що міститься в бібліотеці .NET Standard.

2.5.4. JavaScript

JavaScript – це погано типізована мова. Це означає, що не потрібно заздалегідь визначати тип змінної.

Тип буде визначено автоматично під час виконання програми. Це теж значить, що змінною можна користуватись для накопичення різних типів даних.

Змінні оголошуються за допомогою основних слів var або let. var більше не використовується.

Стиль Dart такий же, як і Dart. JavaScript — це багатопарадигмальна мова програмування. Підтримуються об'єктно-орієнтований, імперативний та функціональний стилі.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		35

Таблиця 2.2 - Типи даних JavaScript

Тип	Опис
Function	Підвид Object, який спрощує визначення типу для функцій
String	послідовність з символів
Object	Використовується для зберігання даних і для створення інших структур
null	Невизначений тип
Symbol	символ
BigInt	вбудований об'єкт, який надає спосіб представлення цілих чисел, які більші за максимальне число типу
Number	числа
Boolean	логічне значення, має два значення: true і false

2.5.5. Dart

Змінні використовуються для зберігання даних у Dart, як і в інших мовах програмування. Стиль нижнього регістру використовується для іменування змінних, тобто ім'я змінної починається з малої літери. Якщо ім'я змінної складається з двох або більше слів, перше слово починається з малої літери, а всі наступні слова починаються з великої літери. Тип даних визначає, які значення може зберігати об'єкт. Dart має багато вбудованих типів. Загалом, Dart — це об'єктно-орієнтована мова. Усі значення, що використовуються в програмах Dart, представляють об'єкти.

В час започаткування та покращень Dart взяв впливу від ранніх мов, таких як Smalltak, Java, JavaScript тощо. Його синтаксис подібний до інших мов С.

Таблиця 2.3 – типи даних Dart

Тип	Опис
Map	колекція, яка зберігає інформацію ключ-значення
Set	колекція унікальних елементів
List	список - набір значень або ж просто масив
Runes	символи в кодуванні UTF-32
Bool	має два значення: true і false
String	послідовність з символів
double	з плаваючою комою
Int	цілочислове

2.5.6. С#

С# — це об'єктно-орієнтована мова програмування, розроблена у 1998-2001 роках Microsoft групою інженерів як мова розробки програм для Microsoft .NET Framework. Мова програмування С# використовує синтаксис, дуже схожий на С і близький до С++ і Java. Типи даних особливо важливі в С#, оскільки вони є строго типізованими мовами. [2]Це означає, що всі операції підлягають постійному контролю компілятора на відповідність типу, а недійсні операції не компілюються. Тому суворий контроль типів може усунути помилки та підвищити надійність програми. Щоб забезпечити контроль типів, усі змінні, вирази та значення повинні мати певний тип.

Коли ми говоримо про С#, ми зазвичай маємо на увазі технології, пов'язані з платформою .NET. Те ж саме відбувається і в протилежному напрямку, коли справа доходить до .NET, подумайте про С#. Звичайно, між поняттями існує суворий зв'язок, але не синоніми. С# — це мова, розроблена для взаємодії з .NET Framework.

Таблиця 2.4 – Типи даних в С#

Тип	Опис
Object	значення будь-якого типу
String	послідовність з символів
decimal	десяткові
Ulong	цілі додатні(8 байт)
Float	число з плаваючою комою (4 байта)
Double	число з плаваючою комою (8 байт)
Bool	логічне значення
Char	символ
Long	цілі (8 байт)
UInt	цілі додатні (4 байта)
Int	цілі (4 байта)

2.5.7 Порівняння фреймворків для розробки

Кожен фреймворк має багато переваг і недоліків, виграним вибором є фреймворк Flutter. Він підтримує велику кількість платформ та має дуже обширний функціонал. Але кросплатформність часто негативно впливає на роботу самої програми, десь працює швидко та плавно, а десь може навіть не запускатися, хоча з економічної точки зору це дуже доцільно.[11]

Таблиця 2.5 – Порівняння деяких фреймворків

Характеристика	React Native	Flutter	Xamarin
Open source	Так	Так	Має платні пакети
Спільнота	Дуже велика	Невелика	Велика
Продуктивність	Висока, близька до нативної	Дуже висока	Висока, близька до нативної
Підтримувані платформи	Android, IOS	Android, IOS, Google, Desktop, WEB	Android, IOS, UWP
Реліз	2015	2017	2011
Розробник	Facebook + Спільнота	Google + Спільнота	Microsoft
Мова розробки	JavaScript+React	Dart	C#, .NET

2.6 Бази даних

Більшість мобільних розробників, ймовірно, знайомі з SQLite. Він існує з 2000 року і, можливо, є найбільш використовуваним механізмом реляційної бази даних у світі. SQLite має ряд переваг, які ми всі визнаємо, однією з яких є його вбудована підтримка на Android.

Популярні бази даних мобільних додатків:

- ❖ MySQL: багатопотокова і проста у використанні база даних SQL з відкритим вихідним кодом.
- ❖ PostgreSQL: потужна реляційна база даних з відкритим вихідним кодом, яка легко налаштовується.
- ❖ Redis: сховище ключів і значень з відкритим вихідним кодом, яке не потребує обслуговування, яке використовується для кешування даних у мобільних додатках.

SQLite - це база даних SQL із відкритим джерелом, яка зберігає дані у текстовому файлі на пристрої. Android поставляється з вбудованою базою даних SQLite. Існує багато способів збереження даних у настільному додатку. База даних - це один з основних способів. [3]

Можливо, доведеться надати програму встановлення, якщо не використовувати базу даних на основі файлів, наприклад SQLite.

Також можете просто записати у файл – текстовий файл, файл XML, поділяючи об'єкти тощо. Коли потрібно вибрати базу даних, основним питанням зазвичай є вибір реляційної (SQL) або нереляційної (NoSQL) структури.

Обидва варіанти мають свої переваги, а також кілька ключових характеристик, які слід враховувати при виборі.

Реляційні бази даних використовують мови структурованих запитів для визначення та обробки даних.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		40

З одного боку, це дає величезні можливості для розвитку: SQL є однією з найбільш гнучких і часто використовуваних мов запитів, тому його вибирають, щоб мінімізувати деякі ризики, і він особливо корисний, коли доводиться мати справу зі складними запитами.[10] SQL, з іншого боку, має багато обмежень.

Побудова запитів на цій мові вимагає визначення структури даних, і подальші зміни структури даних можуть завдати шкоди загальній системі.

Нереляційні бази даних, у свою чергу, забезпечують динамічну структуру даних, яку можна зберігати багатьма способами: орієнтовано на стовпці, на документ, на основі графіків або на основі пар.

У реляційних базах даних дані представлені у вигляді таблиць, а в нереляційних – у вигляді документів, пар ключ-значення, графіків або сховищ із широкими стовпцями. Це робить бази даних SQL кращим вибором для програм, які включають транзакції з кількома записами, наприклад системи облікових записів або застарілі системи, створені для реляційних структур.

SQLite - це окрема база даних без SQL-сервера. Це вбудована кросплатформна база даних, яка підтримує досить повний набір команд SQL і надає вихідний код.

Механізм баз даних — це бібліотека, яка складає програму, а SQLite стає частиною програми. Вся база даних зберігається в одному стандартному файлі на машині, на якій запущена програма.

Кілька процесів або потоків можуть без проблем одночасно зчитувати дані з бази даних. Увійти до бази даних можна лише в тому випадку, якщо на даний момент не обслуговуються інші запити; інакше спроба запису завершиться невдачею, і програмі буде повернуто код помилки.

Іншим варіантом розробки є автоматична повторна спроба ведення журналу через задані інтервали.

SQLite підтримує типи TEXT (емульований рядок у Java), INTEGER (емульований довгий у Java) та REAL (емульований подвійний рядок у Java). Решта типів необхідно конвертувати перед збереженням у базі даних. SQLite не перевіряє типи даних, тому можна записувати цілі числа в стовпці рядка і навпаки.

SQLite ретельно стежить за цілісністю даних, реалізуючи механізм зміни даних за допомогою транзакцій. Коротка примітка щодо торгів: угоди або виконуються повністю, або не виконуються зовсім. Проміжних станів взагалі не може існувати. [4]Якщо не користуєтеся транзакцією явно, завжди створюється прихований платіж. Він починається до виконання команди і зберігає дані, як тільки закінчиться. SQLite реалізує більшість стандарту SQL, але не підтримує моделі клієнт-сервер.

Таблиця 2.6 - типи даних SQLite

Тип	Опис
Blob	Масив бінарних даних
Real	Число з плаваючою комою
Text	Символ чи рядок
Null	Порожнє позначення
Integer	Ціле позначення

Висновки до 2 розділу

У цьому розділі ми дізналися про операційну систему Android, її поширення, функціонал та можливості, основні інструменти для створення мобільних додатків. Описано в загальному сенсі чому більшість людей на сьогоднішній день віддають перевагу мобільним додатку, на відміну від повноцінних програм на стаціонарних машинах.

Ми порівняли фреймворки розробки, доступні мови, архітектури та середовища та дізналися більше детально про кожен фреймворк, про платформи, які вони підтримують.

Розглянуто типи даних у більшості фреймворків. Також описано їх характеристики, у деяких відібрано переваги та недоліки. Описано одну з найпоширеніших баз даних та її особливості.

В загальному розумінні існує багато середовищ для розробки дизайн-проектів та функціоналу для мобільних та веб-додатків. Для програмного забезпечення також є безліч технологій на вибір.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		43

РОЗДІЛ 3

ДЕТАЛІ РЕАЛІЗАЦІЇ ДОДАТКУ

3.1 Загальні відомості по розробці

У цьому розділі буде розглянуто деталі реалізації додатку. Для виконання мети даної роботи, буде розроблено мобільний додаток для операційної системи Android, функціонал якого по можливості об'єднає функціонал додатків розглянутих у першому розділі. Розроблений додаток буде мати англomовний інтерфейс для розуміння та користування більш ширшим колом людей, тобто усіма тими, хто володіє англійською мовою. Набір функцій додатку буде в міру великий задля запобігання перевантаження інтерфейсу програми. За попередніми підрахунками додаток матиме 4-5 загальних основних функцій, що допоможуть користувачу в майбутньому заощадити власні кошти. Тож, підсумуємо майбутній загальний необхідний функціонал та можливості:

- ❖ Швидкий та безпроблемний запуск
- ❖ Інтерфейс англійською мовою
- ❖ Додавання категорій витрат
- ❖ Додавання надходжень до гаманця
- ❖ Розділення власних транзакцій по категоріям
- ❖ Сучасний інтерфейс
- ❖ Швидкість роботи
- ❖ Збереження інформації
- ❖ Додавання фінансових цілей
- ❖ Діаграма витрат по категоріям
- ❖ Сортування усіх транзакцій по категоріям

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		44

Загалом розробка додатку не такий легкий процес та займає немало часу, тож треба поділити усю розробку на основні пункти, по яким треба діяти.

Отже, виділимо основні етапи розробки :

- ❖ Розробка дизайну
- ❖ Розробка функціоналу програми
- ❖ Тестування

3.2 Розробка дизайну

AndroidStudio використовує мову розмітки XML високого рівня для макета програми.

Android має різні типи файлів XML:

- ❖ анімація;
- ❖ ресурсний;
- ❖ креслення;
- ❖ меню;
- ❖ Інші файли XML.

На рисунку 3.1 показані лише деякі ресурси. Система постійно оновлюється, додаючи нові типи. У будь-якій програмі або сучасному Android-додатку присутні файли типу XML.

Саме за допомогою них відбувається візуалізація програми та створення інтерфейсу. Це все у купі відповідає за враження користувача від інтерфейсу. А плавність роботи дозволяє виконувати операції у програмі більш ефективно.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		45

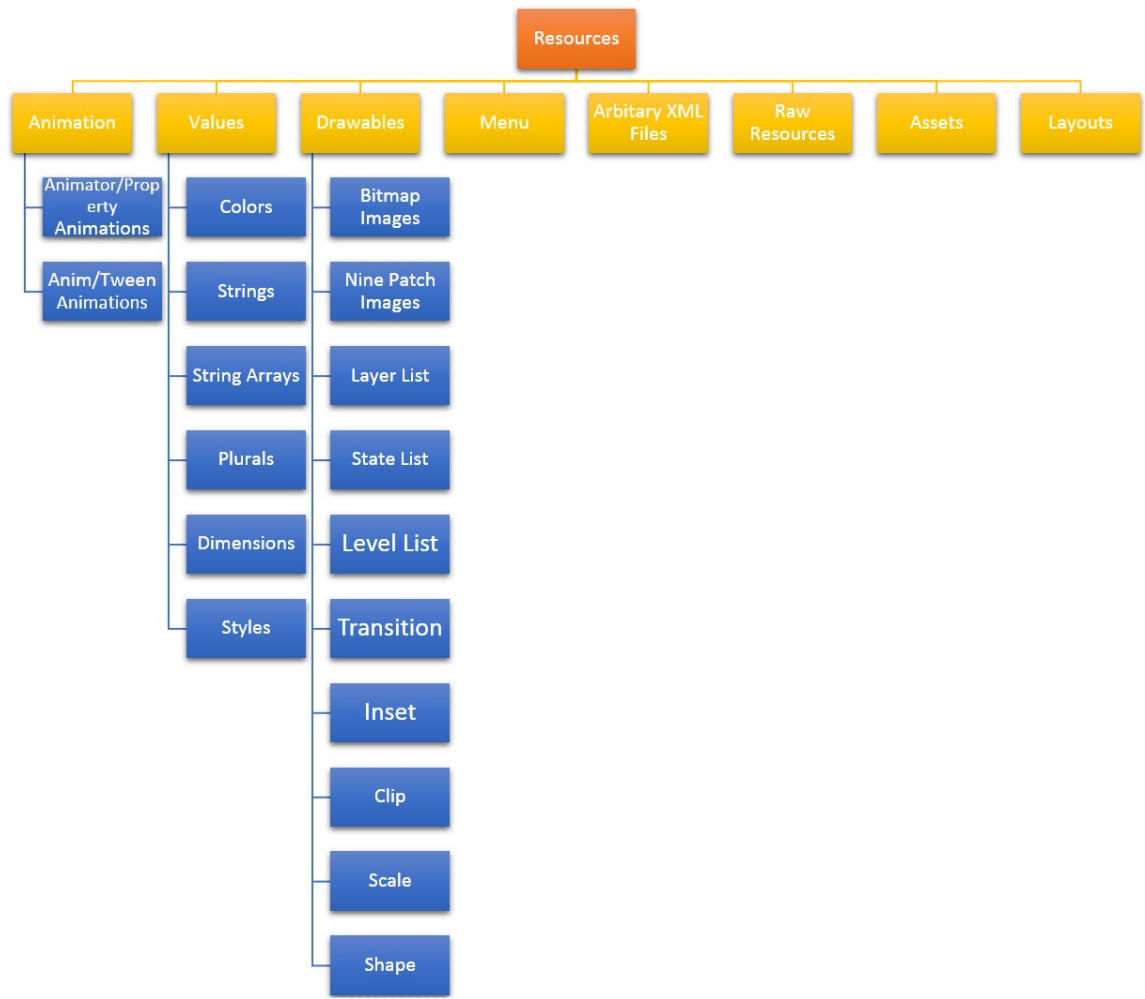


Рисунок 3.1 - Ресурси ОС Android

Ресурси в Android є декларативними. Переважна більшість ресурсів зберігаються в каталозі res у вигляді файлів XML зі значеннями підкаталогу, drawable-ldpi, drawable-mdpi, drawable-hdpi, layout, але є й інші типи ресурсів.

Найпоширенішими ресурсами є рядки, кольори та растрові зображення.

Програма не рекомендує використовувати жорстко закодовані рядки коду - замість цього використовуйте відповідні ідентифікатори, щоб змінити текст рядкових ресурсів, не змінюючи вихідний код.

Статичний клас R генерується на основі вказаних ресурсів і створюється під час компіляції проекту. Після створення клас містить статичні підкласи всіх типів ресурсів, у яких описано принаймні один екземпляр.

Ресурси меню можна використовувати для опису звичайних і контекстних меню. Меню, описані у форматі XML, завантажуються як програмні об'єкти за допомогою методу `inflate`, що належить до служби `MenuInflater`. Зазвичай це відбувається в обробнику `onCreateOptionsMenu`.

Існує п'ять стандартних типів макета:

- ❖ абсолютне планування;
- ❖ компоновка каркаса;
- ❖ Лінійне розташування;
- ❖ відносне розташування;
- ❖ Розкладка столу.

`AbsoluteLayout` (застарілий, застарілий) – вказує, що кожен елемент макета матиме абсолютну позицію щодо верхнього лівого кута екрана, встановлену за допомогою координат x і y . Тобто в `AbsoluteLayout` у верхньому лівому куті екрана є координати $x = 0, y = 0$.

Позиції вказуються в атрибутах елемента `android: layout_x` та `android:layout_y`.

Якщо елемент повинен займати більше однієї клітинки, використовуйте атрибут `android: layout_span`.

За замовчуванням `<TableRow>` організовує рядки таблиці, якщо потрібно організувати не рядки, а стовпці, там потрібно використовувати атрибут `android: layout_column`.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		47

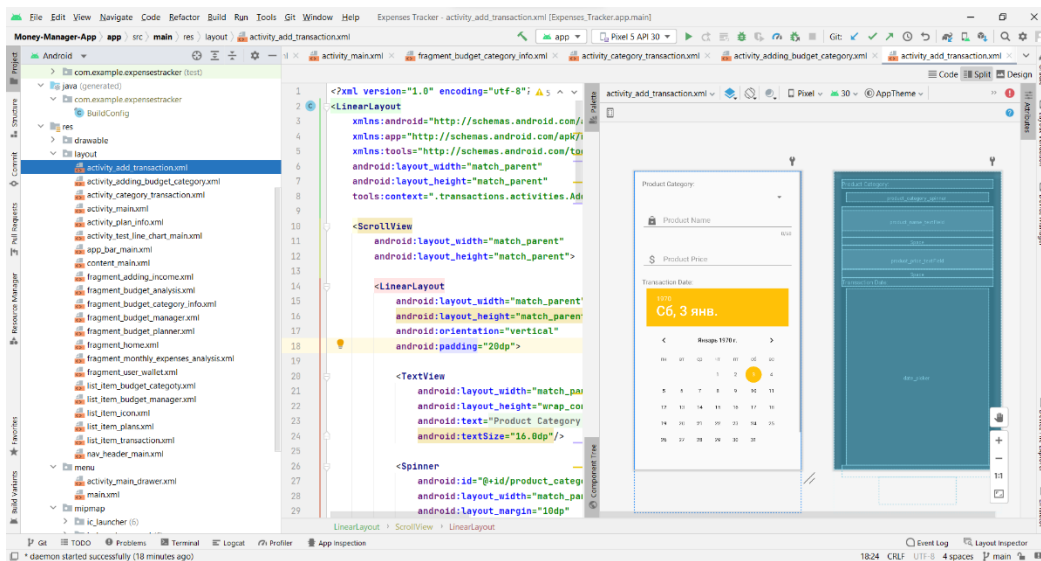


Рисунок 3.3 – Макет меню додавання транзакції

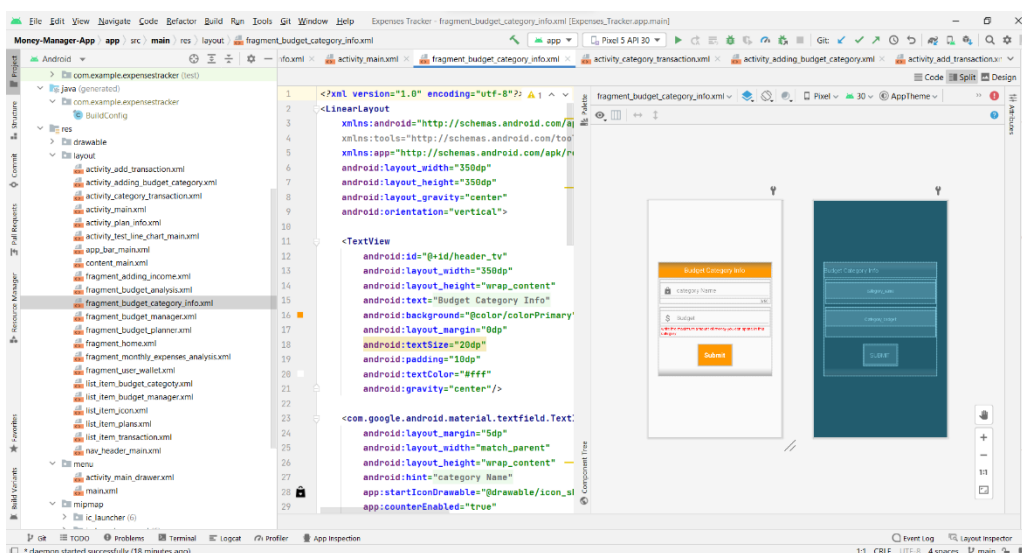


Рисунок 3.4 – Макет меню додавання категорії

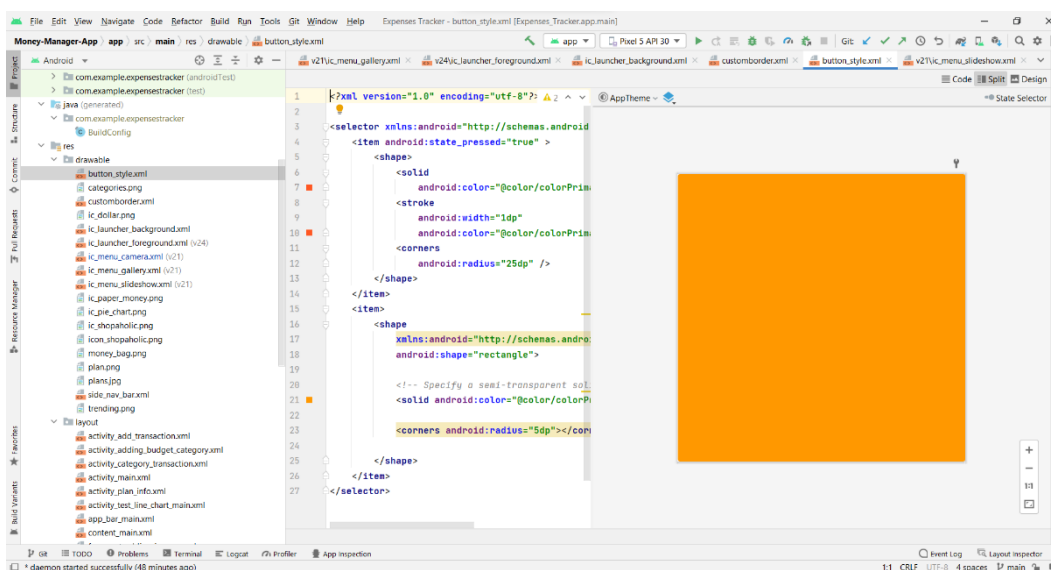


Рисунок 3.5 – Основний колір додатку

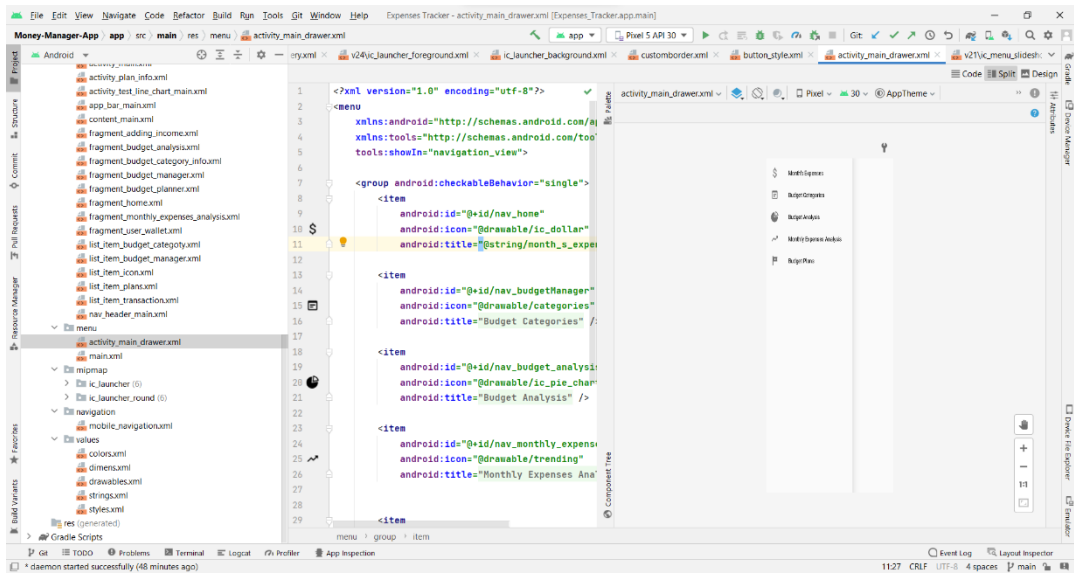


Рисунок 3.6 – Макет основного меню

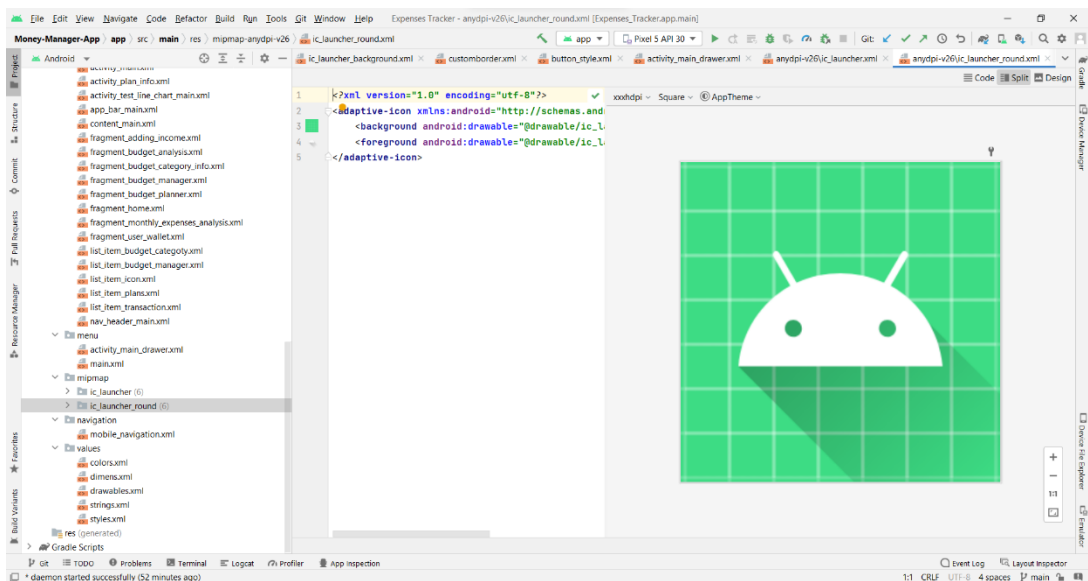


Рисунок 3.7 – Макет значка

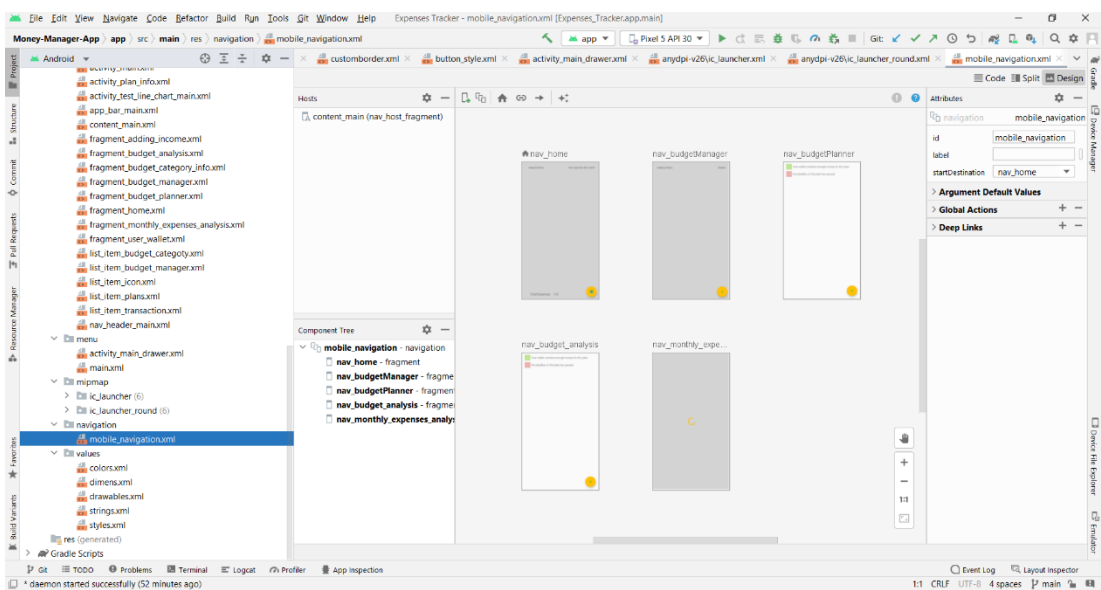


Рисунок 3.8 – Макет меню навігації

Зм.	Арк.	№ документа	Підпис	Дата
-----	------	-------------	--------	------

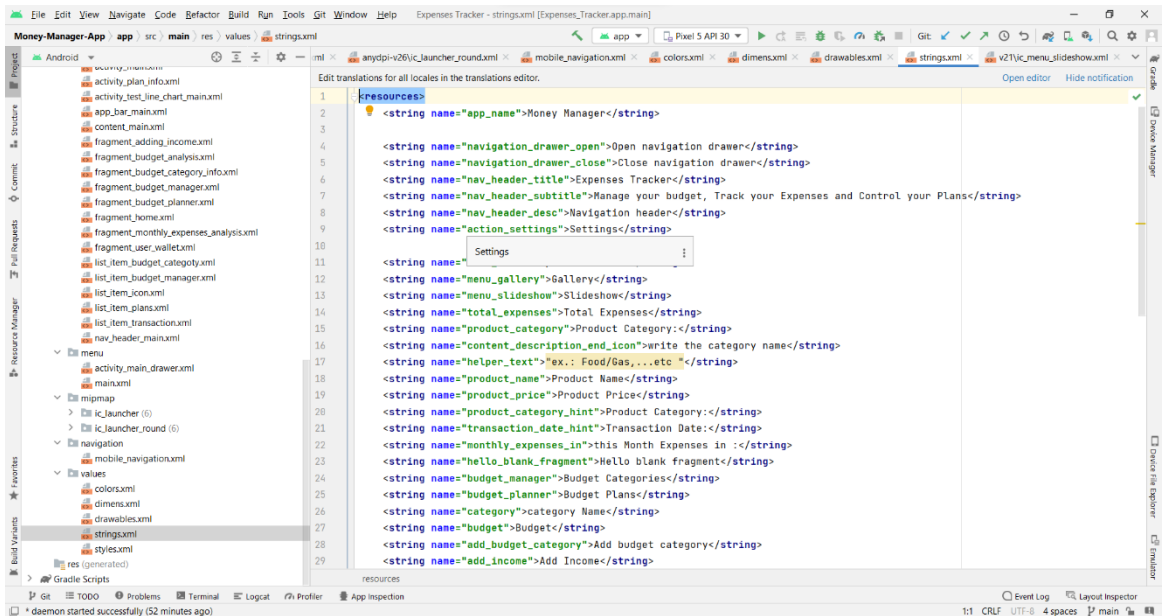


Рисунок 3.9 – Текст меню додатка

3.3 Розробка функціоналу

Розробка додатку для персонального фінансового обліку написана мовою програмування Java у програмному середовищі Android Studio на «чистому» Android SDK, що являє собою оригінальним інструментом для розробки. Структуру додатку містить окрема діаграма у кінці цього проекту. У ній показано головне меню (рисунок 3.10), яке містить:

- ❖ Місячні витрати
- ❖ Категорії бюджету
- ❖ Аналіз бюджету
- ❖ Аналіз місячних витрат
- ❖ План бюджету

Також головне меню помаранчевого кольору, як і уся програма переважно. В головному меню написаний девіз програми «Manage your budget, track your expenses and control your plans». Усі витрати у додатку будуть записуватися у валюті «долар» для більшого розуміння аудиторією. Будь-яку валюту світу можна конвертувати у долар та додати транзакцію по категорії у додатку, в якій би країні світу б не знаходилися.

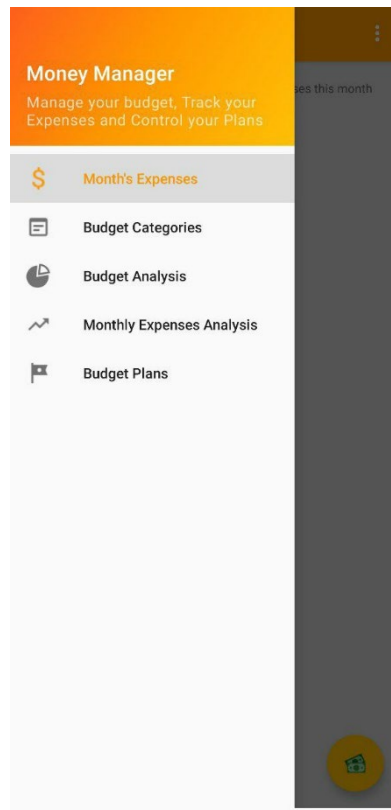


Рисунок 3.10 – Основне меню програми

Верстка графічного дизайну розроблена за допомогою розширеної мови розмітки XML, а функціональна частина - за допомогою мови програмування Java. Наступним кроком проведено з'єднання графічного інтерфейсу з функціональною частиною.

Отже, програма має широкий функціонал. Основа програми – це категорії витрат, на які і витрачається бюджет. Потім ідуть транзакції. Кожна транзакція належить до певної категорії. Наприклад, користувач сховався у кіно. Він відкриває мобільний додаток, додає транзакцію під назвою білет у кіно за категорією «Розваги», яку він додав раніше. У кожній категорії є власний бюджет. Коли певний бюджет для категорії перевищений, то про це повідомляється користувача у меню місячних витрат. Ця категорія обводиться червоним. Саме так можна зрозуміти що на цій категорії потрібно зекономити.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		52

Діаграми програми візуалізуються за допомогою бібліотеки AnyChart. AnyChart Android Charts — це чудова бібліотека візуалізації даних для легкого створення інтерактивних діаграм у програмах Android. Він працює на основі API 19+ (Android 4.4) і має десятки вбудованих типів діаграм.

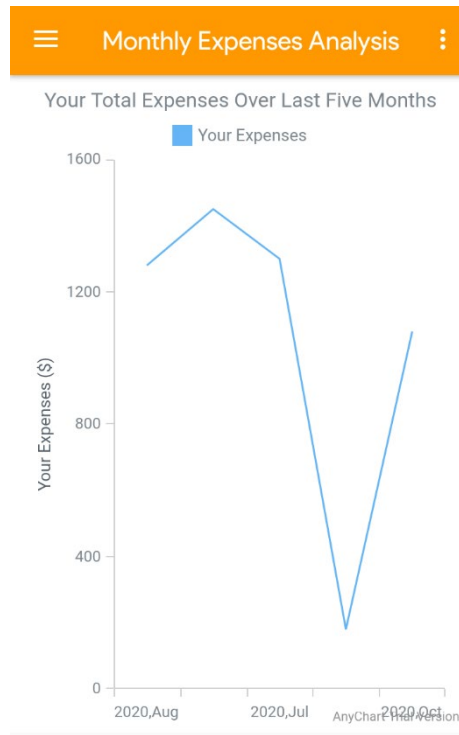


Рисунок 3.11 – Діаграм витрат по місяцям

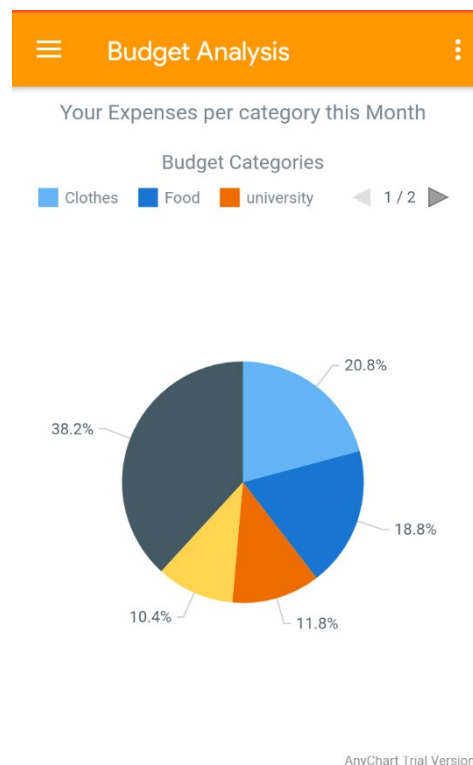
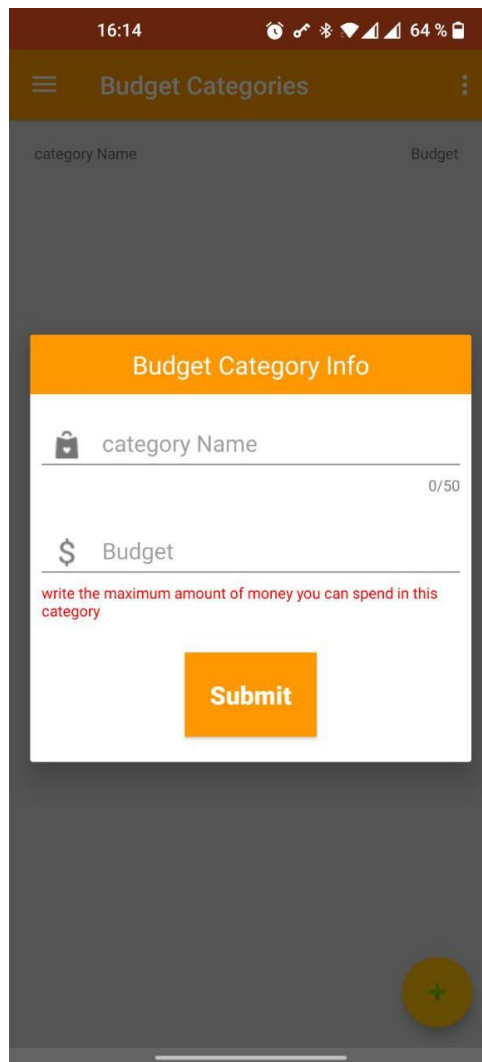


Рисунок 3.12 – Діаграма витрат по категоріям

3.4 Робота з додатком

Для початку роботи з додатком та обліку власних коштів, встановлення кількості витрат у місяць, впровадження фінансових цілей перш за все краще додати усі власні можливі категорії, на які витрачаєте власні гроші. Для цього потрібно зайти у додаток, потім у головне меню та обрати розділ «Budget Categories». Після цього унизу інтерфейсу можна побачити плюс, потрібно натиснути на нього. Відкриється меню. Тут треба вказати назву категорії та максимальну суму, яку згодні витратити на цю категорію в місяць.



The screenshot shows a mobile application interface for 'Budget Categories'. At the top, the status bar displays the time 16:14 and battery level at 64%. The app header is dark brown with a hamburger menu icon on the left and a vertical ellipsis on the right. Below the header, there are two labels: 'category Name' and 'Budget'. The main content area features a white card with an orange header 'Budget Category Info'. Inside the card, there is a text input field for 'category Name' with a lock icon on the left and a character count '0/50' on the right. Below this is another text input field for 'Budget' with a dollar sign icon on the left. A red warning message reads: 'write the maximum amount of money you can spend in this category'. At the bottom of the card is an orange 'Submit' button. In the bottom right corner of the app screen, there is a green circular button with a white plus sign.

Рисунок 3.13 – Меню додавання категорії

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		54

Обравши усі необхідні категорії, додаємо суму гаманця, для цього у правому верхньому куті натискаємо 3 крапки, обираємо меню «Add Income». Пишемо суму капіталу . Після нових надходження до гаманця діємо так само згодом. Статус гаманця можна подивитися натиснувши ті ж самі 3 крапки та обравши меню «Your Wallet». Для меню Settings поки функцій не розроблено, але згодом можливо щось буде розроблено.

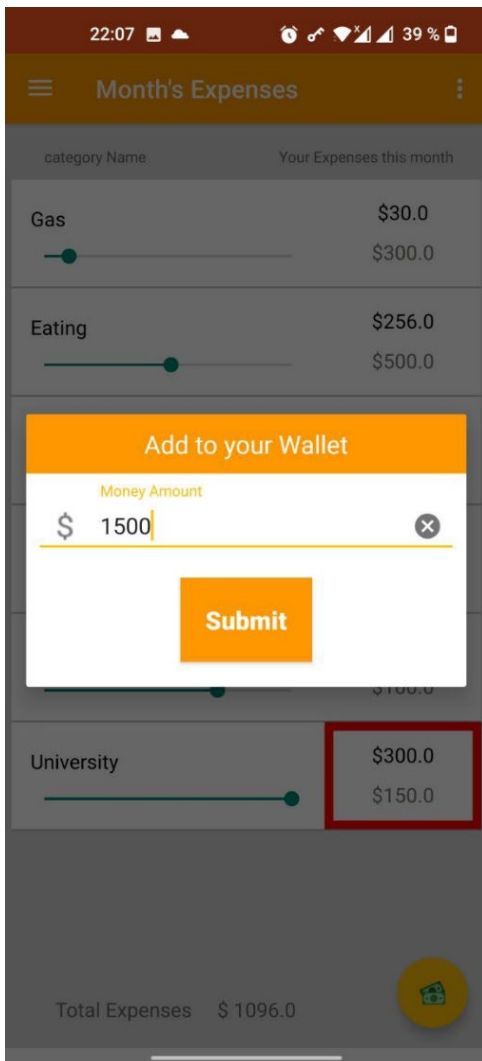


Рисунок 3.14 – Меню додавання суми гаманця

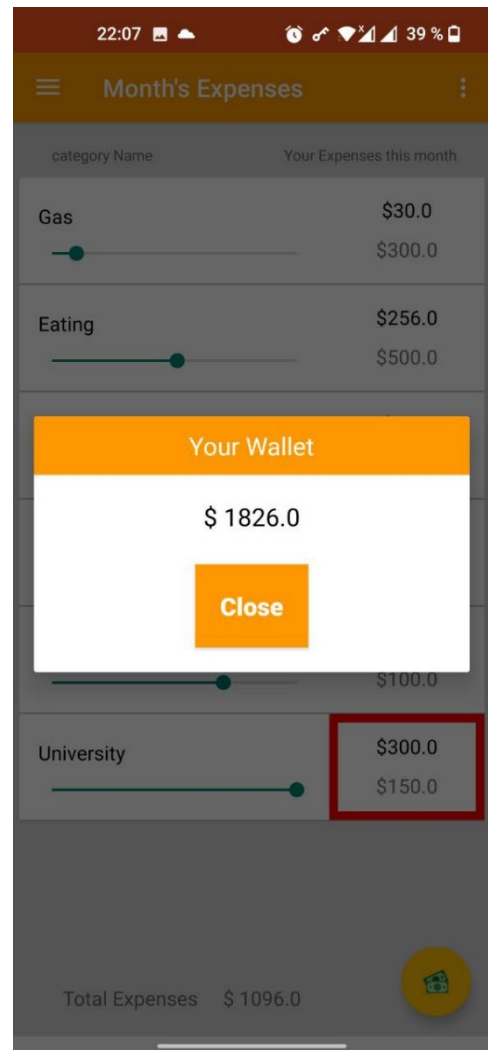


Рисунок 3.15 – Меню суми у гаманці

Зм.	Арк.	№ документа	Підпис	Дата

Зробивши попередні кроки, приступаємо до додавання транзакцій. Для цього необхідно відкрити меню «Month's Expenses», а потім натиснути на значок грошей у правому нижньому куті. Відкривається меню додавання транзакції, тобто витрати. Тут вводимо що саме придбано, або на що витрачено кошти, ціну, обираємо назву категорії, за якою ці гроші витрачено та дату цієї трати. Краще це робити, після кожної витрати одразу, в цьому і полягає суть програми, а інакше усі витрати згодом забуваються. Роблячи це на автоматі кожного дня, це діло становиться рутинним та не займає багато часу, а результат буде вражаючий. Згодом можна побачити, на що йдуть кошти. Навіть найменші дрібниці не залишаться без уваги.

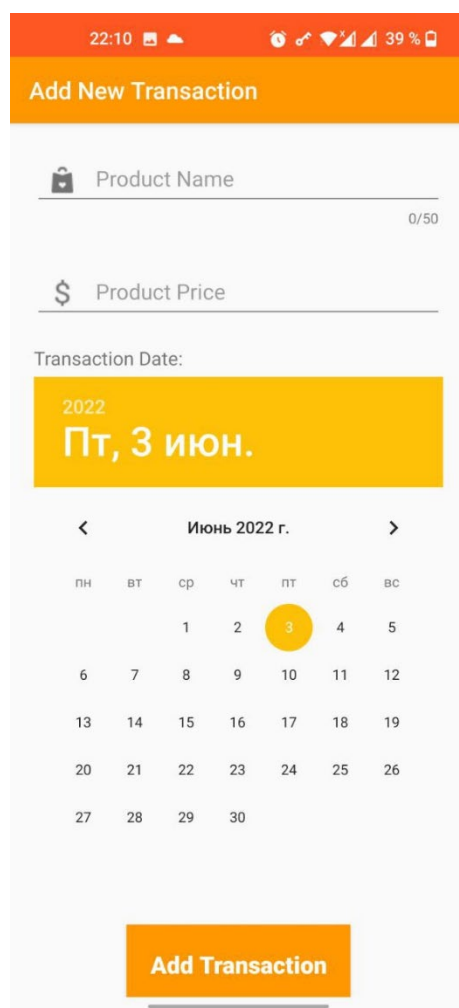


Рисунок 3.16 – Меню додавання транзакції

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		56

Також у меню «Month's Expenses» натиснувши на потрібну категорію можна побачити список усіх витрат по категорії. Про будь-яку витрату тут можна дізнатися чимало інформації, яка була введена до цього.

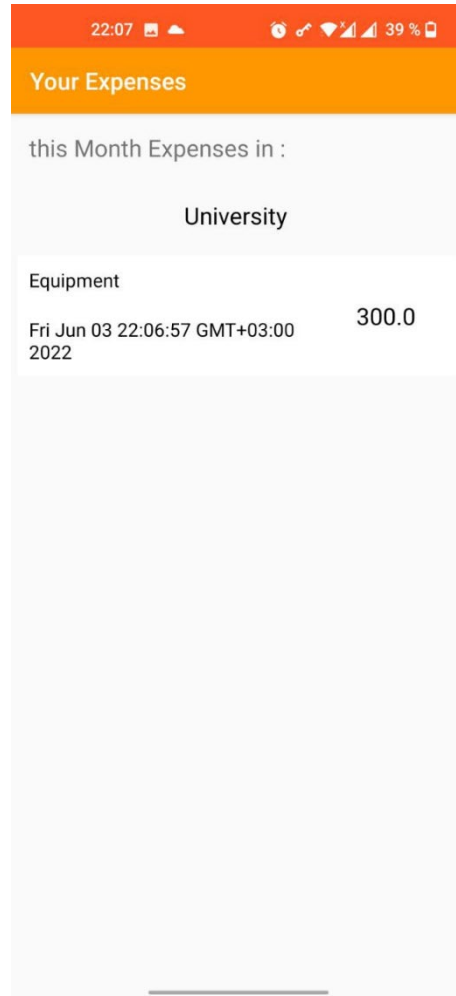


Рисунок 3.16 – Меню інформації про окремо взяту витрату

Користуючись програмою щодня, відриваючи її по декілька разів на день, вводячи дані про витрати, надходження, програми стає більш функціональною щодня. Не відриваючи кожну категорію можна побачити суму витрат по ній. Побачити де ця сума перевищує зазначений раніше бюджет на неї, а де ні. Саме про це повідомляє червоний прямокутник справа від назви категорії. (Рисунок 3.17)

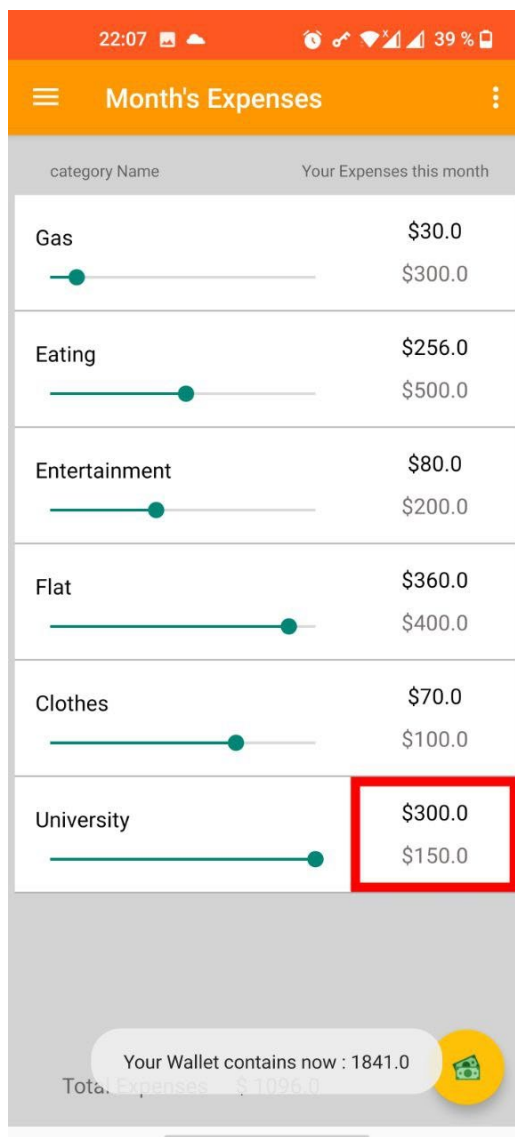


Рисунок 3.17 – Меню місячних витрат після місяця користування буде виглядати приблизно так

Отже, чим більше користуватися програмою, тим більша користь від неї. Наприклад уже у кінці місяця буде остаточно видно на що витрачаються кошти. Для цього потрібно перейти у розділ головного меню «Budget Analysis». Тут можна побачити наглядну діаграму витрати у відсотках витрат за місяць. (Рисунок 3.18) Вона відображає усі категорії на які витрачені були кошти.

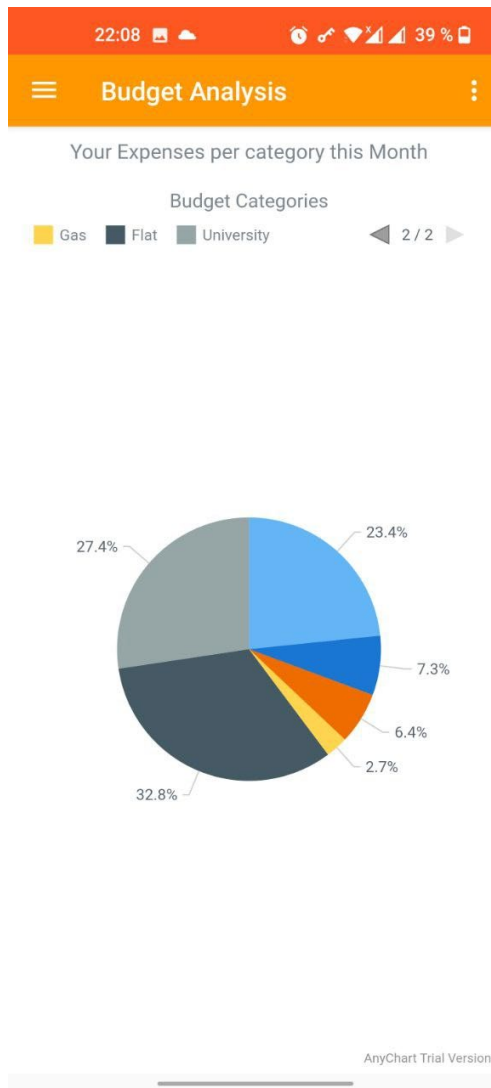


Рисунок 3.18 – Діаграма місячних витрат

У цьому додатку присутній функціонал додавання фінансової цілі. Для цього потрібно перейти до розділу головного меню «Budget Plans». Тут натискаємо плюс у правому нижньому куті, як вже було в одному з розділів та відкривається меню додавання бюджетного плану. (Рисунок 3.19) В цьому меню обираємо назву плану, пишемо опис, та вибираємо бюджет. Також вводимо дату, якої потрібно закінчити цей план. Далі додаємо його. У результаті отримуємо план. (Рисунок 3.20) Він порівнюється з сумою у гаманці. Якщо коштів вистачає, то підсвічується зеленим. Ні, червоним. Червоним також якщо пройшла зазначена дата.

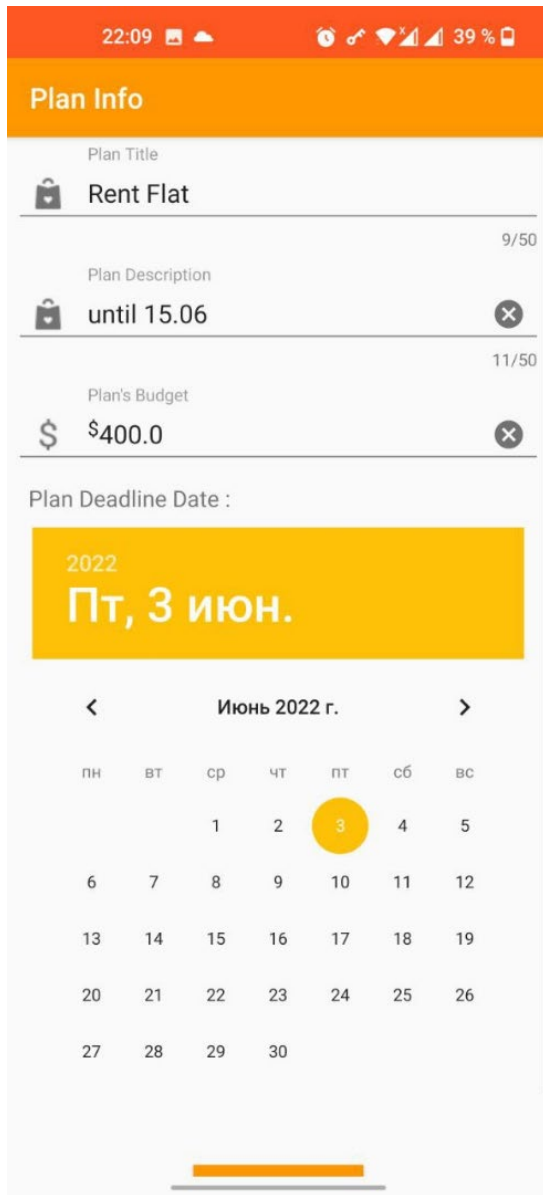


Рисунок 3.19 – Меню додавання плану

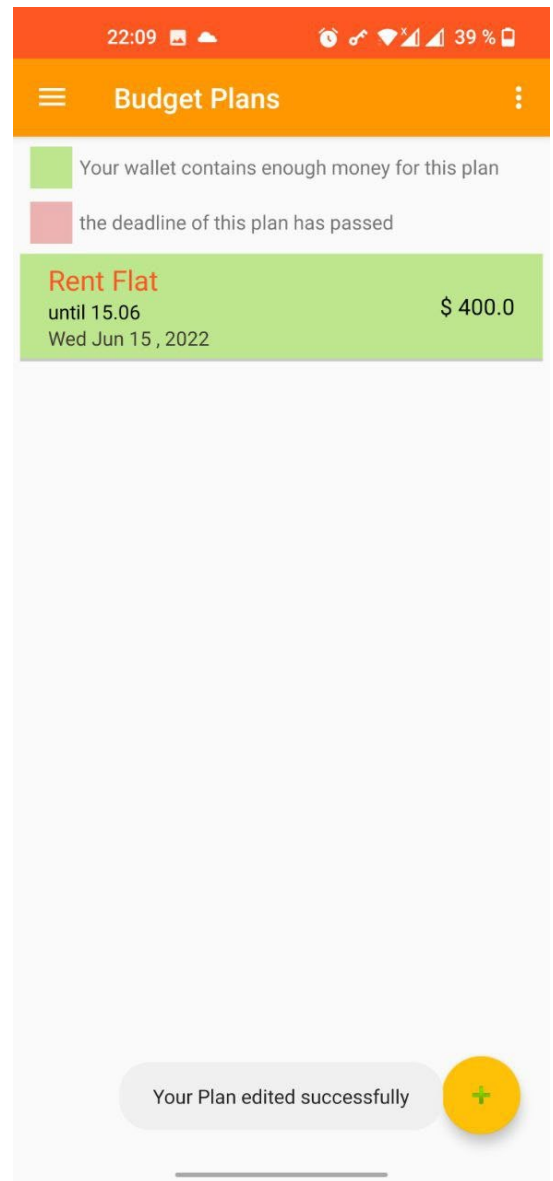


Рисунок 3.20 – Меню зазначених планів

Висновки до розділу 3

Розглянуто детальну взаємодію користувача з будь-якими можливими операціями у додатку, що відповідають за рух грошових коштів (витрату, надходження, бюджетні плани). Також написано коротку інструкцію щодо користування програмою, роботи з діаграмами.

Знімки екрана, надані в цьому розділі, показують, як програма працює під час змін, щоб налаштувати, використовувати категорії, шаблони різних дій програми, створити різні типи витрат і відобразити їх у списку на головному екрані, а також створити інфографіку на основі цих даних.

Отже, загалом додаток досить функціональний. Має зручний інтерфейс. Він легкий у використанні. Розібратися з додатком можна і без будь-якої інструкції або порад. Це все дякуючи «user-friendly» - інтерфейсу.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		61

ВИСНОВКИ

У даній дипломному проєкті створено додаток для обліку особистих витрат. Додаток працює швидко та плавно, доступний на більшості девайсів під операційною системою Android, доступ до інтернету для роботи з додатком непотрібний.

Програма надає можливість перегляду всіх витрат за місяць по конкретним категоріям, перегляду звіту про бюджет за вибраний час, забезпечує вибір категорій витрат та встановлення ліміту на них та надає можливість ставити бюджетні цілі та досягати їх, як і зазначалося у завданні.

Отже, оглянуто літературу на вибрані теми. Розглянуто структуру архітектури, інструментів та описано можливості щодо розробки додатків. Також розглянута сама розробка програми та результат її роботи

, користь. Створено програму відповідно до вимог з відповідним інтерфейсом користувача. Розроблений додаток протестований. Загалом додаток легкий та швидкий у використанні. А найголовніше, має сучасний інтерфейс та функціонал, що допоможе майбутньому користувачу.

При розробці програми були враховані всі вимоги, зазначені в цій бакалаврській роботі.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		62

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Son K. The method of android application NDK / К. Son, J. Lee. // International Conference on Awareness Science and Technology. – 2012.
2. Hermes D. Android Mobile Development: C# and Xamarin.Forms Fundamentals - Dan Hermes., 2013. – 475 с.
3. Frameboxxindrode – Режим Доступу <https://frameboxxindore.com/uk/android/what-is-the-best-database-for-android-app.html> - Дата доступу 11.05.2022 – Бази Даних
4. Oliveira L.B., Siqueira I.G., Loureiro A.A.F. On the performance of ad hoc routing protocols under a peer-to-peer application // Journal of Parallel and Distributed Computing. - 2005. - Vol.65, №11. - P.1337-1347
5. Avada Media – Режим доступу: - <https://avada-media.ua/ua/services/flutter/> - Дата доступу 15.05.2022 - Flutter
6. Карпов А. С. Изучение предпочтений пользователей мобильных приложений для планирования личных финансов / А. С. Карпов, Н. С. Павлов. – 2020. – №2020. – С. 73–78.
7. Skill Factory – Режим доступу: -<https://blog.skillfactory.ru/glossary/android-sdk/> - Дата доступу 20.05.2022 – Android SDK
8. A Developer’s First Look At Android. // LINUX FOR YOU. – 2008. – С. 48–50
9. Habr – Режим Доступу: - <https://habr.com/ru/post/16770/> - Дата доступу 10.05.2021 – Архітектура Android
10. Kumar A. Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase / Ashok Kumar., 2018. – 394 с.
11. Proglib – Режим доступу: - <https://proglib.io/p/top-10-android-freymvorkov-obzor-i-sravnenie-2020-08-05> - Дата доступу: 03.05.2021 – Порівняння фреймворків

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		63

12. Програмування – Режим доступу:

https://otherreferats.allbest.ru/programming/00693474_0.html – Дата доступу:

21.05.2022- Android

13. Android – Режим доступу: - https://www.android.com/intl/ru_ru/what-is-android/

- Дата доступу: 07.05.2022 – What is Android?

14. Nikaein N., Bonnet C. Topology management for improving routing and network performances in mobile ad hoc networks // Mobile Networks and Applications (MONET). - 2004. - Vol.9, №6. - P.583-594 (2004).

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ документа	Підпис	Дата		64

ДОДАТОК 1

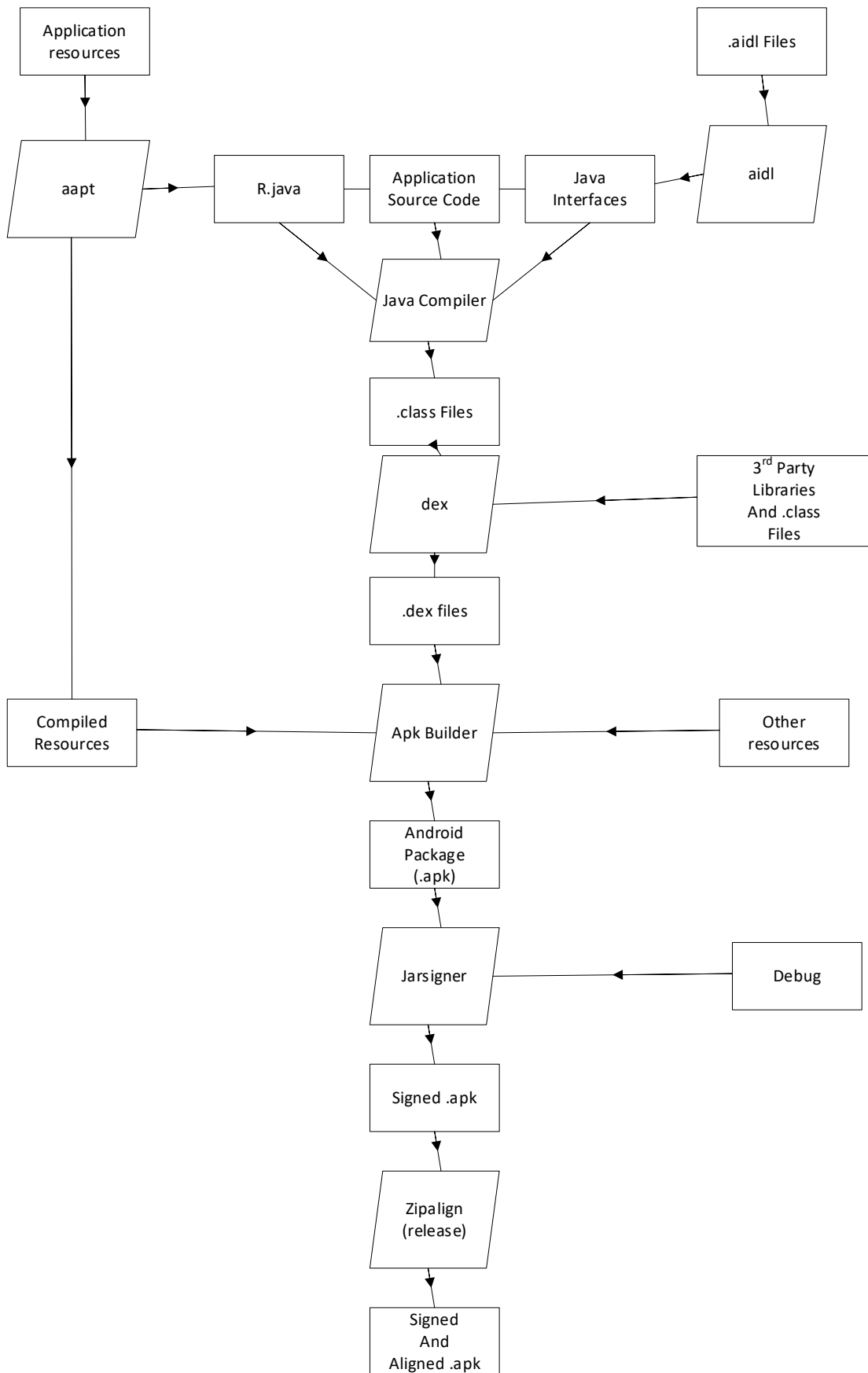
Мобільний додаток для обліку особистих витрат

Схема взаємодії програми

ІАЛЦ.467200.006 Д1

Аркушів 1

Київ – 2022 р.



					ІАЛЦ.467200.006 Д1		
Зм.	Арк.	№ документа	Підпис	Дата	Мобільний додаток для обліку особистих витрат		
Розробив		Сірокомський М.					
Перевірив		Павлов В.Г.			Літ.	Аркуш	Аркушів
						1	1
Н. контр.		Сімоненко А. В.			НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ Група ІВ-82		
Затверд.							
					Схема взаємодії програми		

ДОДАТОК 2

Мобільний додаток для обліку особистих витрат

Діаграма класів додатку

ІАЛЦ.467200.005 Д2

Аркушів 1

Київ – 2022 р.

ДОДАТОК 3

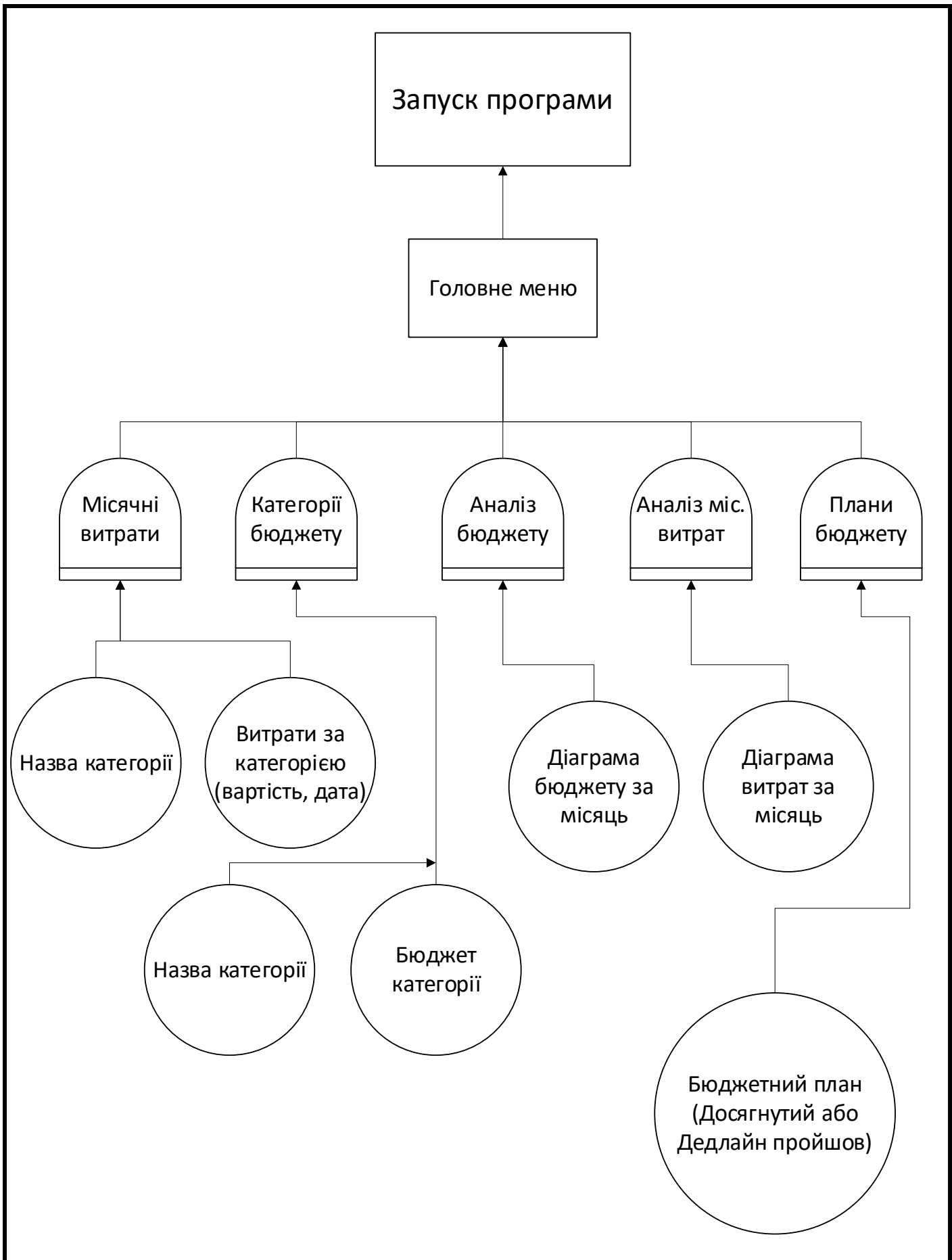
Мобільний додаток для обліку особистих витрат

Структурна схема додатку

ІАЛЦ.467200.004 ДЗ

Аркушів 1

Київ – 2022 р.



					ІАЛЦ.467200.004 ДЗ		
Зм.	Арк.	№ документа	Підпис	Дата	Мобільний додаток для обліку особистих витрат		
Розробив		Сірокомський М.					
Перевірив		Павлов В.Г.			Літ.	Аркуш	Аркушів
						1	1
Н. контр.		Сімоненко А. В.			НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ Група ІВ-82		
Затверд.							
Структурна схема додатку							

