

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка мобільного застосунку для керування
фінансів Sky Wallet мовою React Native та C#»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

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

_____ АНТОН СОКОЛ
(підпис)

Виконав: здобувач вищої освіти групи ПД-41

_____ АНТОН СОКОЛ

Керівник: _____ Світлана ШЕВЧЕНКО

к.п.н., доцент

Рецензент: _____

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« ____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Соколу Антону Юрійовичу

1. Тема кваліфікаційної роботи: «Розробка мобільного застосунку для керування фінансів Sky Wallet мовою React Native та C#»

керівник кваліфікаційної роботи к.п.н., доцент Світлана ШЕВЧЕНКО,
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. № 36.

2. Строк подання кваліфікаційної роботи «28» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи:

3.1 Інструменти розробки програмного забезпечення

3.2 Інструменти обробки та візуалізації даних

3.3 Науково-технічна література, пов'язана з розробкою

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Аналіз предметної області

4.2 Засоби програмної реалізації

4.3 Проектування та розробка програмного забезпечення

4.4 Тестування програмного забезпечення

4.5 Висновки

5. Перелік графічного матеріалу: *презентація*

5.1 Порівняння аналогів

5.2 Програмні засоби реалізації

5.3 Діаграми

5.4 Інтерфейс користувача

5.5 Апробація результатів дослідження

6. Дата видачі завдання «28» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	28.02-06.03.2024	
2	Аналіз та дослідження існуючих аналогів	07.03-13.03.2024	
3	Огляд існуючих алгоритмів керування фінансами	14.03-21.03.2024	
4	Проектування застосунку для трекінгу фінансів	22.03-3.04.2024	
5	Програмна реалізація застосунку	4.04-22.04.2024	
6	Тестування застосунку	23.04-28.04.2024	
7	Оформлення роботи: вступ, висновки, реферат	29.04-05.05.2024	
8	Розробка демонстраційних матеріалів	06.05-12.05.2024	
9	Попередній захист роботи	13.05-31.05.2024	

Здобувач вищої освіти

_____ (підпис)

Антон СОКОЛ

Керівник кваліфікаційної роботи

_____ (підпис)

Світлана ШЕВЧЕНКО

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 43 стор., 1 табл., 29 рис., 17 джерел.

Мета роботи – спрощення процесу керування особистими фінансами.

Об'єкт дослідження – керування особистими фінансами та контроль над ними.

Предмет дослідження – мобільний додаток для відслідковування та керування особистими фінансами.

Короткий зміст роботи: У роботі проаналізовано алгоритми та методи для керування особистими фінансами в контексті задачі формування фінансової грамотності населення. Проаналізовано інструментальні аналоги для трекінгу фінансів: Moneyfy, CoinKeeper. Розроблено алгоритм роботи застосунку та програмно реалізовані ключові функціональні можливості, зокрема: додавання, видалення, редагування - рахунків, категорій витрат та доходів, їх кастомізація. Також передбачено перегляд статистики як за певний період часу, так і за певною категорією. Проведено функціональне та модульне тестування додатку. В роботі використано бібліотеку Realm для локального збереження даних, React Native для створення користувацького інтерфейсу.

Сферою використання застосунку є організація самостійного контролю населення в процесі керування особистими фінансами.

ЗМІСТ

ВСТУП	7
1 ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	9
1.1 Фінансовий менеджмент - наука про керування грошима	9
1.2 Основні методи відслідковування фінансового положення	11
1.3 Огляд аналогів	16
1.3.1 Застосунки аналоги	16
1.3.2 Переваги та недоліки	21
2 АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РОЗРОБКИ	24
2.1 Засоби розробки	24
2.1.1 Мова програмування JavaScript для розробки додатків	24
2.1.2 Середовище розробки Microsoft Visual Studio Code	26
2.1.3 Огляд бібліотек та фреймворків JavaScript для створення мобільних додатків	28
3 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	30
3.1 Планування розробки ПЗ	30
3.2 Діаграма використання додатку	31
3.3 Діаграма класів	33
3.4 Схема роботи застосунку	34
3.5 Проектування інтерфейсу користувача	36

3.6 Реалізація функціональної частини	43
3.6.1 Створення навігації додатку	43
3.6.2 Створення загальних показників балансу	44
3.6.3 Створення загальних показників балансу	44
3.6.4 Створення кнопок "Плюс", "Мінус" та "Обмін між рахунками"	45
3.6.5 Створення кнопки профіль	47
3.7 Тестування додатку	49
ВИСНОВКИ	50
ПЕРЕЛІК ПОСИЛАНЬ	51
ДОДАТОК А. ДЕМООНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)	53

ВСТУП

Актуальність теми. Сьогодні все більшої популярності набирають додатки для фінансового менеджменту, адже більшість людей намагаються контролювати свої фінанси. У такій важливій справі важко тримати все у голові, тому тут на допомогу приходять сервіси та застосунки для трекінгу фінансів.

У сучасному світі багато людей мають проблеми з керуванням своїми грошима. Найчастішими наслідками цього є те, що вони не мають змогу в повному обсязі використовувати свої можливості, адже не користуються всіма перевагами фінансової грамотності.

Онлайн банкінги дуже допомогли суспільству відслідковувати свої витрати та заощадження на рахунках у онлайн режимі, але для більшості дуже незручно кожного разу заходити у безліч додатків та перегляди баланс, та й готівкові витрати для заощадження відслідковувати не вийде. Якщо у людей буде застосунок, який вони можуть використовувати будь-коли, в якому вони можуть додати будь-яку кількість рахунків, мати можливість додавати доходи та витрати, створювати постійні категорії: витрати на автомобіль, витрати на продукти або походи у кіно та інші розваги; бачити статистику всіх своїх фінансових операцій. Sky Wallet - застосунок, розроблений на базі React Native та бібліотеки Realm, що стане надійним асистентом у керуванні фінансами.

На відміну від інших додатків на ринку, Sky Wallet виділяється своєю унікальною функціональністю, яка враховує досвід користувачів попередніх застосунків. При цьому він також пропонує інноваційні можливості, що роблять керування фінансами ще більш зручним та ефективним.

Таким чином, користувачі стають більш свідомими у використанні особистих коштів за допомогою Sky Wallet.

Об'єкт дослідження - керування особистими фінансами та контроль над ними.

Предмет дослідження - мобільний додаток для відслідковування та керування особистими фінансами.

Мета роботи - спрощення процесу керування особистими фінансами.

Для досягнення поставленої мети необхідно виконати наступні **завдання**:

1. Проаналізувати процес, пов'язаний з керуванням та контролем особистих фінансів.
2. Проаналізувати переваги та недоліки існуючих додатків – аналогів з керування та контролю особистих фінансів.
3. Проаналізувати технічні засоби та технології для реалізації функціоналу додатку для трекінгу фінансів.
4. Спроекувати дизайн та розробити дизайн додатку на основі потреб користувачів.
5. Провести тестування додатку.

Методи дослідження. Для вирішення вищезгаданих завдань у дослідженні використано наступні методи: системно-структурні методи, порівняльний аналіз. Для розробки програмного забезпечення використано бібліотеку Realm для локального збереження даних, React Native для створення користувацького інтерфейсу.

Наукова новизна одержаних результатів. Наукова новизна полягає у тому, що у розробленому додатку можливо додати будь-яку кількість рахунків, мати можливість додавати доходи та витрати, створювати постійні категорії, що дозволяє спростити процес контролю особистими рахунками.

Практична значущість результатів полягає у створенні програмного продукту у вигляді мобільного додатку для трекінгу фінансів Sky Wallet на основі React Native та C#, який дозволить забезпечити спрощений процес керування усіма особистими рахунками. Додаток призначений для користувачів смартфонів, що мають на меті підвищити свій рівень фінансової грамотності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Фінансовий менеджмент - наука про керування грошима

Фінансовий менеджмент – це процес управління фінансовими ресурсами, який включає планування, контроль та моніторинг, і є ключовим елементом управління як компаній, так і особистих фінансів. Здатність грамотно розпоряджатися прибутком і контролювати витрати для досягнення поставлених фінансових цілей – одна із задач фінансового менеджменту.

Зміст поняття «фінансовий менеджмент» розглядається з точки зору його складових частин – «фінанси» та «менеджмент» [1].

Фінанси – це відносини, що виникають у процесі створення і перерозподілу суспільних благ і багатств. Це сукупність грошових відносин, які формують і використовують фонди грошових ресурсів. Зокрема, це: 1) сукупність грошових коштів, що є в розпорядженні підприємств, держави тощо; 2) кошти, гроші, фінансові справи (згідно з тлумачним словником); сукупність грошових відносин, через які здійснюється формування, розподіл та використання фондів грошових ресурсів з метою відшкодування витрат, пов'язаних з процесом виробництва та виконанням державних функцій.

Менеджмент – це сукупність принципів, методів, засобів і форм управління виробництвом, спрямованих на підвищення його ефективності (згідно з тлумачним словником). Деякі вчені визначають менеджмент як систему економічного управління виробництвом, яка включає принципи, методи, форми та прийоми управління. У загальному розумінні, термін «менеджмент» означає управління, керівництво, завідування [2].

Фінансовий менеджмент може бути розділений на кілька видів, таких як державний сектор, бізнес і особисті фінанси, кожен з яких має свої особливості і

може бути ефективним у конкретній галузі. Ключовими аспектами фінансового менеджменту є планування, аналіз та контроль.

- Фінансове планування включає розробку планів з використанням фінансових ресурсів організації з урахуванням її цілей та потреб.
- Фінансовий аналіз оцінює фінансову стійкість організації через аналіз фінансової звітності та даних, а також оцінку фінансових ризиків.
- Фінансовий контроль здійснюється в моніторингу та контролі за використанням фінансових ресурсів організації шляхом визначення стандартів та процедур.

Компанія фінансового менеджменту спрямована на управління її фінансовими ресурсами для досягнення ефективності та фінансових цілей таких, як підвищення капіталу чи розширення бізнесу. Основні аспекти цього бюджету включають бюджетування, управління кредитами та інвестиціями, фінансовий аналіз та планування. Для ефективного управління фінансами та ведення бюджету необхідна фінансова грамотність. Фінансова грамотність означає розуміння фінансових понять та принципів, здатність до аналізу фінансових даних та управління ресурсами. Фінансова грамотність є важливою для кожної людини.

За методологією ОЕСР у звіті «Фінансова грамотність, фінансова інклюзія та фінансовий добробут в Україні» за 2021 рік загальний індекс фінансової грамотності України становить 12,3 бали (або 58% від його максимального значення) [3]. Цей рейтинг визначається на основі показників фінансової поведінки населення та рівня фінансових знань. На рис 1.1 представлена порівняльна таблиця темпу приросту Індексу фінансової грамотності у країнах, які брали участь в опитуваннях ОЕСР у 2016 р. і 2020 р.. Країни порівняння обрані за чотирма критеріями: 1) наявність країни у двох дослідженнях ОЕСР щодо фінансової грамотності (2016 р. та 2020 р.); 2) індекс фінансового розвитку менший за 0,5; 3) розташовані в Європі; 4) нижче середнього, вище середнього або високий рівень ВВП на душу населення за методологією Світового банку.

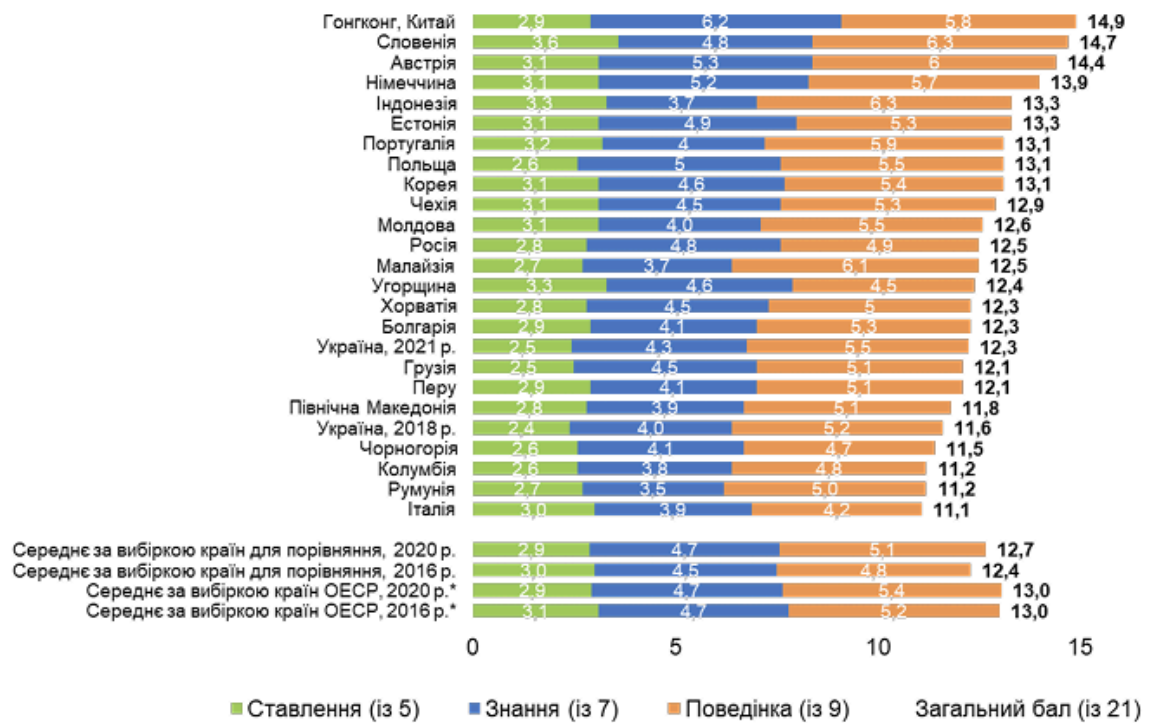


Рис.1.1 Рейтинг країн з найбільш фінансово грамотним населенням

Індекс фінансової грамотності в Україні наразі демонструє гарну динаміку – за менше ніж 3 роки відбулося зростання на 6,0% у бальному вираженні; усі компоненти індексу покращилися. Саме ця поведінка має прямий вплив на розвиток фінансового ринку країни, рівень тіньової економіки та ставлення населення до реформ і ініціатив влади.

1.2 Основні методи відслідковування фінансового положення

Основними засобами для організації контролю персональної економічної активності можуть бути описані наступним переліком [3]:

- планування подумки;
- паперові засоби;
- таблиці Excel, Google Sheets, тощо;
- онлайн ресурси та мобільні додатки.

Кожен з цих методів має свої переваги та недоліки, роблячи його у деяких умовах ідеальним.

- Планування подумки - є одним з найстаріших та найменш ефективним методом зі списку. Основна властивість нашого мозку - забувати, а з огляду на те, що кожен день людина має справу з великою кількістю справ та інформації, можна легко втратити важливу інформацію.

- Паперовий метод ведення обліку залишається популярним способом відстеження доходів та витрат, незважаючи на зростання популярності цифрових інструментів. Цей метод простий, доступний та не потребує спеціальних навичок.

Одним з найпоширеніших методів ведення обліку на папері є система Какебо, яка має високу варіативність спеціальних щоденників для різних сфер та задач. Основним правилом є ведення двох щоденників, один з яких призначено для планування витрат, запису доходів та збережень, інший для запису витрат у режимі реального часу. Перший блокнот може бути розлінованим будь яким чином за бажанням користувача, але повинен обов'язково містити наступні елементи:

- план доходів на місяць – оформлюється у вигляді таблиці чи списку і включає дані про всі надходження коштів, на початок місяця вносяться дані про доходи, які гарантовано надійдуть, всі інші доходи вносяться іншим кольором або іншими літерами, наприклад, великими;

- план заощаджень на місяць – включає суми, які користувач хотів би відкласти в скарбничку (готівкову чи у вигляді банківського рахунку);

- план витрат на місяць – містить інформацію про всі постійні витрати, наприклад, витрати на оренду квартири, комунальні платежі, витрати на інтернет та мобільний зв'язок, витрати на проїзд в громадському транспорті тощо.

Гроші, які залишаються після віднімання обов'язкових витрат та заощаджень, згідно з методикою Какебо поділяють у визначений користувачем пропорції на наступні категорії:

- витрати на життя – продукти харчування, побутова хімія, одяг, взуття, медичні витрати (заплановані візити до лікарів та медикаменти, що вживаються постійно, наприклад, вітаміни) тощо;

- культура та освіта (в тому числі, спорт) – навчання, екскурсії, тренінги та тренування тощо;
- розваги – зустрічі з друзями, відвідування культурних заходів, короткотривалі подорожі тощо;
- інші витрати.

Наприкінці кожного місяця виконується аналіз даних з блокноту з метою визначення відхилень від плану та виявлення пунктів, на яких вдалося зекономити або витрати перевищили заплановані. Результати аналізу застосовуються для подальшого планування на наступний місяць.

Переваги та недоліки ведення паперового щоденника для контролю особистих фінансів:

Переваги:

- Простота та доступність: паперовий щоденник не потребує спеціальних навичок чи знань, що робить його зручним для користувачів будь-якого віку, зокрема старшого покоління.
- Гнучкість: можете розлінувати блокнот відповідно до своїх потреб та уподобань, а також змінювати макет щомісяця.
- Автономність: ведення паперового щоденника не залежить від сторонніх спеціалістів чи програмного забезпечення.

Недоліки:

- Вартість: друковані блокноти можуть бути дорогими, а створення власного потребує часу та ресурсів.
- Необхідність різнокольорових інструментів: для наочного ведення записів потрібні різнокольорові ручки або олівці.
- Ризик помилок: при внесенні даних вручну можливі помилки в розрахунках.
- Незручність виправлень: зміна даних потребує закреслення або витирання, що може зробити блокнот негарним.
- Обмежений простір: площа блокнота обмежує обсяг даних, які можна зафіксувати.
- Відсутність захисту: дані в блокноті не захищені від стороннього доступу.

- Складність аналізу: формування статистичних даних з паперових записів може бути складним та трудомістким.

Висновок: паперовий щоденник може бути зручним та доступним інструментом для контролю особистих фінансів, особливо для тих, хто не любить використовувати цифрові технології. Проте, він має й певні недоліки, такі як ризик помилок, складність виправлень, обмежений простір та складність аналізу даних.

Таблиці Excel є більш сучасним методом, який задовольняє потреби організації контролю фінансової активності. Приклад таблиці сімейного бюджету наведений на рисунку 1.2.

Наименование		Июл	Авг	Сен	Окт	Ноя	Дек	2007г
Доходы		48 080	65 080	45 080	48 080	45 000	45 000	629 496
Доходы родителей		29 080	29 080	29 080	29 080	29 000	29 000	310 496
Доходы мамы		25 080	25 080	25 080	25 080	25 000	25 000	262 496
1 З/п		15 000	15 000	15 000	15 000	15 000	15 000	180 000
2 Доходы от бизнеса		10 000	10 000	10 000	10 000	10 000	10 000	82 000
3 Прогонты по вкладам		80	80	80	80			496
Доходы папы		4 000	4 000	4 000	4 000	4 000	4 000	48 000
2 Доходы бабушки		7 000	7 000	7 000	7 000	7 000	7 000	84 000
3 Доходы старшего сына		8 000	8 000	8 000	8 000	8 000	8 000	96 000
Кредиты			20 000					115 000
5 Ден. доходы		4 000	1 000	1 000	4 000	1 000	1 000	24 000
Расходы		85 870	104 170	73 970	81 470	78 970	73 770	1 110 180
1 Общие расходы		34 570	39 570	39 670	35 170	34 670	39 470	478 210
2 Индивидуальные расходы		26 300	29 600	24 300	26 300	24 300	24 300	315 100
3 Вклады		10 000	25 000		10 000	10 000		149 000
4 Возврат займа		15 000	10 000	10 000	10 000	10 000	10 000	126 000
САЛЬДО		-37 790	-39 090	-28 890	-33 390	-33 970	-28 770	-480 684
Остаток		-316 574	-355 664	-384 554	-417 944	-451 914	-480 684	

Рис.1.2 Приклад таблиці Excel для ведення сімейного бюджету

Переваги та недоліки ведення обліку в Excel:

Переваги:

- Простота інтерфейсу: Excel має інтуїтивно зрозумілий інтерфейс, що робить його доступним для користувачів будь-якого рівня підготовки.
- Гнучкість: можна легко створювати власні шаблони таблиць для ведення обліку, які можна використовувати повторно.
- Автоматизація розрахунків: Excel дозволяє використовувати формули для

автоматизації розрахунків, що економить час та зменшує ризик помилок.

- Введення даних: можна використовувати графічні форми для спрощення введення даних.

- Списки та підстановки: Excel дозволяє створювати списки підстановки та автоматичні підстановки, що економить час та зменшує ризик помилок.

- Аналітика: Excel пропонує широкий спектр інструментів для аналізу даних, включаючи пошук, фільтрацію, сортування, статистичні формули, зведені таблиці та діаграми.

Недоліки:

- Складність створення шаблонів: створення якісних шаблонів таблиць може потребувати певних знань та досвіду.
- Розростання таблиць: з часом таблиці можуть ставати дуже великими та складними, що ускладнює роботу з ними.
- Копіювання формул: для формування даних за новий період необхідно копіювати формули та шаблони, що може бути трудомістким.
- Макроси: для автоматизації формування звітів може знадобитися знання мови програмування VBA та написання макросів.
- Ручне введення: введення даних вручну може призвести до пропуску позицій.
- Відсутність структури: Excel не має вбудованих засобів для структурування списків за категоріями.
- Захист даних: засоби захисту даних в Excel обмежені, що робить дані уразливими до стороннього доступу.

Висновок: Excel може бути ефективним інструментом для ведення обліку особистих фінансів, але його безкоштовні шаблони не достатньо прості і громіздкі у застосуванні. Преміум-шаблони бюджету *Excel для Microsoft 365* більш гнучкі та привабливі для використання, проте є платними.

1.3 Огляд аналогів

Сьогодні найбільш досконалими та зручними методами для коригування своїми фінансами є мобільні застосунки.

1.3.1 Застосунки аналоги

1. Monefy

Застосунок українського розробника, для керування власними фінансами. Додаток доволі простий у використанні, має інтуїтивно зрозумілий інтерфейс, що сприяє швидкій адаптації при переході з іншого застосунку [4].

Застосунок має платну версію, придбавши яку, користувачі відкривають доступ до певного функціоналу, недоступному у стандартній версії.

На головному екрані можна одразу помітити діаграму всіх витрат за день. Додаток показує графіки та статистику витрат, де одразу видно, яка категорія скільки відсотків займає від загального кола. Це дозволяє користувачам легко відстежувати свої витрати та бачити, де вони витрачають більше грошей. Баланс та великі зрозумілі кнопки “+” та “-”. На рис. 1.3 представлено головний екран застосунку



Рис. 1.3 Головний екран Monefy

В додатку реалізовано велику кількість категорій витрат (рис. 1.4), із можливістю додавання нових та редагування існуючих. Наприклад, витрати можна розділити на їжу, транспорт, одяг, платежі, дім, подарунки і т.д.

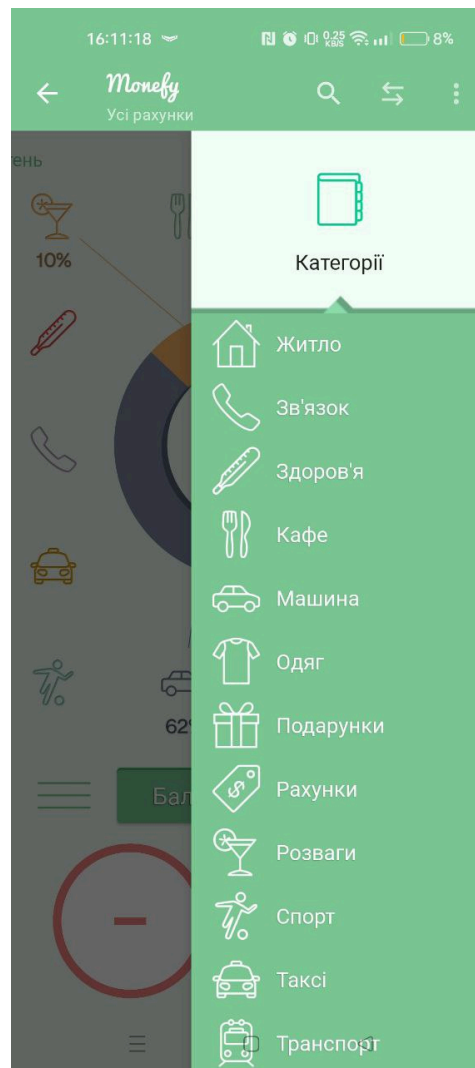


Рис. 1.4 Перелік основних категорій витрат та доходів у застосунку Monefy

У програмі реалізовано достатньо простий та зручний спосіб запису інформації. На рисунку 1.5 показано, що навіть користувач, який ніколи раніше не мав справу з такими застосунками, відразу розуміє, що до чого.

Присутня функція створення бюджету рис.1.6. Він створюється один на всі рахунки та категорії. Бюджет відображається на спеціальній вкладці, яка об'єднує всі транзакції по рахункам.

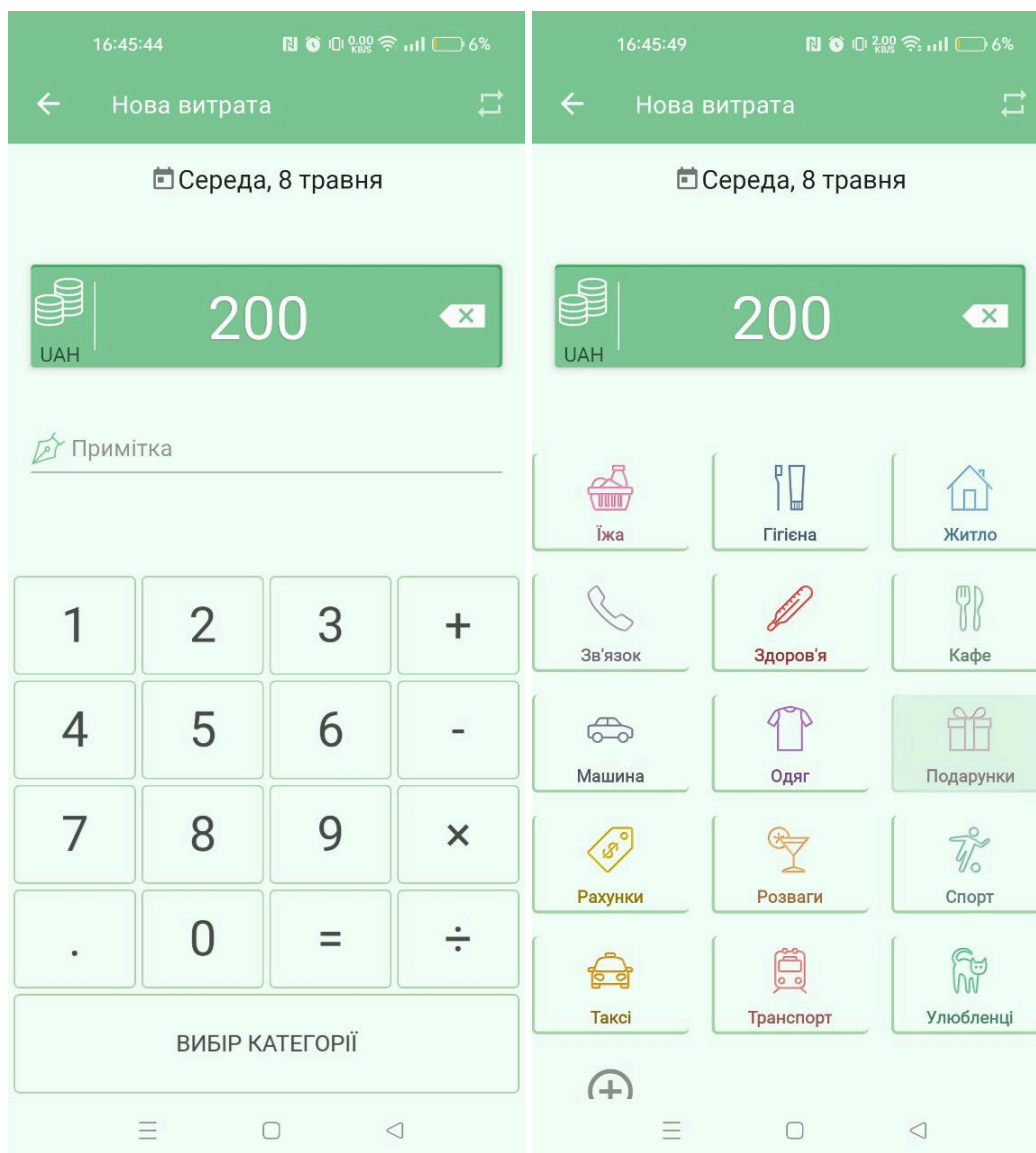


Рис.1.5 Реалізація запису витрат Moneyfy

Суть полягає в тому, що для бюджету встановлюється сума, яка має бути витрачена за місяць, тому у вкладці «Усі рахунки» при перевищенні зазначеного бюджету баланс буде від’ємним навіть, якщо у решти баланс додатній.

Якщо розглядати дану вкладку з відключеною функцією бюджету, то баланс буде рахуватись як сума балансів усіх рахунків.

Додаток доступний на платформах Android та iOS, і доступний для завантаження з Play Market та App Store.



Рис.1.6 Результати тестування функції “Бюджет”

2. CoinKeeper [5].

Ще один застосунок для трекінгу фінансів, розроблений компанією “Disrapp”, офіс якої до 2011 року знаходився у Львові. На відміну від попереднього додатку, цей має менш інтуїтивний інтерфейс та нав’язливу рекламу, перше, що бачить користувач - пропозиція купівлю платної версії.

Насамперед, зрозумілий інтерфейс притягує користувачів через свою легкість та швидкість, натомість CoinKeeper вирішив зробити ставку на дизайн, а не на зручність.

Ось так на рисунку 1.7 реалізовано процес додавання витрат, насамперед незрозумілий, виглядає гарно, але UI концепт розроблений не вірно.

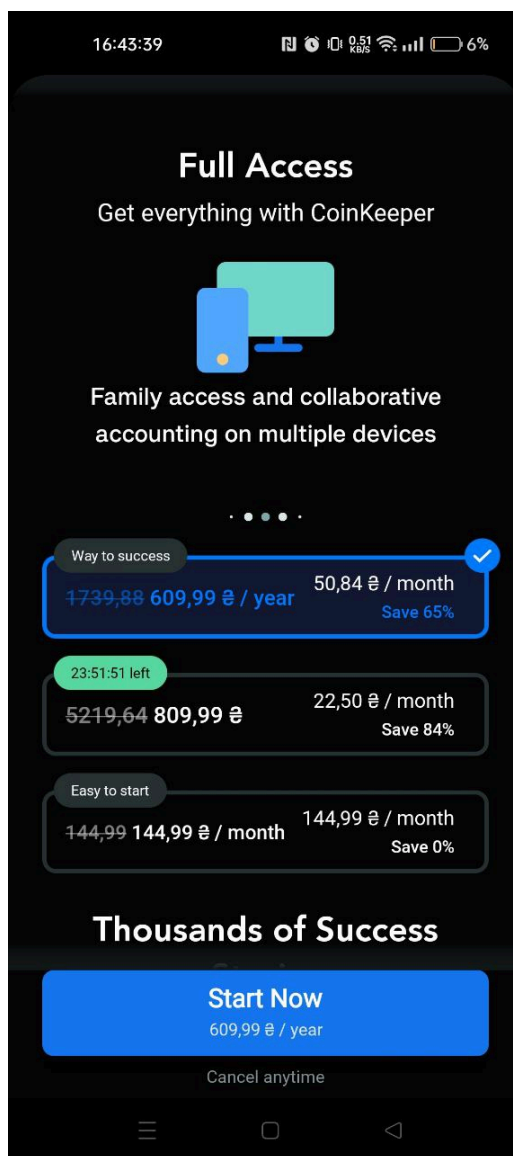


Рис.1.7 Стартова сторінка з пропозицією купівлі повної версії CoinKeeper

Незважаючи на це, загальна статистика доходів та витрат є зрозумілою. Поточний баланс записаний під кожним рахунком, сума витрат під кожною категорією є досить цікавою ідеєю, адже користувачам не потрібно вираховувати відсоткове значення або заходити на сторінку повної статистики, як, наприклад, у минулому застосунку, тут все видно одразу (рис.1.8 – 1.9)

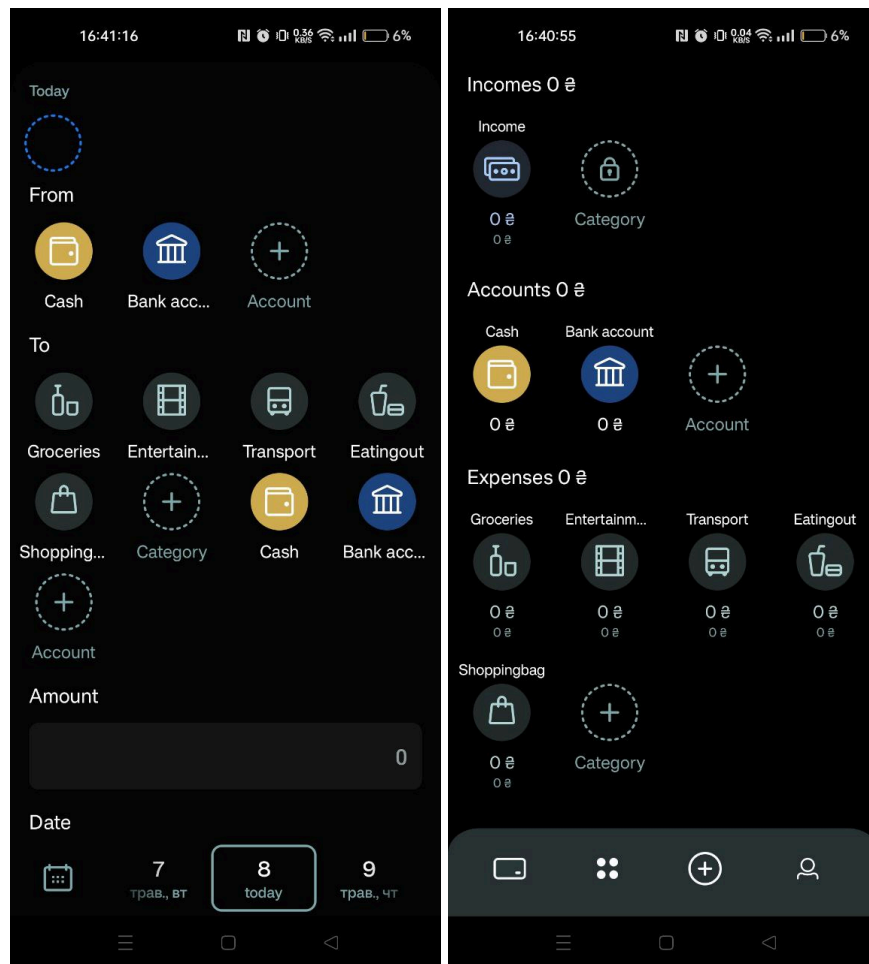


Рис. 1.8 - 1.9 Реалізація запису витрат застосунку CoinKeeper та статистика

1.3.2 Переваги та недоліки

Проведений аналіз даних застосунків, визначення їх переваг та недоліків дозволив здійснити порівняльну характеристику їх основних функціональних та нефункціональних вимог (таблиця 1.1).

Таблиця 1.1

Порівняння існуючих програм аналогів

Назва	Переваги	Недоліки
Monefy	<ul style="list-style-type: none"> ● Інтуїтивний інтерфейс ● Резервна копія на хмарному сервісі ● Відсутність реклами 	<ul style="list-style-type: none"> ● Застарілий інтерфейс ● Деякі спливаючі повідомлення закривають кнопки ● Більшість корисного функціоналу у про-версії
CoinKeeper	<ul style="list-style-type: none"> ● Гарний дизайн 	<ul style="list-style-type: none"> ● Набридлива реклама купівлі про-версії ● Не зрозумілий інтерфейс ● Майже всі корисні функції - платні
SkyWallet	<ul style="list-style-type: none"> ● Інтуїтивний інтерфейс ● Відсутність реклами ● Відсутність платної версії ● Локальне збереження даних без доступу в Інтернет 	<ul style="list-style-type: none"> ● Відсутність української локалізації ● Відсутність паролю на вхід

Аналіз доступних додатків - аналогів дозволяє знайти всі недоліки та попередити їх з метою не допустити таких самих помилок у своєму додатку та дозволяє сформулювати вимоги до програмного продукту.

SkyWallet має бути реалізованим для мобільної платформи та мати наступний функціонал:

- здатність коригувати всі дані в режимі реального часу;
- здатність створювати, видаляти та редагувати нові категорії, рахунки;
- здатність переглядати статистику по кожній з категорій витрат.

2 АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РОЗРОБКИ

2.1 Засоби розробки

2.1.1 Мова програмування JavaScript для розробки додатків

Основною мовою програмування для створення додатку було обрано JavaScript [6]. JavaScript — це універсальна мова програмування, яка широко використовується для веб-розробки, і вона також залишила свій слід у розробці мобільних додатків (рис. 2.1). Завдяки фреймворкам, таким як React Native і Ionic, JavaScript дозволяє розробникам ефективно створювати кросплатформні мобільні додатки. Популярність JavaScript і велика спільнота розробників дозволяють легко знайти ресурси та підтримку. Її гнучкість і сумісність з різними платформами роблять JavaScript цінним вибором для розробки мобільних додатків.

JavaScript вражає своєю універсальністю, адже з її допомогою можливо створювати не лише мобільні додатки. Її легкість дозволяє уникати більшості проблем при розробці в таких сферах як [6]:

1. Веб-розробка. Використання JavaScript для веб-розробки є універсальним засобом. Додавання динамічних елементів на веб-сторінку, таких як спливаючі вікна, анімована графіка є однією особливістю JavaScript. Автоматична підтримка кросбраузерності та кросплатформенності відрізняє його від інших мов програмування.

2. Розробка ігор. JavaScript підходить також для створення ігор, особливо HTML5-ігор. Гейм-розробники можуть використовувати цю мову для реалізації інтерактивних елементів гри, таких як рух персонажів, користувацькі інтерфейси та ігрова логіка.

3. Розробка мобільних додатків. Використання мови JavaScript для мобільних додатків користується великою популярністю, адже є кілька фреймворків, таких як React Native та Ionic, що надають розробникам можливість

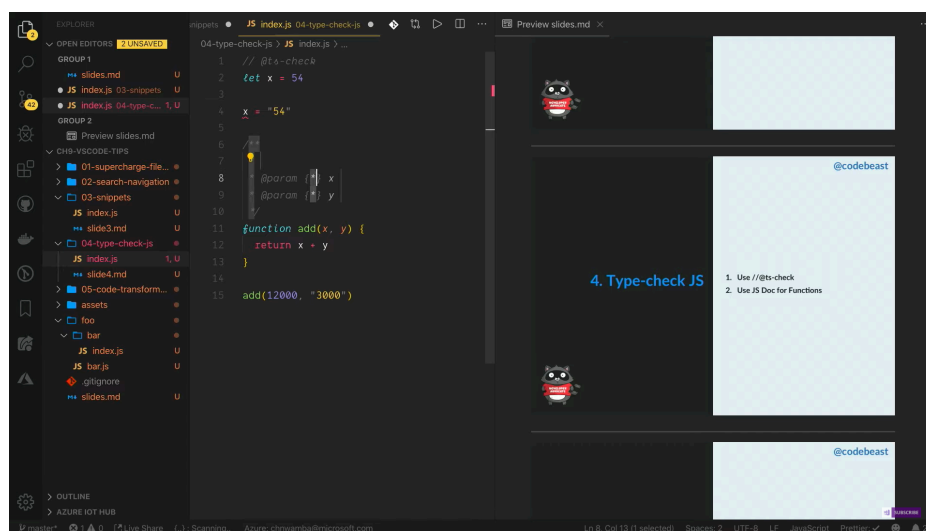
створювати якісні додатки. Ці фреймворки дозволяють створювати мобільні додатки на різні платформи, включаючи IOS та Android.

4. Розробка на стороні сервера. JavaScript використовується не тільки розробкою клієнтської частини додатку. Її використовують для налаштування на стороні сервера. Node.js - популярне середовище для використання JavaScript на стороні сервера. Надає платформу для масштабованих мережевих додатків.

5. Веб-скрепінг. JavaScript можна використовувати для веб-скрепінгу, тобто процесу вилучення даних з веб-сайтів. За допомогою JavaScript можна автоматизувати процес збору інформації з веб-сторінок. Це може бути корисно для проведення аналізу даних та різноманітних досліджень.

6. Інтерактивні карти. JavaScript можна використовувати для створення інтерактивних карт. Існує кілька бібліотек, як-от Leaflet і OpenLayers, які надають можливість для розробників створювати інтерактивні карти за допомогою JavaScript. Один з прикладів використання JavaScript для створення інтерактивних карт - карта відслідковування доставки.

7. Чат-боти. JavaScript може бути використаний для розробки чат-ботів, які здатні імітувати людське спілкування. Чат-боти можуть застосовуватися для різних завдань, таких як підтримка клієнтів чи продажі. JavaScript часто використовується для створення інтерфейсу та написання логіки чат-ботів.



```
1 // @ts-check
2 let x = 54
3
4 x = "54"
5
6
7
8 @param {number} x
9 @param {number} y
10
11 function add(x, y) {
12   return x + y
13 }
14
15 add(12000, "3000")
```

4. Type-check JS

1. Use `//@ts-check`
2. Use JS Doc for Functions

Рис. 2.1 Приклад написання коду мовою JavaScript

2.1.2 Середовище розробки Microsoft Visual Studio Code

Для забезпечення швидкої та з найменшою кількістю проблем мовою JavaScript, дуже важливо вибрати правильне середовище розробки. У даному дослідженні - Microsoft Visual Studio Code [7].

Visual Studio Code – це спрощений, але потужний редактор вихідного коду, який запускається на комп'ютері й доступний для Windows, macOS і Linux (рис. 2.2). Програма має вбудовану підтримку JavaScript, TypeScript і Node.js, а також багату екосистему розширень для інших мов (наприклад, C++, C#, Java, Python, PHP і Go) і середовищ виконання (наприклад, .NET і Unity).

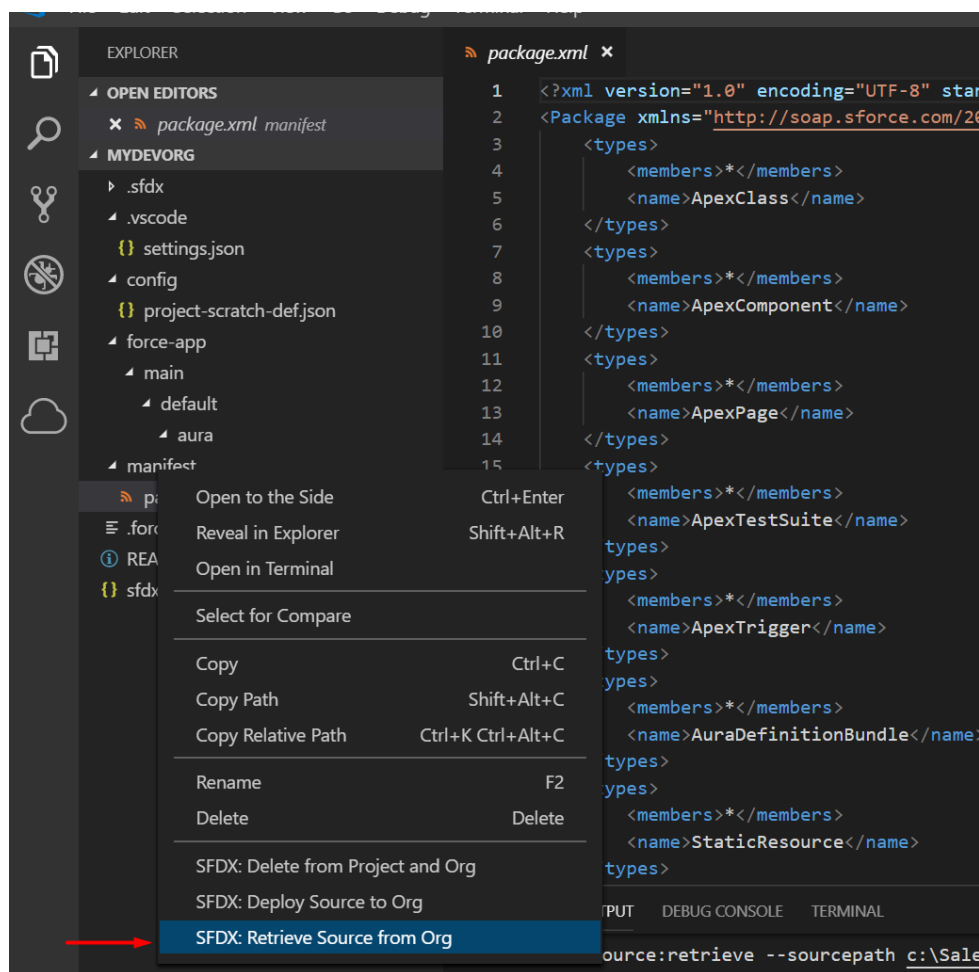


Рис. 2.2 Середовище розробки Visual Studio Code

Також для запуску додатку на емуляторах, тестування на різних пристроях та версіях Android використовувалась Android Studio.

Android Studio [8] — це офіційне інтегроване середовище розробки (IDE) для розробки програм для Android. Базуючись на потужному редакторі коду та інструментах розробника від IntelliJ IDEA, Android Studio пропонує ще більше функцій (рис. 2.3), які підвищують продуктивність під час створення програм для Android, наприклад:

- гнучка система збирання на основі Gradle;
- швидкий і багатфункціональний емулятор;
- уніфіковане середовище, де можна розробляти для всіх пристроїв Android; редагування в режимі реального часу для оновлення компонентів, що складаються, в емуляторах і на фізичних пристроях;
- шаблони коду та інтеграція GitHub, які допоможуть вам створювати загальні функції програми та імпортувати зразок коду;
- інструменти Lint для визначення продуктивності, зручності використання, сумісності версій та інших проблем.

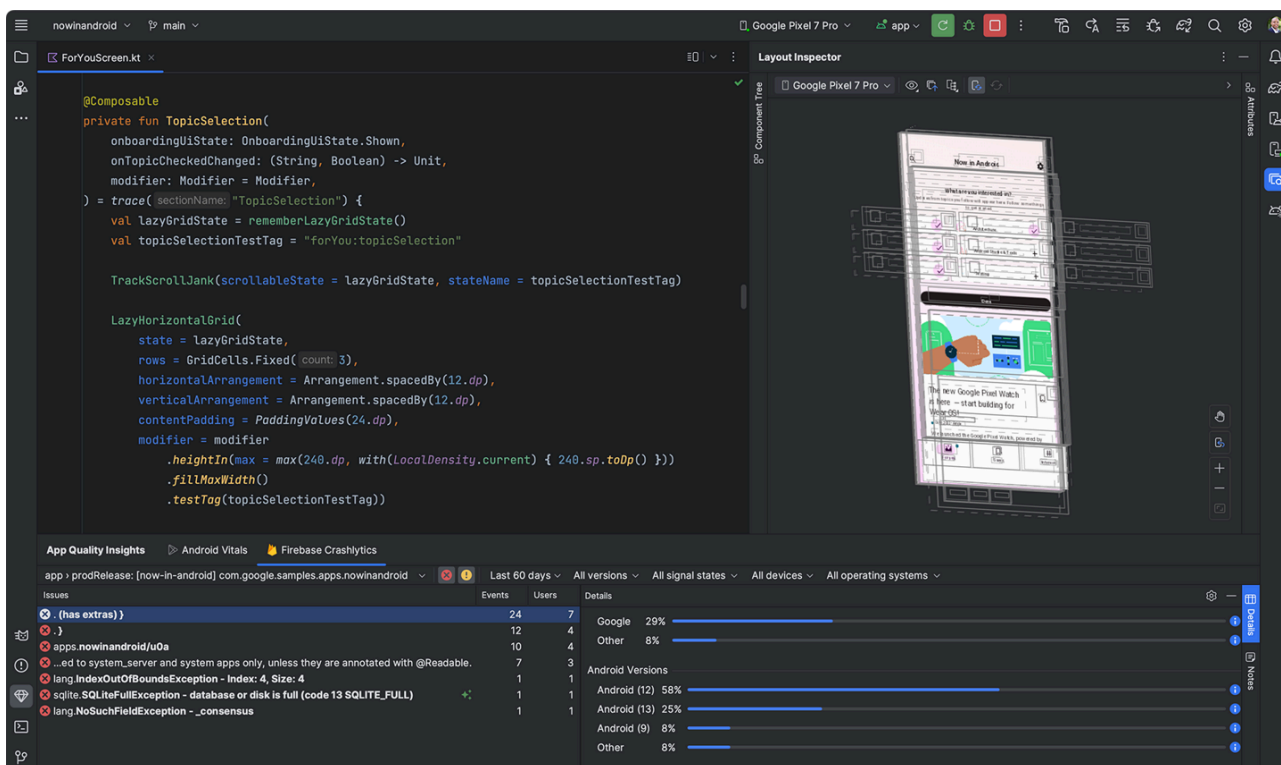


Рис. 2.3 Середовище розробки Android Studio

2.1.3 Огляд бібліотек та фреймворків JavaScript для створення мобільних додатків

Для підвищення ефективності розробки та досягнення максимальної ефективності під час розробки мобільного застосунку мовою JavaScript корисно використовувати додаткові бібліотеки та інструменти, що збільшують потенціал та можливості розробки.

Однією з найбільш корисних бібліотек для розробки є React [9]. React надає можливість створювати інтерактивні інтерфейси користувача. Вам достатньо описати, як частини інтерфейсу програми виглядають у різних станах. React буде своєчасно оновлювати їх, коли дані змінюються.

Використовуючи React є можливість створювати інкапсульовані компоненти з власним станом, а потім об'єднувати їх у складні інтерфейси користувача.

Оскільки логіка компонента написана на JavaScript, а не міститься в шаблонах, можна з легкістю передавати різні дані по всьому додатку і тримати стан поза DOM.

Для реалізації проєкту також використовувався фреймворк React - React Native. React Native [10] - це фреймворк, який базується на React, і призначений для розробки мобільних додатків (рис. 2.4).

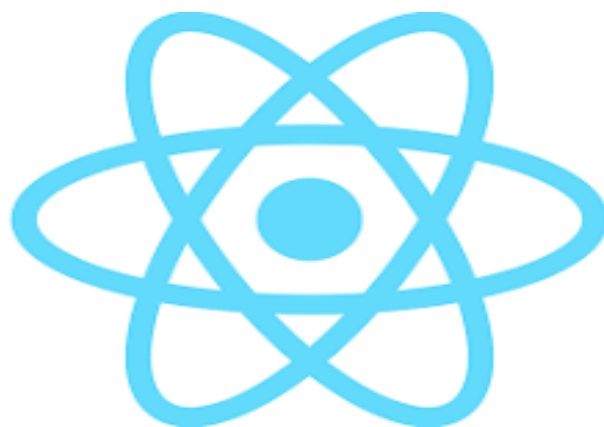


Рис. 2.4 Логотип бібліотеки React та React Native

Він використовує ті ж самі принципи та синтаксис, що й React, але дозволяє розробляти мобільні додатки з використанням JavaScript та React, замість традиційних нативних мов програмування, таких як Java або Kotlin для Android і Swift або Objective-C для iOS.

Отже, React Native можна розглядати як розширення React, яке дозволяє використовувати React-підходи до розробки інтерфейсів користувача на мобільних платформах. Він надає можливість неодноразового використання компонентів React та знань React для швидкого створення мобільних додатків для обох платформ.

3. ПРОЕКТУВАННЯ І РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Планування розробки ПЗ

Розробку будь-якого програмного продукту необхідно відстежувати, записуючи список задач, які необхідно виконати. Для кращої колективної роботи, кожен учасник команди має мати доступ до *dashboard*, де є відображення завдань: активні, завершені, заплановані тощо. Популярною платформою для цього є KanbanFlow.

KanbanFlow [11] - онлайн-інструмент для керування проєктами. Він надає зручну дошку з записами усіх запланованих або виконаних задач, що дозволяє учасникам розробки керувати задачами у режимі реального часу.

Перевагами цього інструменту є:

- спільний доступ та робота над задачами;
- гнучкість налаштувань проєкту;
- прозорий та зрозумілий інтерфейс.

Здатність створювати типізовані задачі або використовувати типи за замовчуванням:

- заплановані задачі, to-do;
- задачі, які можуть бути в майбутньому, future;
- задачі в процесі, in progress;
- завершені задачі, done.

Як можна помітити, основних типів задач достатньо для комфортної розробки та планування додатку (рис. 3.1).

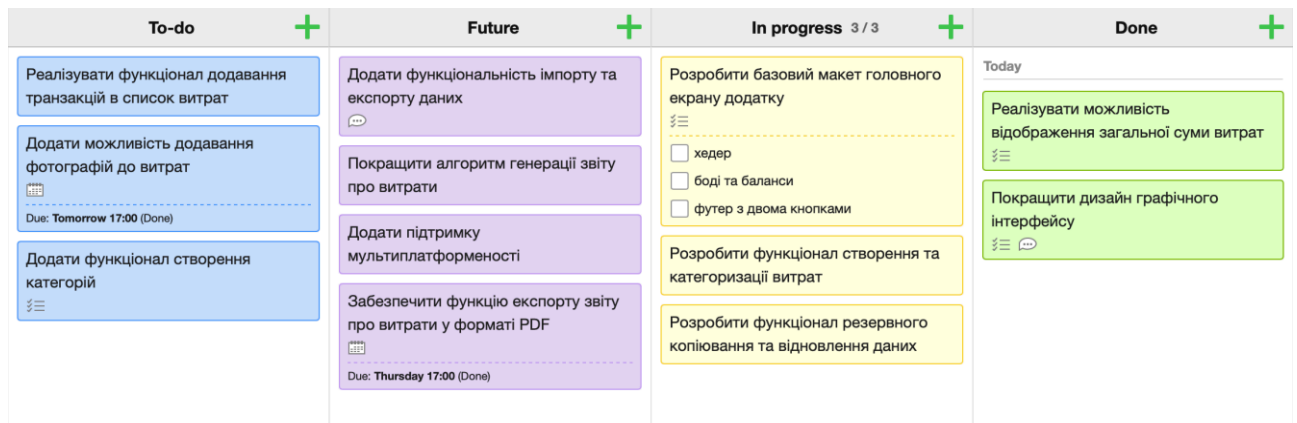


Рис. 3.1 Дашборд KanbanFlow для керування проектом

Функція створення нового тикету має багато корисних параметрів, що дозволяє повністю кастомізувати задачу: обрати колір, призначити виконавця (рис. 3.2).

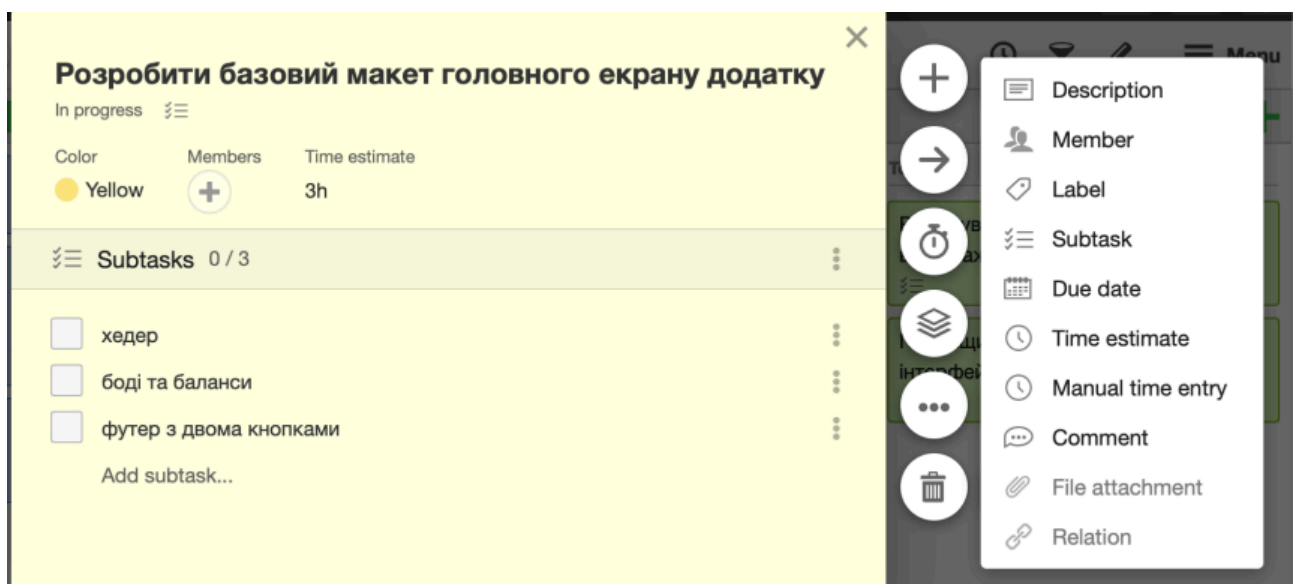


Рис. 3.2 Створення нової задачі зі статусом In Progress

3.2 Діаграма використання додатку

Діаграма використання [12] - діаграма, що відображає взаємодію користувача та додатка. Діаграма використання вказує, як користувачі діють з різними функціями та можливостями додатку.

Актор - сутність, що взаємодіє з програмою. Іноді інша система може бути представлена у вигляді актора, система, що знаходиться поза контекстом програми. Беручи до уваги специфіку програми, для діаграми використання було обрано одного актора, адже багатокористувацьке середовище не передбачено. Важливою особливістю додатку є те, що локальне сховище даних автоматично зберігає інформацію про рахунки та транзакції користувача, забезпечуючи безпеку зберігання даних.

Під час створення діаграми використання важливо враховувати не лише акторів, але й сценарії використання системи. Таким чином, можна детально описати функціонал додатку:

Головний екран додатку містить переходи до меню перегляду статистики, меню перегляду списку рахунків та категорій, меню внесення прибутку та витрат.

Меню перегляду статистики дозволяє користувачу вибрати дату перегляду статистики, а також параметри, такі як: прибуток або витрати, категорію, рахунок

Меню перегляду рахунків надає можливість створювати, редагувати та видаляти рахунки.

Меню перегляду категорій має доступні функції створення, редагування та видалення категорій і витрат, і прибутку.

Меню внесення витрат та прибутку - мають однакові параметри: сума, категорія, рахунок, опис та дата.

На рисунку 3.3, за допомогою описаного вище функціоналу, є можливість побачити візуалізовану діаграму використання.

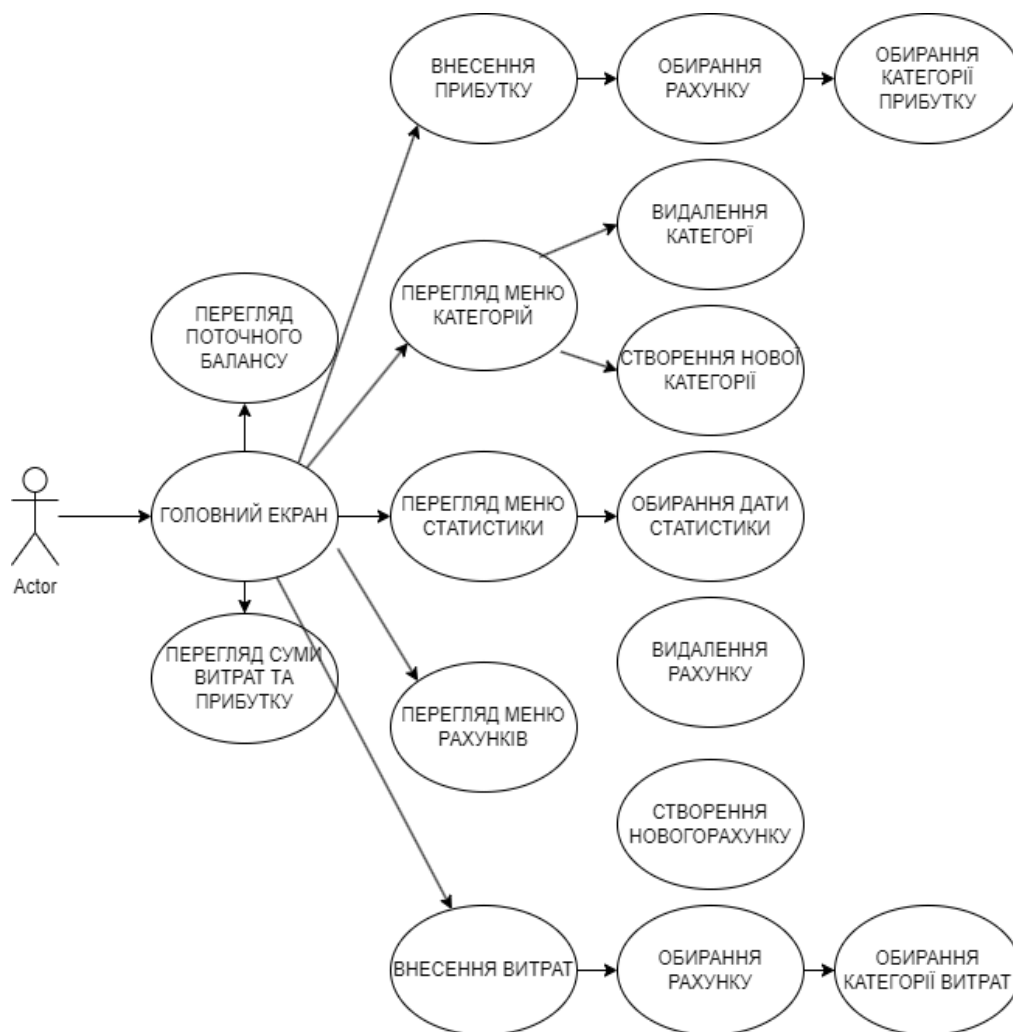


Рис. 3.3 Use case діаграма

3.3 Діаграма класів

Діаграма класів [13] дозволяє візуалізувати класи, що використовувалися під час створення додатку. При реалізації функціоналу застосунку використовувалися 4 класи: User, Accounts, Categories, Transactions.

Клас User - представляє собою інформацію про користувача, його електронну пошту та пароль для входу.

Клас Accounts - представляє собою користувацькі рахунки, які використовуються для зберігання інформації про поточний фінансовий стан, про проведенні транзакції.

Клас Transactions - представляє собою список транзакцій користувача: його доходи та витрати. Зберігає в собі дані про дату транзакції, рахунок та категорію.

Клас Categories - інформацій клас, для підведення статистики та проведення транзакцій. Вказує на вид витрат та доходів.

Реалізована на рисунку 3.5, діаграма класів вказує на зв'язки класів між собою та їх основні функції.

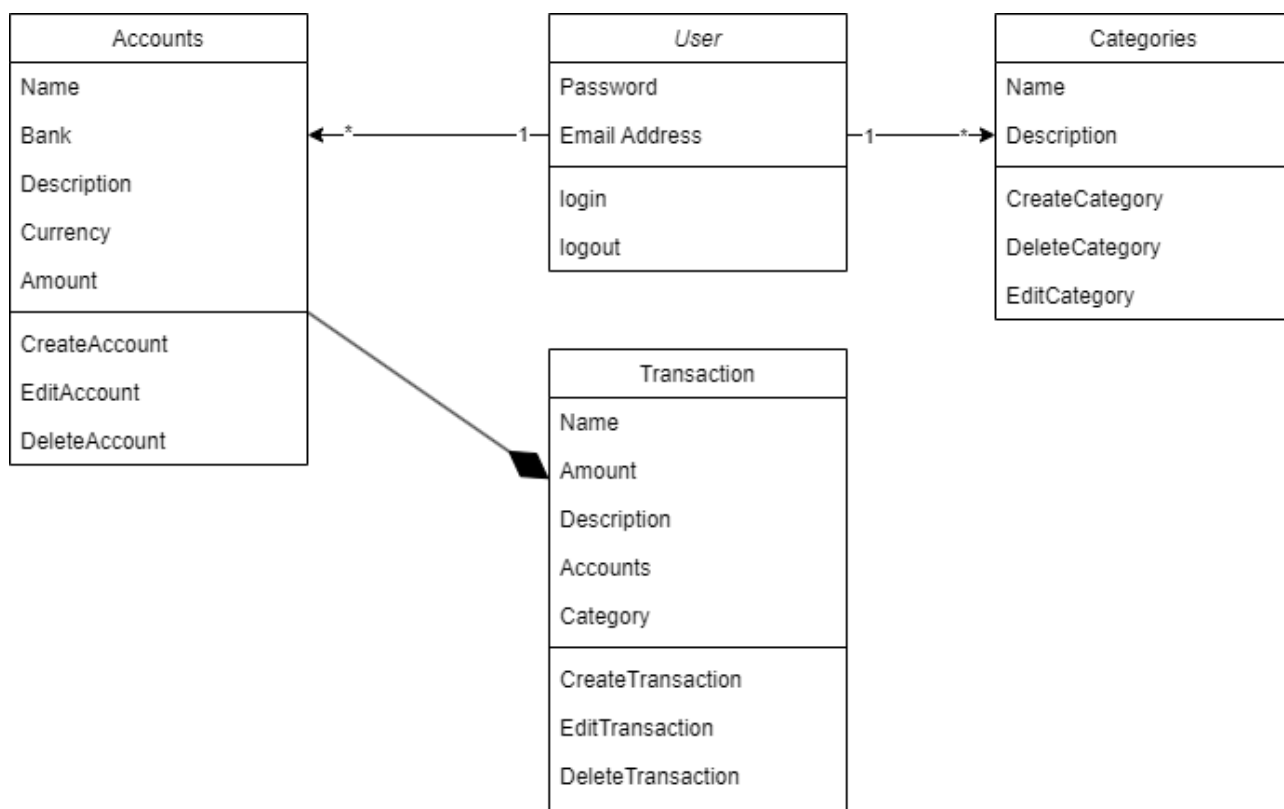


Рис. 3.5 Діаграма класів застосунку Sky Wallet

3.4 Схема роботи застосунку

Схема роботи, яка візуалізована на рисунку 3.6, застосунку Sky Wallet показує основні елементи, їх призначення та функціонал. Для початку розберемо які елементи включає в себе головний екран додатку:

- кнопка “Статистика”,
- кнопка “Рахунки”,

- кнопка “Профіль”,
- кнопка “Плюс”.

Після натискання на кнопку “Статистика” користувач має змогу вибрати параметри за якими буде переглядатися статистика:

- кнопка “Статистика доходів”;
- кнопка “Статистика витрат”;
- кнопка “Вибір дати”.

Кнопка “Рахунки” надає можливість створення нових рахунків, їх редагування та видалення. Однією з особливостей даного екрану є те, що користувач може побачити статистику кожного рахунку окремо, натиснувши на відповідний.

Редагування категорій здійснюється з екрану “Профіль”, натиснувши відповідну кнопку “Категорія”, здійснюється перехід до списку категорій витрат та доходів.

Кнопка “Плюс” - контекстну меню для вибору дії:

- “Витрата”;
- “Дохід”;
- “Обмін між рахунками”.

Функція “Обмін між рахунками” реалізована таким чином, щоб кожна така транзакція не відзначалась у статистиці як дохід або витрата, адже ці гроші тільки переміщуються між рахунками, що робить її дуже корисною під час аналізу певного періоду витрат для створення фінансового плану на наступний. Саме тому, цей функціонал є дуже корисним, коли користувач має багато рахунків, між якими постійно курсують фінансові операції. Наприклад, зняття готівки в банкоматі, баланс двох рахунків змінюється, але фінансовий стан - ні.

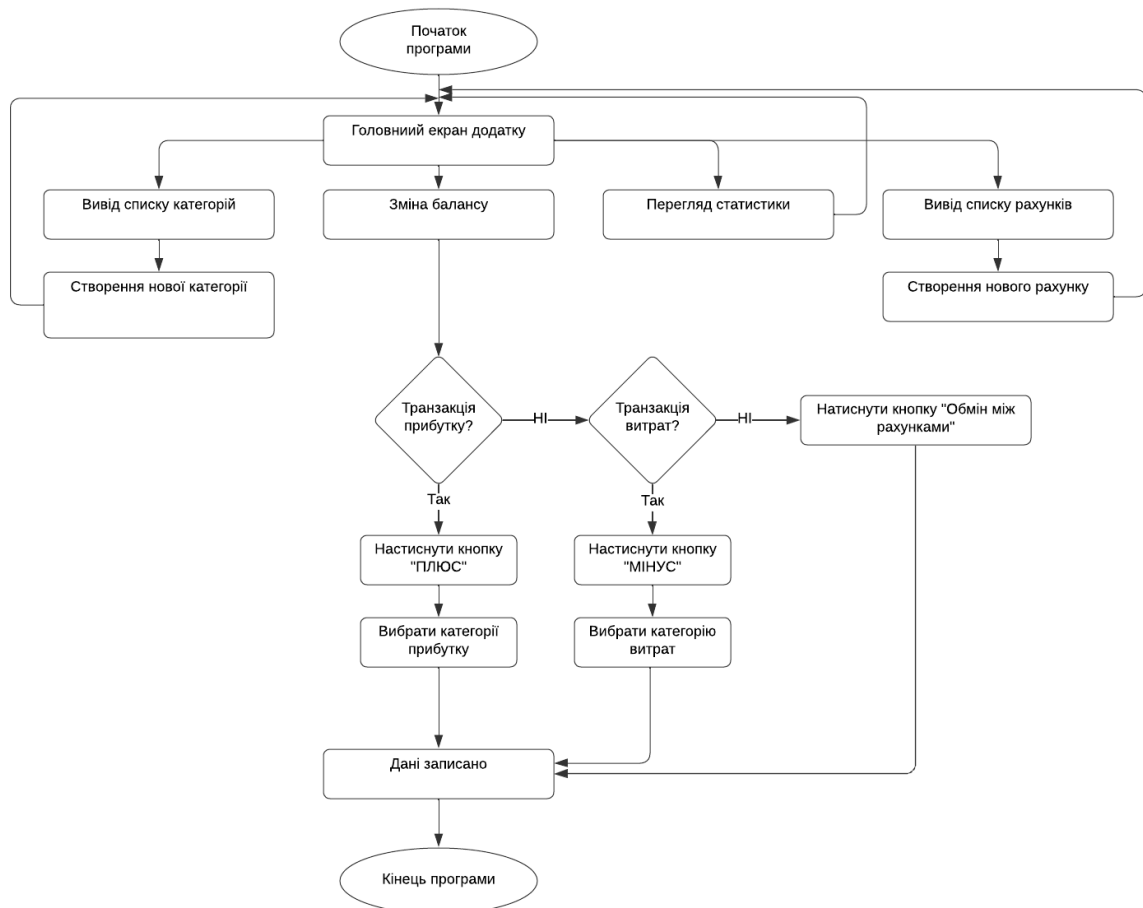


Рис. 3.6 Схема роботи застосунку Sky Wallet

3.5 Проєктування інтерфейсу користувача

Головною метою даного етапу розробки є створення зручного та інтуїтивно зрозумілого інтерфейсу, що сприятиме легкій навігації додатком та взаємодії користувача з основними функціями. Перш ніж розпочати проєктування необхідно проаналізувати, які основні компоненти повинні бути розміщені на головному екрані застосунка. Ці компоненти мають включати:

- верхня частина екрану повинна мати логотип додатку, для легкої ідентифікації застосунку;
- показники фінансового стану, а також загальні показники витрат та доходів, мають бути розміщені на головному екрані, для швидкого

перегляду; показники мають відображати значення у вигляді числових значень;

- кнопка “Плюс”, призначена для вибору між витратами, доходами та обміном між рахунками; ці кнопки мають дозволяти швидко та безпроблемно проводити фінансові операції, саме тому вони повинні знаходитися в нижній частині екрану;
- для навігації додатком, так само як і для кнопки “Плюс”, вибрано місце у нижній частині додатку.

Вибір місця розташування головним інтерактивних інтерфейсів є дуже важливим, адже від цього залежить наскільки швидко та зручно користувачі можуть використовувати додаток. Для вирішення цієї проблеми достатньо ознайомитися з дослідженнями [14] Стівена Хубера про те, як люди тримають свої телефони. Суть полягає в тому, що майже в кожному випадку були поширені три основні способи тримання телефону. За словами Джоша Кларка, 49% користувачів тримали телефони однією рукою, 36% - однією рукою, використовуючи вказівний або великий палець іншої руки для взаємодії з екраном. Решта 15% тримали телефон двома руками і набираючи текст великими пальцями. Стівен Хубер виявив, що 75% користувачів торкаються екрану тільки одним великим пальцем. Таким чином, виник термін "керований великим пальцем".

Провівши певні дослідження [15] у 2016 Саманта Ігрем розробила схему для “керування великим пальцем”, за результатом, зображеним на рисунку 3.7, нижню частину екрану було обрано найбільш доступним та зручним місцем для розміщення навігації та меню мобільного додатку.

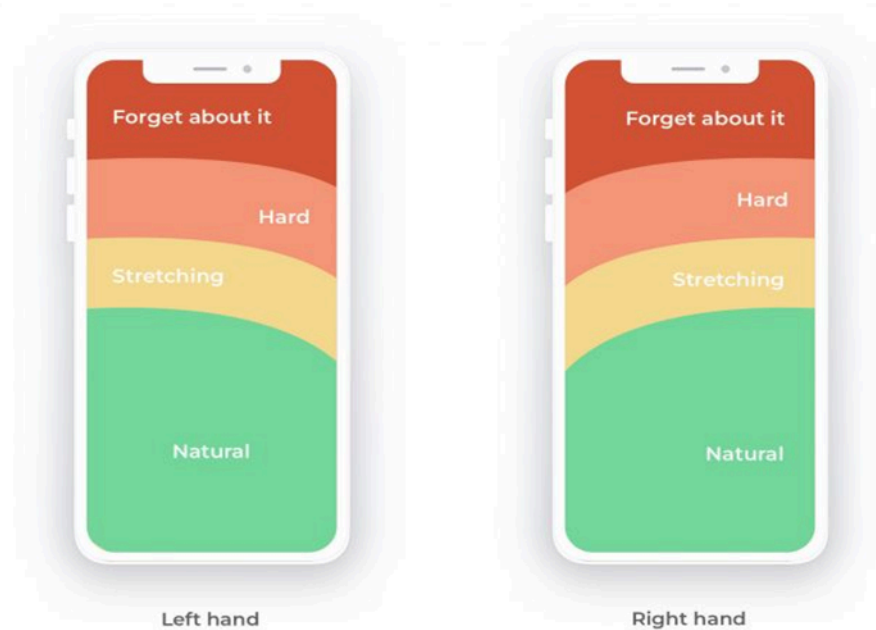


Рис. 3.7 Результат досліджень Саманти Ігрем.

1. Екран “Статистика” повинен відображати розподіл витрат за категоріями та рахунками.
2. Кнопка “Рахунок” має відкривати список із існуючими рахункам, а також функцією додати рахунок.

При розробці дизайну інтерфейсу було враховано принципи зручності та інтуїтивності. Елементи, що відповідають за контроль над функціями, розміщені в логічному порядку - це дозволяє користувачу швидко знайти потрібне вікно.

Проектування проводиться з урахуванням кросплатформеності, тестувалось на різних екранах. Було забезпечено адаптивність під різні види смартфонів та версії Android [16]. Нижче наведено кілька скріншотів застосунку (рис.3.8 – 3.10):

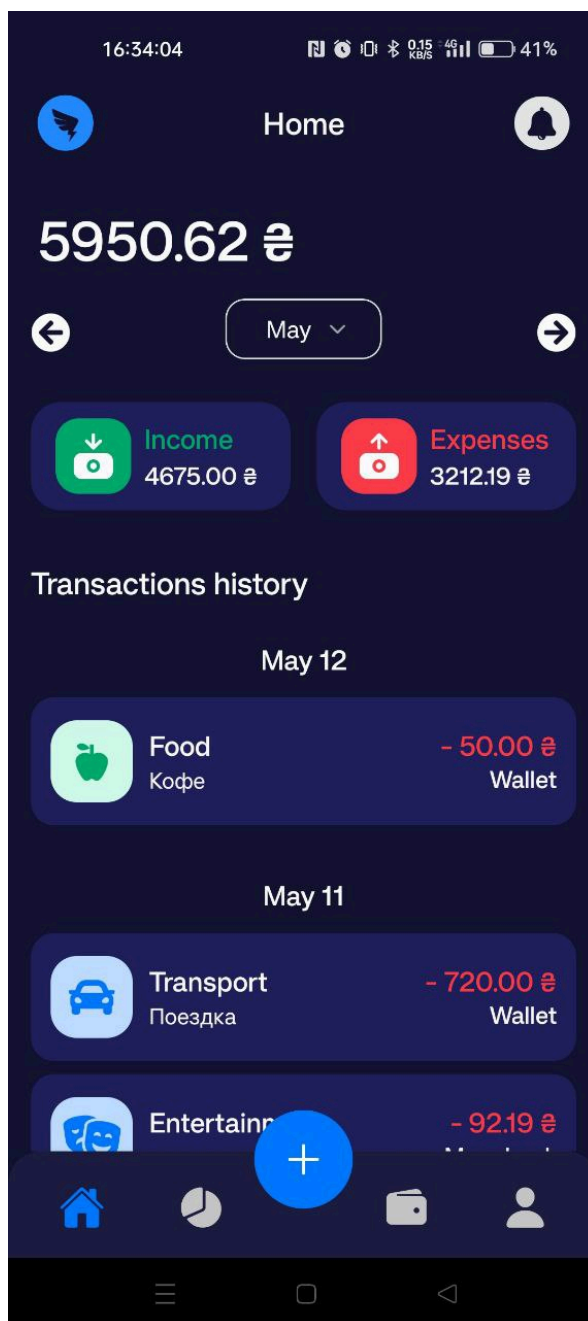


Рис.3.8 Скріншот головного екрану

Головний екран (рис. 3.8) представляє користувачу головні елементи керування, які були описані вище. Для кращого відображення наявного балансу, суми витрат та доходів, в застосунок одразу були введені дані.

Під час перегляду вікна статистики (рис.3.9), як вже було описано вище, користувач має змогу вибрати потрібні параметри перегляду.

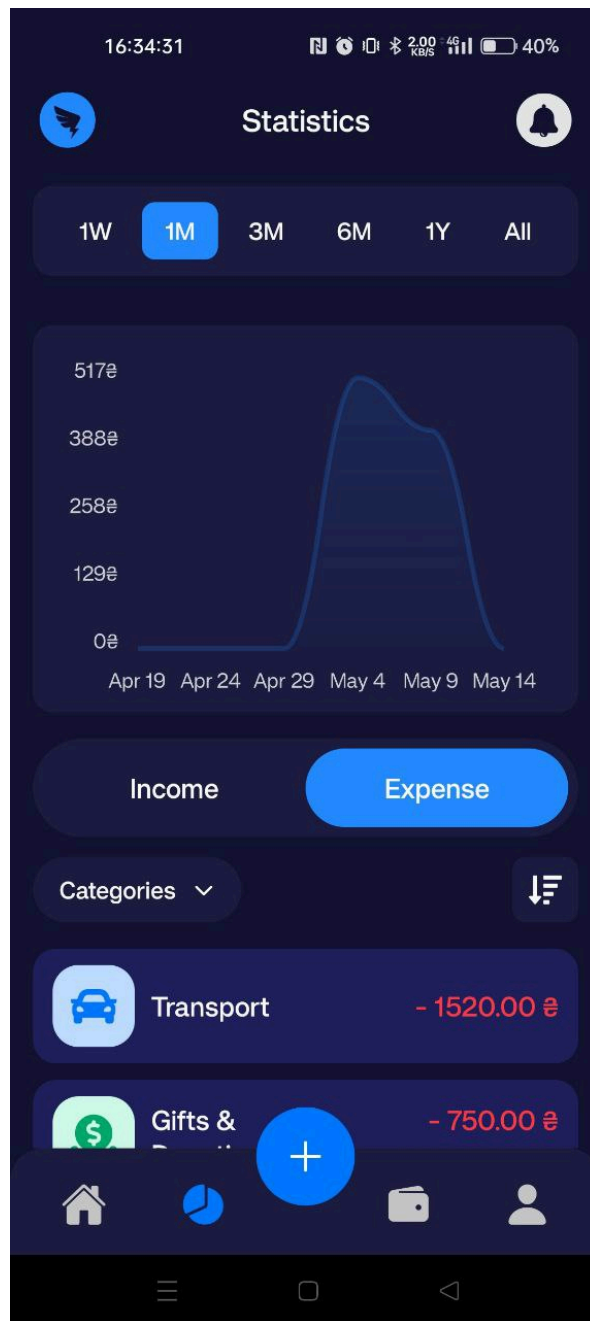


Рис. 3.9 Вікно статистики застосунку

Відображення рахунків виконано у режимі слайд-шоу. Це зроблено для того, щоб під кожним з наявних рахунків, користувач мав можливість перегляду виконаних їм транзакцій з кожного з них. Дизайн кожного з рахунків реалізовано у вигляді банківської картки, незважаючи на те, який тип рахунку був обраний під час створення. На рисунку 3.10 продемонстровано те, яким чином відбувається перегляд наявних рахунків.

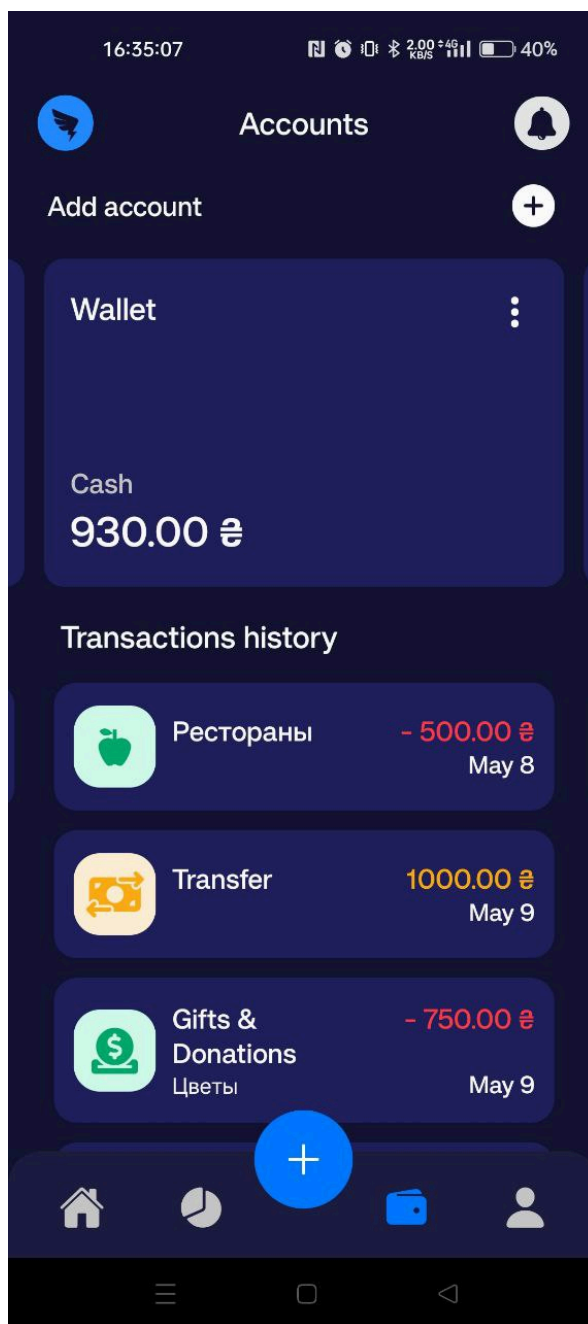


Рис. 3.10 Экран вікна “Рахунки”

При натисканні на “Категорії” відкривається вікно зі списком всіх доступних категорій. Вікно має 2 кнопки: “Витрати” - для показу списку категорій витрат, “Доходи” - список категорій прибутку. Під кожним зі списків є кнопка “Додати нову”, категорія додається у той список у якому ви зараз знаходитесь. Наприклад, користувач купив собі маленьке кошения і хоче записувати витрати, які стосуються тільки цього кошения. Він обирає список категорій витрат, натискає

“Додати нову” і створює потрібну йому категорію, вказуючи назву та обираючи відповідну іконку. Рисунок 3.11 демонструє список категорій витрат.

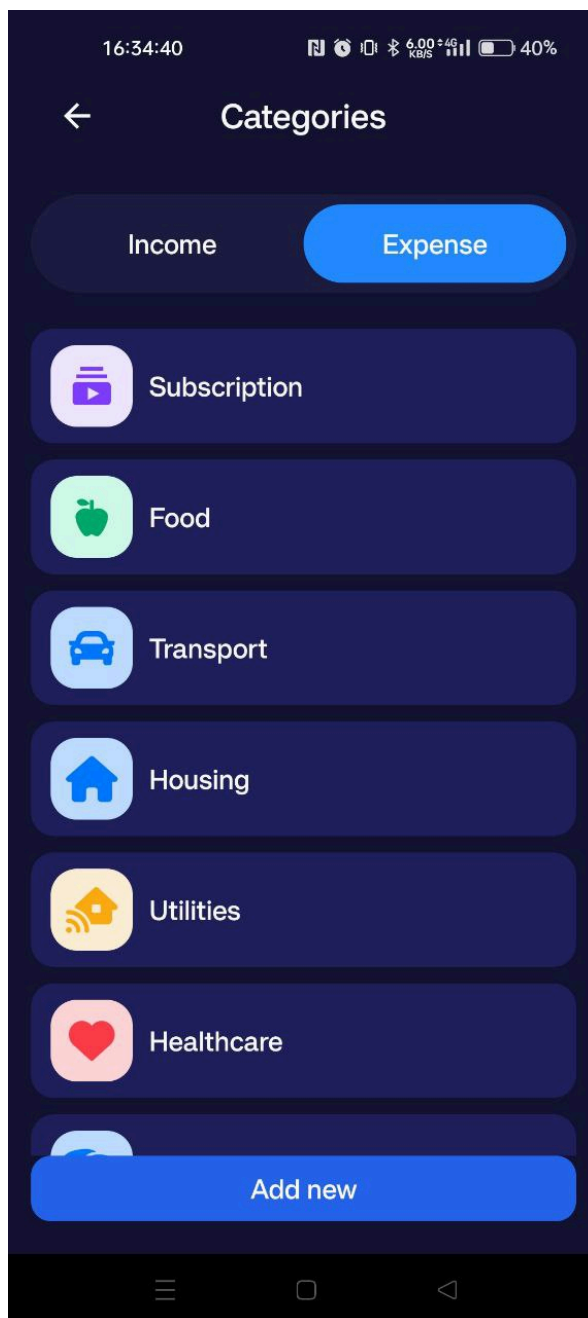


Рис. 3.11 Список категорій витрат

3.6 Реалізація функціональної частини

3.6.1 Створення навігації додатку

Для розробки інтерфейсу додатку були використані можливості фреймворку React Native, який надав необхідні засоби та інструменти для створення графічного інтерфейсу. Для початку було створено ряд з п'яти комірок у нижній частині екрану. Цей ряд призначений для меню (рис. 3.12) навігації додатку та має 5 комірок, кожна з яких відповідає за перехід до певного вікна застосунку: “Головний екран”, “Статистика”, “Плюс”, “Рахунки”, “Профіль”. Вибір місця був обґрунтований проведеними дослідженнями, описаними у попередньому розділі.

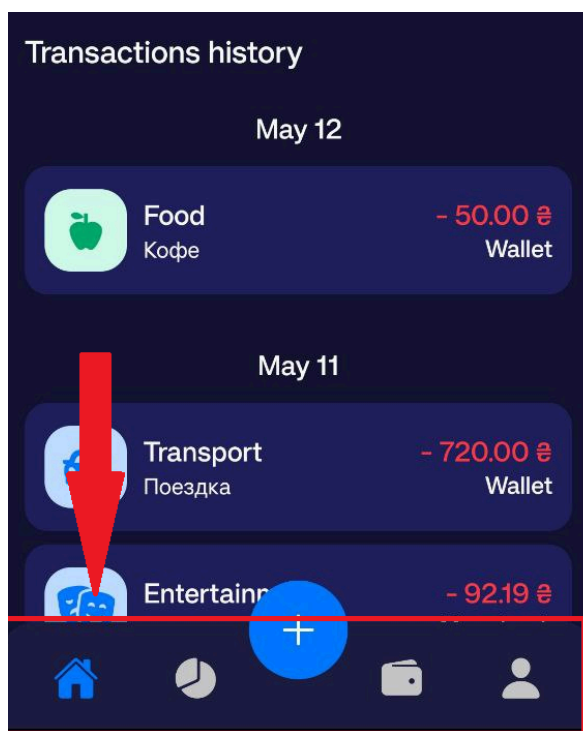


Рис. 3.12 Меню навігації застосунку

3.6.2 Створення загальних показників балансу

Під час розробки графічного інтерфейсу, було визначено які головні елементи повинні відображатися на головному екрані. Одним з таких елементів було меню відображення поточного стану балансу та показники загальних витрат та прибутків (рис. 3.13).

Загальні показники витрат та прибутків відображають користувачу загальну суму виконаних транзакцій. За замовчуванням, кожен місяць цей показник обнуляється, але користувач все одно має можливість перегляду даних за попередні місяці.

Поточний баланс відображає загальну суму користувача на рахунках. Не додається до поточного балансу, тільки сума тих рахунків, які були створені користувачем із використанням функції - “Ігнорувати в поточному балансі”.

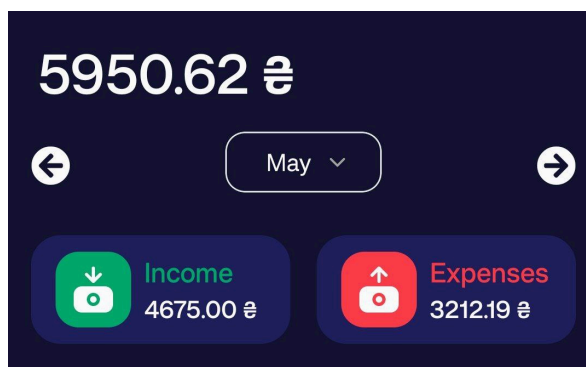


Рис. 3.13 Відображення балансів на головному екрані

3.6.3 Створення меню статистики

Ще одним головним елементом та функцією додатку є меню статистики (рис. 3.14). “Статистика” - дозволяє користувачеві переглядати всі виконані транзакції за певний період часу, також доступні певні параметри, що дозволяють перегляди доходи або витрати за певним видом категорії. Статистика представлена у вигляді стовпчикової діаграми.

Для полегшення перегляду було прийнято рішення не додавати параметр перегляду статистики по кожному з рахунків, адже раціональніше було відобразити під кожним з рахунків список всіх виконаних транзакцій, підписуючи кожен категорію під транзакцією.

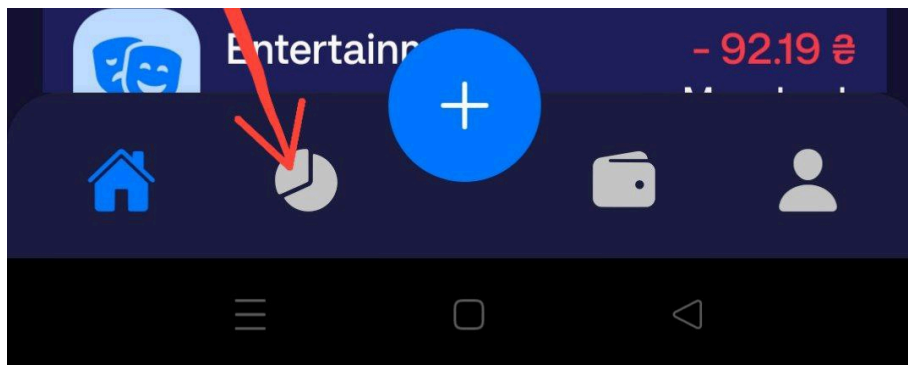


Рис. 3.14 Розташування кнопки для переходу у меню статистики

3.6.4 Створення кнопок "Плюс", "Мінус" та "Обмін між рахунками"

Ще одним основним компонентом програми з трекінгу фінансів є функціонал кнопок, які дають можливість користувачеві додавати або віднімати суму до/з балансу певного рахунку. Також однією з особливостей є функція обміну між рахунками, що реалізовано таким чином, щоб перевівши фінанси з одного рахунку на інший, різниця балансів не відображалась у загальній статистиці.

Меню реалізовано таким чином, щоб під час натискання на кнопку "Плюс", відкривалось діалогове меню в якому користувач матиме змогу вибрати потрібний йому пункт. Це показано на рисунку 3.15 нижче.

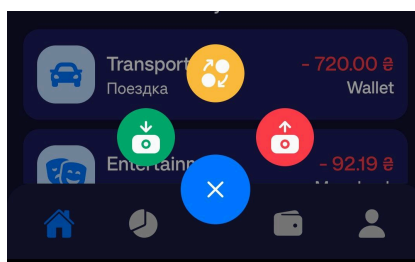


Рис. 3.15 Діалогове вікно меню вибору дії

Різниця кнопок “Витрати” та “Прибуток” полягає в декількох параметрах. Наприклад, під час натискання кожної з кнопок, користувач отримує список категорій створений спеціально для цієї функції, що продемонстровано на рисунку 3.16. Також, “Витрати” мають додатковий параметр у вигляді додавання файлу.

Такі поля параметрів як сума, опис, вибір дати та вибір потрібного рахунку, залишаються для кожної з функції однаковими.

Після введення всіх даних, користувач натискає кнопку “Далі”, що зберігає всі дані. Після проведення транзакції, вона також автоматично відображається у меню статистики, а також у списку проведених фінансових дій під кожним з користувацьких рахунків.

Як вже було зазначено [15], кнопка “Витрати” була розташована праворуч, ближче до великого пальця, адже за статистикою кількість витрат завжди є більшою, ніж кількість доходів.

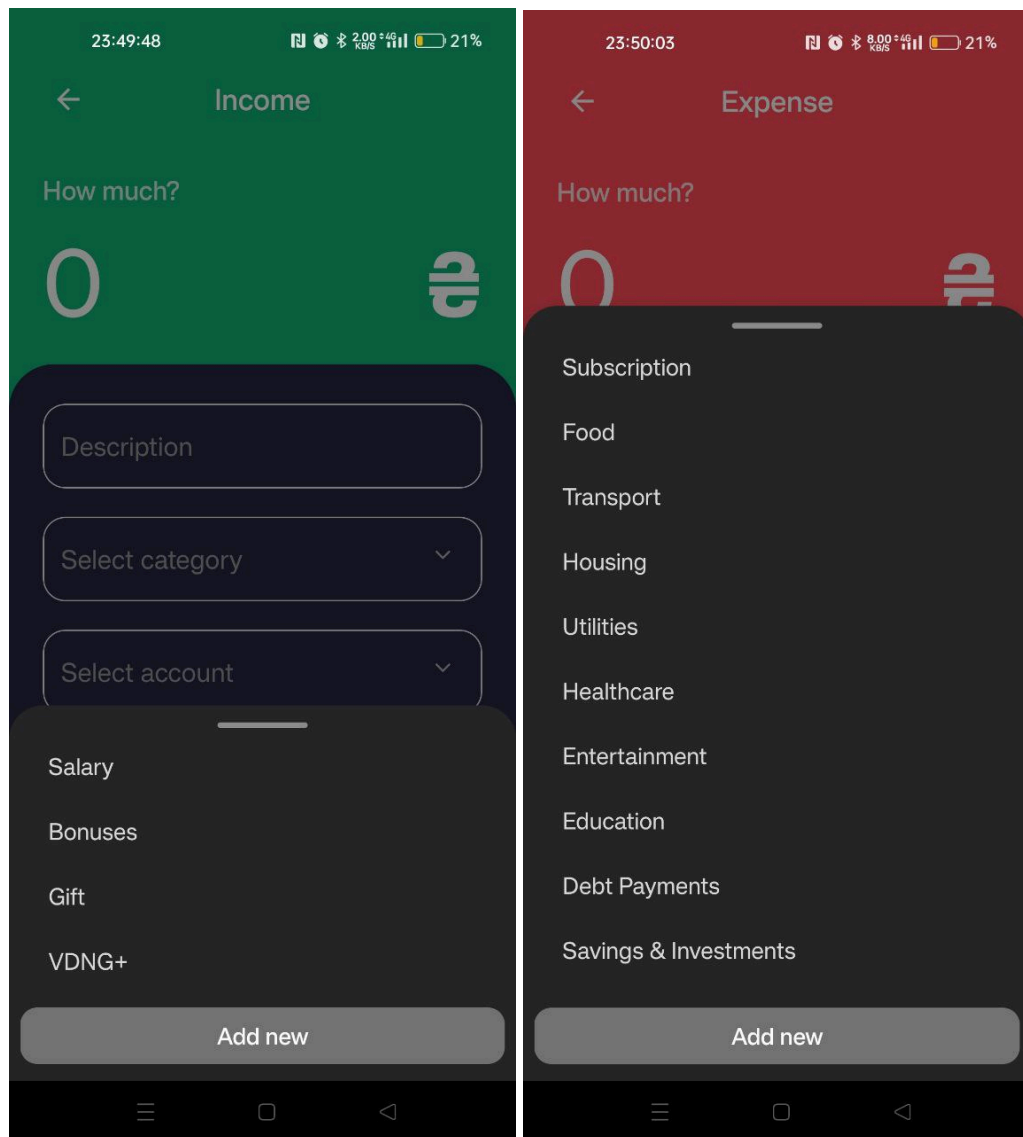


Рис. 3.16 Списки категорій для витрат та доходів

3.6.5 Створення кнопки профіль

Функціонал меню профілю включає в себе тільки дві основні функції. Перша - логін та логат, що дозволяє користувачу зберігати свої дані не тільки у локальному сховищу, розміщеному у пам'яті телефону, а й на сервері. Це зроблено для того, аби користувач при зміні пристрою, не втратив особисті дані. Друга - відкриття списку доступних категорій, під час перегляду яких є можливість створення нової категорії потрібного типу.

На рисунку 3.17 - 3.18 показано як виглядають екран профілю та параметри створення нової категорії.

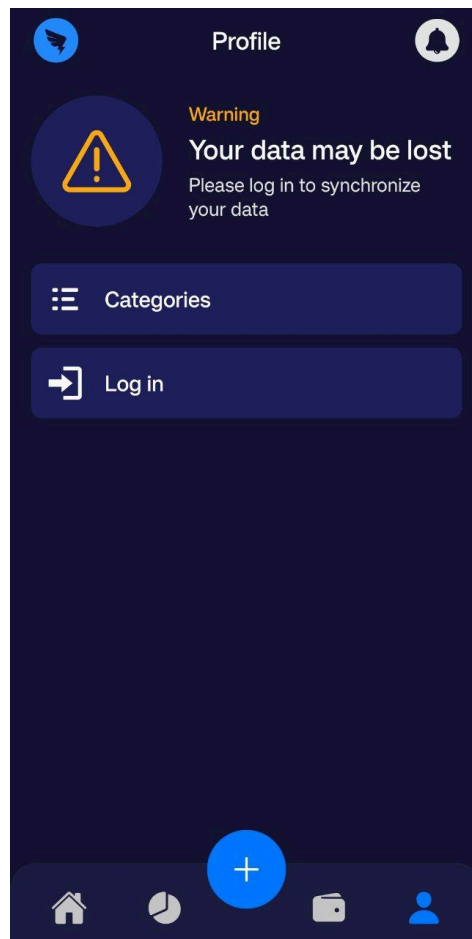


Рис. 3.17 Екран профілю

Для створення нової категорії (рис. 3.18) користувач повинен обрати потрібний йому список категорій, адже від цього залежить в який саме буде додана нова категорія, а також її назву та іконку, що допоможе надалі краще орієнтуватися по всьому списку.

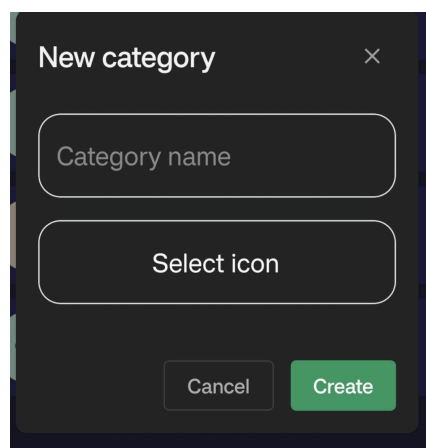


Рис. 3.18 Вікно створення нової категорії

3.7 Тестування додатку

Тестування програмного забезпечення є необхідною складовою процесу розробки будь-якого програмного продукту, тому що цей процес дозволяє виявити та виправити всі недоліки та баги, що можуть вплинути на функціональність, якість та швидкість роботи застосунку.

Для забезпечення якості тестування в умовах обмеженого часу під час розробки додатку Sky Wallet було проведено два етапи тестування:

- Тестування під час розробки: на цьому етапі виявлялися критичні та блокуючі помилки, які одразу ж виправлялися.
- Тестування готового продукту: після завершення розробки проводилося виявлення багів та виконувалися мінорні виправлення.

Під час розробки додатку помилки відображалися у консолі IDE Visual Studio Code, яке використовувалося як середовище розробки. Через великий розмір проекту та обсяг коду на останньому етапі тестування спочатку здійснювався дебаг. У разі виявлення помилок вони виправлялися. Після цього проводилося код-рев'ю і повторний дебаг. Якщо помилок більше не знаходили, програма запускалася для подальшої перевірки.

ВИСНОВКИ

Під час роботи над дипломним проектом було виконані усі поставлені задачі:

1. Проведено аналіз процесів, пов'язаний з керуванням та контролем особистих фінансів. Виділено основні засоби для організації контролю персональної економічної активності такі, як планування подумки, паперові засоби, таблиці Excel, Google Sheets, онлайн ресурси та мобільні додатки.

2. Більш детально проаналізовано існуючі додатки для трекінгу особистих фінансів, оцінено їх функціонал, інтерфейс та загальні характеристики. В результаті аналізу були виявлені як переваги, такі як детальна статистика, зручний спосіб керування, так і недоліки: обмежений функціонал, не зрозумілий інтерфейс. Результати даного дослідження були використані при плануванні та розробці мобільного додатку для відслідковування особистих фінансів, для уникнення популярних помилок та недоліків.

3. Обґрунтовано вибір програмних засобів та технологій для розробки мобільного додатку для відслідковування та керування особистими фінансами. В роботі використано бібліотеку Realm для локального збереження даних, React Native для створення користувацького інтерфейсу.

4. Розроблено мобільний додаток на основі обраних технологій і з урахуванням зазначених вимог.

5. Проведено тестування веб-застосунку. Застосунок відповідає зазначеним вимогам.

ПЕРЕЛІК ПОСИЛАНЬ

1. Фінансовий Менеджмент: навч. посіб. / І.А. Бігдан та ін. Харків: 2017.
2. Research Methods for Business Students by Mark N.K. Saunders, Philip Lewis, and Adrian Thornhill, 2019
3. Фінансова грамотність, фінансова інклюзія та фінансовий добробут в Україні у 2021. *Звіт за результатами дослідження. Вересень 2021 р. Київ, Україна.* URL: https://bank.gov.ua/admin_uploads/article/Research_Financial_Literacy_Inclusion_Welfare_2021.pdf?v=4
4. Monefy. Official Documentation. URL: <https://monefy.me/> (дата звернення 15.02.2024)
5. CoinKeeper. Official Documentation. URL: <https://about.coinkeeper.me/eng> (дата звернення 15.02.2024)
6. Сфери застосування мови програмування JavaScript. URL: <https://it-club.com.ua/applications-of-javascript/>
7. Visual Studio Code. Official Documentation. URL: <https://code.visualstudio.com/>
8. Android Studio. Official Documentation. URL: <https://developer.android.com/studio>
9. React. Official Documentation. URL: <https://ru.legacy.reactjs.org/>
10. React Native. Official Documentation. URL: <https://reactnative.dev/>
11. KanbanFlow. Official Documentation. URL: <https://kanbanflow.com/>
12. Юлія Каграманова. Dou Forum. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти. URL: <https://dou.ua/forums/topic/40575/>
13. Jade Morales. What is a UML class diagram? URL: <https://www.mindonmap.com/uk/blog/what-is-uml-class-diagram/>
14. Steven Huber. Human Factors Research. Design for Fingers, Touch, and People. URL:

<https://www.uxmatters.com/mt/archives/2017/03/design-for-fingers-touch-and-people-part-1.php>

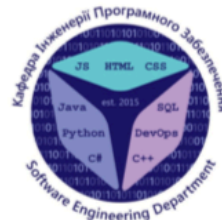
15. Samantha Ingram. Smashing Magazine. The Thumb Zone: Designing For Mobile Users. URL: <https://www.smashingmagazine.com/2016/09/the-thumb-zone-designing-for-mobile-users/>
16. The Art of Software Testing by Glenford J. Myers, Corey Sandler, and Tom Badgett, 2011.
17. Сокол А.Ю., Шевченко С.М. Трекер фінансів Sky Wallet. Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях», 24 квітня 2024 р., Київ, Державний університет інформаційно-телекомунікаційних технологій. Збірник тез. К.: ДУІКТ, 2024. С.67-68.

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка мобільного застосунку для керування фінансів Sky Wallet мовою
React Native та C#

Виконав студент 4 курсу
групи ПД-41
Сокол Антон Юрійович
Керівник роботи

К.п.н., доцент кафедри ІПЗ Шевченко Світлана Миколаївна
Київ – 2024

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ




- **Мета роботи:** поліпшення роботи з обліком власних фінансів
- **Об'єкт дослідження:** керування та контроль особистих фінансів.
- **Предмет дослідження:** технології розробки мобільного додатку для відслідковування та керування особистими фінансами.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Огляд та аналіз існуючих методів відслідковування особистих фінансів.
2. Аналіз існуючих рішень на ринку трекерів фінансів, їхніх функціональних можливостей, переваг та недоліків.
3. Проектування архітектури застосунку, забезпечення його кросплатформеності.
4. Проектування інтерфейсу користувача.
5. Розробка основних функцій.
6. Проведення тестування додатку на відповідність поставленим вимогам.

3

АНАЛІЗ АНАЛОГІВ

Назва	Переваги	Недоліки
Monefy 	<ul style="list-style-type: none"> • Інтуїтивний інтерфейс • Резервна копія на хмарному сервісі • Відсутність реклами • Можливість поставити пароль на вхід 	<ul style="list-style-type: none"> • Застарілий інтерфейс • Деякі спливаючі повідомлення закривають кнопки • Більшість корисного функціоналу у про-версії
CoinKeeper 	<ul style="list-style-type: none"> • Інтуїтивно зрозумілий дизайн • Великий вибір локалізації застосунку • Функція під'єднання реальної банківської картки 	<ul style="list-style-type: none"> • Набридлива реклама купівлі про-версії • Відсутність підказок • Майже всі корисні функції - платні
Sky Wallet 	<ul style="list-style-type: none"> • Інтуїтивний інтерфейс • Відсутність реклами • Відсутність платної версії • Локальне збереження даних без доступу в Інтернет • Здатність створювати рахунок у популярних валютах. 	<ul style="list-style-type: none"> • Відсутність української локалізації • Відсутність паролю на вхід

4

ВИМОГИ ДО ДОДАТКУ

Функціональні

1. Можливість до створення, редагування та видалення категорій витрат та доходів.
2. Створення рахунків у валютному еквіваленті.
3. Перегляд статистики, як за весь час так і за певний проміжок.

Нефункціональні

1. Кросплатформеність.

5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



TypeScript



Figma



6

USE CASE Діаграма



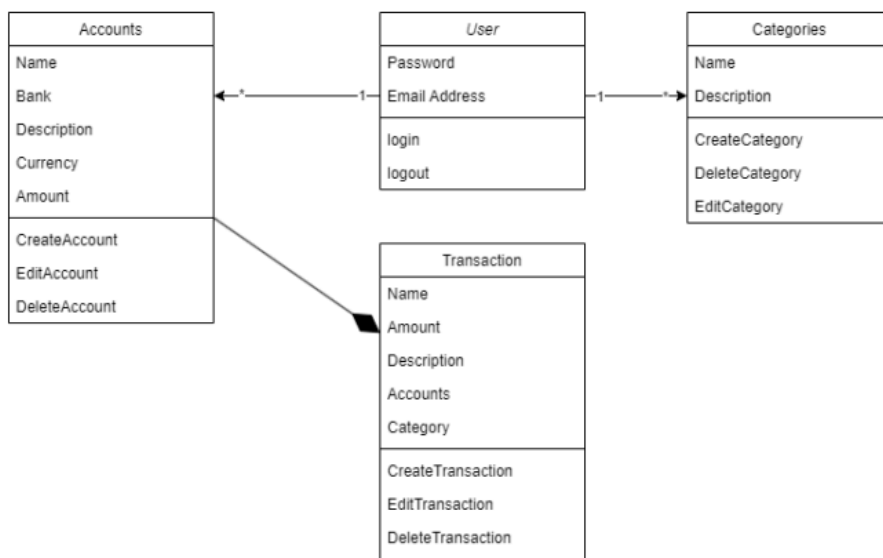
7

СХЕМА РОБОТИ ДОДАТКУ



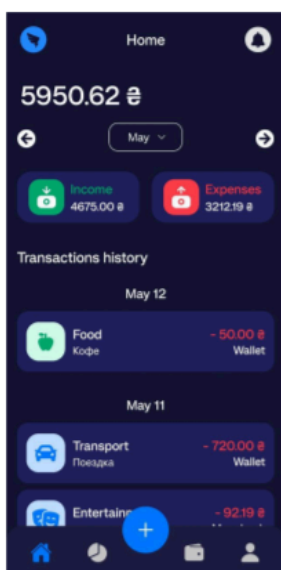
8

ДІАГРАМА КЛАСІВ

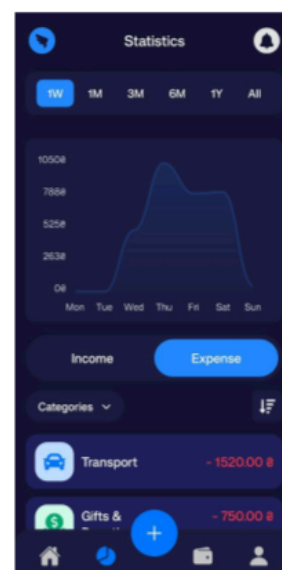


9

ЕКРАННІ ФОРМИ ДОДАТКУ



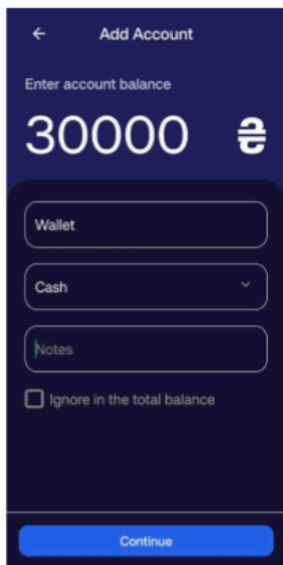
Головний екран додатку



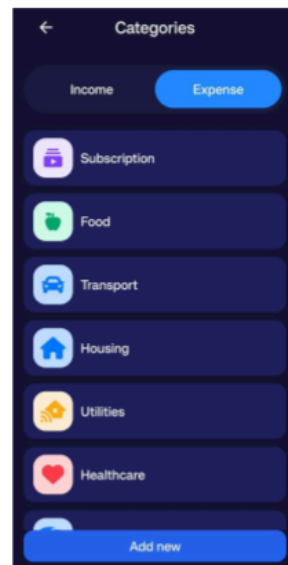
Екран перегляду статистики

10

ЕКРАННІ ФОРМИ ДОДАТКУ



Екран створення нового рахунку



Екран списку категорій

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Сокол А.Ю., Шевченко С.М. Трекер фінансів Sky Wallet. Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях», 24 квітня 2024 р., Київ, Державний університет інформаційно-телекомунікаційних технологій. Збірник тез. К.: ДУІКТ, 2024. С.67-68.

13

ВИСНОВКИ

Під час роботи над дипломним проєктом було виконані усі поставлені задачі:

1.Проведено аналіз процесів, пов'язаний з керуванням та контролем особистих фінансів. Виділено основні засоби для організації контролю персональної економічної активності такі, як планування подумки, паперові засоби, таблиці Excel, Google Sheets, онлайн ресурси та мобільні додатки.

2.Більш детально проаналізовано існуючі додатки для трекінгу особистих фінансів, оцінено їх функціонал, інтерфейс та загальні характеристики. В результаті аналізу були виявлені як переваги, такі як детальна статистика, зручний спосіб керування, так і недоліки: обмежений функціонал, не зрозумілий інтерфейс. Результати даного дослідження були використані при плануванні та розробці мобільного додатку для відслідковування особистих фінансів, для уникнення популярних помилок та недоліків.

3.Обґрунтовано вибір програмних засобів та технологій для розробки мобільного додатку для відслідковування та керування особистими фінансами. В роботі використано бібліотеку Realm для локального збереження даних, React Native для створення користувацького інтерфейсу.

4.Розроблено мобільний додаток на основі обраних технологій і з урахуванням зазначених вимог.

5.Проведено тестування веб-застосунок. Застосунок відповідає зазначеним вимогам.

14