

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного
забезпечення

ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської роботи
на ступінь вищої освіти бакалавр
на тему: «Розробка програмного забезпечення «Фінансовий
моніторинг» мовою Kotlin»

Виконав: студент 4 курсу,
групи ПД– 43
спеціальності
121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Шибасва Т.Д.

(прізвище та ініціали)

Керівник Поперешняк С. В.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтроль _____

(прізвище та ініціали)

Київ – 2022

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти - «Бакалавр»
Спеціальність - 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

О.В. Негоденко

“ ” _____ 2022 року

**ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**

Шибасвій Тамарі Дмитрівні

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка програмного забезпечення «Фінансовий моніторинг» мовою Kotlin»

Керівник роботи кандидат технічних наук Поперешняк С. В.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “18” лютого 2022 року
№ .

2. Строк подання студентом роботи 03.06.2022.

3. Вихідні дані до роботи:

3.1. Середовище розробки Android Studio;

3.2. Структура мобільного додатку для ОС Андроїд;

3.3. Науково-технічна література, пов'язана з розробкою;

3.4. Методи та принципи побудови мобільних додатків;

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

4.1. Аналіз предметної області

4.2. Аналіз програмного забезпечення для створення мобільного додатку

4.3. Огляд засобів реалізації для створення мобільного додатку

4.4. Створення та реалізація мобільного додатку

5. Перелік графічного матеріалу.

5.1. Титульний слайд

5.2. Мета, об'єкт та предмет дослідження

5.3. Актуальність роботи

5.4. Аналоги

5.5. Порівняння з аналогами

5.6. Технічне завдання

5.7. Програмні засоби реалізації

5.8. Інструменти використані для реалізації

5.9. Архітектура додатку

5.10. Апробація результатів дослідження

5.11. Висновки

5. Дата видачі завдання: 11.04.2022

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів бакалаврської роботи | Строк виконання етапів роботи | Примітка |
|-------|--|-------------------------------|----------|
| 1 | Підбір науково-технічної літератури | 19.04.2022 | Виконано |
| 2 | Провести огляд та аналіз існуючих найпопулярніших додатків для фінансового моніторингу | 20.04.2022 | Виконано |
| 3 | Визначити основні переваги та недоліки існуючих додатків для фінансового моніторингу | 21.04.2022 – 22.04.2022 | Виконано |
| 4 | Розробка власного додатку | 23.04.2022 – 27.04.2022 | Виконано |
| 5 | Висновки, оформлення роботи | 27.04.2022 – 30.04.2022 | Виконано |
| 6 | Розробка демонстраційних матеріалів | 30.04.2022 | Виконано |

| | | | |
|---|--------------------------|------------|----------|
| 7 | Попередній захист роботи | 03.06.2022 | Виконано |
| 8 | Здача роботи | 05.06.2022 | Виконано |

Студентка _____ Шибасва Т.Д.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Поперешняк С. В.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи: 55с., 23 рис, 1 табл., 12 джерел.

Ключеві слова: Android Studio, Kotlin, Firebase Realtime DataBase, ОС Андроїд, мобільний додаток, система контролю версій Git, аутентифікація користувача.

Об'єкт дослідження – розробка мобільного додатку для фінансового менеджменту.

Предмет дослідження – мобільне програмне забезпечення для фінансового менеджменту.

Мета роботи – розробка та підтримка протягом усього життєвого циклу мобільного програмного забезпечення для здійснення фінансового менеджменту.

Аналіз особливостей створення мобільних додатків для ОС Андроїд та сучасні можливості реалізації процесу фінансового менеджменту дозволяють сформулювати наступні вимоги до програмного продукту.

Мобільний додаток для здійснення фінансового менеджменту має відповідати таким вимогам:

- повинен працювати на всіх версіях ОС Андроїд, починаючи з 5.0 Lollipop (minSdkVersion 21);
- коректно відображатись на екрані будь-якого девайсу;
- виводити актуальну інформацію про доходи та витрати користувача;
- коректно відображати дані, що зберігаються у базі даних;
- перевіряти наявність нових даних під час входу в додаток або оновлення сторінки;
- динамічно виводити дані за категоріями відповідно до встановлених користувачем фільтрів;
- мати простий та інтуїтивно зрозумілий інтерфейс.

Галузь використання – додаток може бути використаний користувачем, що має телефон або планшет з ОС Андроїд та має на меті відслідковувати рух власних грошових ресурсів.

ЗМІСТ

| | |
|--|-----------|
| РЕФЕРАТ | 8 |
| ЗМІСТ | 9 |
| ВСТУП | 10 |
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ | 12 |
| 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ | 13 |
| 1.1. Загальний аналіз та опис предметної області | 13 |
| 1.2. Особливості розробки мобільних додатків | 13 |
| 1.3. Аналіз наявних на ринку додатків | 16 |
| 1.4. Висновки | 20 |
| 2. ВИМОГИ ТА ОЦІНКА ЯКОСТІ СИСТЕМИ | 23 |
| 2.1 Авторизація | 23 |
| 2.2. Обліковий запис користувача | 23 |
| 2.3. Внесення та збереження даних про грошові надходження | 23 |
| 2.4. Внесення та збереження даних про грошові витрати | 23 |
| 2.5. Перегляд даних про рух грошових коштів користувача | 23 |
| 2.4. Діаграми використання | 23 |
| 3. Проектування та реалізація | 25 |
| 3.1. Інструменти для розробки | 25 |
| 3.1.1. Опис середовища розробки Android Studio | 25 |
| 3.1.2. Git | 27 |
| 3.1.3. Firebase Realtime Database | 28 |
| 3.2. Архітектура системи | 29 |
| 3.2.1. Загальні принципи побудови андроїд додатків | 29 |
| 3.2.2. Шаблон проектування Model-View-ViewModel | 30 |
| 3.2.3. Зберігання даних користувача за допомогою Firebase | 35 |

Realtime Database та автентифікація за допомогою**FirestoreAuth**

| | |
|---|-----------|
| Практичний результат дослідження | 38 |
| Висновки | 42 |
| Список використаної літератури | 42 |
| Додаток | 43 |

ВСТУП

Актуальність теми: Кожна сучасна людина, особливо у непростий період фінансових криз, має приділяти значну увагу плануванню власного родинного бюджету для максимально ефективного використання наявних грошових коштів. Для здійснення такого планування необхідно безперервно слідкувати за поточними витратами та доходами. Але безперервний моніторинг фінансів може викликати певні складнощі у пересічній людини. Для вирішення даної проблеми можна скористатися фінансовим додатком: з телефоном, який завжди є під рукою, вірогідність того, що користувач буде забувати слідкувати за своїм бюджетом, вносити оновлення, слідкувати за доходами та витратами і ставити пріоритети щодо покупок значно меншає.

Ступінь вивчення проблеми: За даними, що наводить Pew Research Center, до кінця 2015 року 68% дорослих американців мали смартфон, а кількість власників планшетів зросла до 45%. Також в останні роки вченими було встановлено, що на даний час деякі вікові групи навіть наближаються до точки насичення смартфонами – до 86% людей віком від 18 до 29 років мають смартфон. Незважаючи на те, що наведені вище дані стосуються Америки, але подібна тенденція зберігається і для решти регіонів світу (особливо європейського, східно-азійського та Австралії).

За даними американської компанії Гугл, кількість активних користувачів ОС Андроїд на 2019 рік сягає 2,5 млрд.

Саме тому мобільні додатки для ОС Андроїд в останні роки набули такої популярності. Також наведені цифри свідчать про те, що із зростанням кількості активних користувачів зріс також і попит на мобільні додатки, що полегшують вирішення побутових проблем користувачів, зокрема найбільшою популярністю відзначаються саме додатки для фінансового менеджменту.

Джерела дослідження: Основними джерелами для теоретичного дослідження була навчально-наукова література, в якій міститься інформація про

особливості функціонування ОС Андроїд, а також про розробку додатків для даної ОС та інтернет ресурси на тему програмування потрібного функціоналу..

Об'єктом дослідження є розробка мобільного додатку для фінансового менеджменту.

Предметом роботи мобільне програмне забезпечення для фінансового менеджменту.

Метою роботи є розробка та підтримка протягом усього життєвого циклу мобільного програмного забезпечення для здійснення фінансового менеджменту.

Завданням роботи є проектування, розробка та подальше тестування мобільного додатку для фінансового менеджменту.

Методика дослідження: спочатку було проведено дослідження та виявлення основних недоліків вже існуючих на ринку додатків. Далі, на основі отриманих даних було створено мобільний додаток з інтуїтивно зрозумілим інтерфейсом, щоб користувачі могли з легкістю вносити дані про свої доходи та витрати, а також розподіляти їх за достатньою кількістю категорій.

Наукова новизна роботи: таким чином, наукова новизна полягає в створенні мобільного додатку, що надає користувачеві змогу створювати достатню кількість категорій для доходів та витрат.

Практична значущість результатів: даний продукт орієнтований на можливість користуванням ним всіх вікових категорій, але насамперед націлений на людей працездатного віку, які отримують власний дохід.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

SDK – software development kit

API – Application Programming Interface

JSON – Java Script Object Notation

БД – База Даних

ОС - операційна система

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Загальний аналіз та опис предметної області

На сучасному етапі розвитку інформаційних технологій розробка мобільних додатків переросла із вузько направленої ніші програм, що покликані вирішувати специфічні проблеми до програмного забезпечення, яке широко використовується у повсякденному житті. Без мобільних додатків неможливо уявити життя сучасної людини, адже, завжди маючи під рукою мобільний пристрій, ми щодня користуємося поштою, месенджерами, нотатками тощо.

Android - операційна система, розроблена компанією Android Inc, що належить Google, вперше побачила світ у вересні 2008 року. Заснована на ядрі Linux, для написання програм використовується мова Java. Система використовується на величезному спектрі пристроїв, це смартфони, планшети, музичні плеєри, розумний годинник, і телевізори, в планах компанії встановлення Android навіть на автомобілі. Дана операційна система є відкритою, тобто її вихідний код знаходиться у відкритому доступі, і будь-хто може з ним ознайомитися. Є найпопулярнішою системою на ринку мобільних девайсів, пов'язано це з тим, що її використовують практично всі виробники мобільної електроніки, і багато девайсів на Android значно дешевші за своїх конкурентів. Для встановлення нових програм користувач може скористатися магазином Google Play, але для цього необхідно обов'язково створювати Google обліковий запис з підтвердженням ліцензійної угоди. Але також у системі є можливість встановлення нових програм і прямисінько з пам'яті пристрою або SD карти

Окремою категорією стоять мобільні додатки, що допомагають користувачу відслідковувати рух власний грошових коштів, тобто наочно побачити суми доходів та витрат за обраний період.

1.2. Особливості розробки мобільних додатків

Розробка програм для Android, а також розробка програм для iPhone мало чим відрізняються. Хоча основним спірним моментом є програмне забезпечення, алгоритми розробки програм нічим не відрізняються.

Спочатку робиться прототип. Розробка програм під Android буде якісною лише у разі детального промальовування макету за дорученням замовника. Останній зможе наочно побачити очікуваний результат, внести зміни на папері набагато простіше, ніж переписувати код.

Далі йде розробка додатку для Android повинна включати створення прототипу. Якщо макет влаштовує, наступним кроком у розробці програм для iOS або Android буде макет. Необхідно проаналізувати основні конкуруючі програми: їх козири, інтерфейс, особливості. Потрібно додати щось своє, у крайньому випадку, щоб бути не гіршим за інших.

Розробка Android-додатків може бути 3 видів:

- Нативна - написані рідною для певної платформи мовою програмування. Для Android цими мовами є Java та Kotlin. Нативні додатки знаходяться на самому пристрої, доступ до яких можна отримати, натиснувши на іконку. Вони встановлюються через магазин додатків (Play Market на Android, App Store на iOS та ін.). Вони розроблені спеціально для конкретної платформи і можуть використовувати всі можливості пристрою – камеру, GPS-датчик, акселерометр, компас, список контактів та все інше. Також вони можуть розпізнавати стандартні жести, встановлені операційною системою або зовсім нові жести, які використовуються в конкретному додатку. Через те, що нативні додатки оптимізовані під конкретну ОС, вони органічно вписуються у будь-який смартфон, відрізняючись високою швидкістю роботи та продуктивністю.
- Із використанням WebView, що фактично показує користувачу власне веб-додаток
- Гібридна - на основі React Native – середовища розробки, заснованої на JavaScript, яка дозволяє створювати додатки з використанням бібліотеки блоків, які можуть обслуговуватися як для iOS, так і для Android. Такі додатки в

основному є комбінацією нативних та веб-елементів.

| Плюси нативних додатків: | Мінуси нативних додатків: |
|--|---|
| <ul style="list-style-type: none"> • швидкість роботи і продуктивність; • високий ступінь безпеки; • розширений інтерфейс; • максимально можлива функціональність; • здатність працювати без Інтернету; • зручність для кінцевого користувача. | <ul style="list-style-type: none"> • охоплення платформ; • тривалі терміни розробки; • відносно висока вартість розробки; • необхідність випускати оновлення в косметичних цілях. |

Рисунок 1. Переваги та недоліки нативних додатків

Розробникам також слід пам'ятати, що всі додатки для ОС Андроїд мають спільні особливості:

- мобільні пристрої мають невеликий розмір екрану, маленьку клавіатуру та обмежені пропускну здатністю для мультимедійних файлів. Необхідно враховувати, щоб дизайн мобільної інформації враховував ці особливості, а не просто повторював існуючу структуру звичайного контенту. Наприклад, актуальною є розробка мобільних версій наукових, навчальних сайтів, які будуть забезпечувати швидкий доступ до них, а також будуть використовувати оптимальні параметри саме для роботи з інформацією за допомогою мобільних пристроїв;
- при розробці мультимедіа, необхідно враховувати формати для фото, аудіо та відео файлів, що підтримуються мобільними пристроями.
- найбільш ефективною є розробка навчальних програм, розрахованих на 10-15 хвилин навчання. Зокрема, короткі модулі навчальних ресурсів академії Хана (Khan Academy), що виконують вказані рекомендації, оскільки мобільний пристрій використовується саме у перервах між основними видами діяльності, і саме через це мобільне навчання має займати короткі проміжки часу;
- потрібно не перевантажувати мобільні навчальні матеріали великими мультимедіа, адже саме вони впливають на швидкість завантаження даної інформації;
- потрібно постійно слідкувати за оновленням бібліотек та вчасно реагувати на зміну в політиці конфіденційності гугла.

1.3. Аналіз наявних на ринку додатків

Після аналізу кількох найбільш актуальних існуючих мобільних розробок були визначені основні переваги та недоліки доменних додатків, зокрема Finance PM, Money Manager, Bluecoins, Alzex finance.

Finance PM



Рисунок 2. Логотип додатку Finance PM

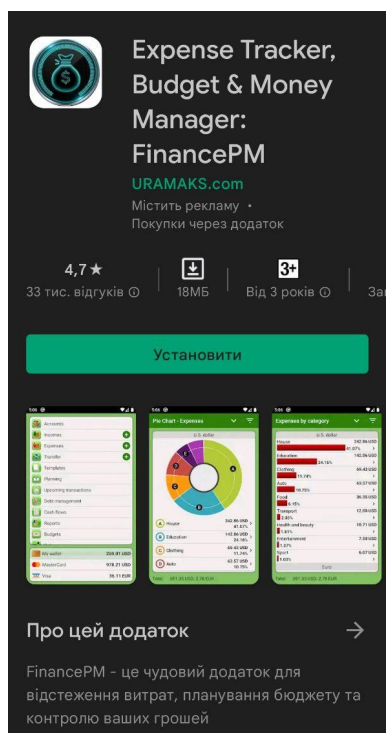


Рисунок 3. Сторінка додатку Finance PM в PlayMarket

За допомогою цієї програми ви зможете зрозуміти, скільки грошей витрачаєте на охорону здоров'я, освіту, автомобіль та інші речі. Ви також можете відкласти достатню суму на відпустку або житло. Додаток дає змогу контролювати та зменшувати свої витрати.

За допомогою Finance PM ви можете керувати будь-якою кількістю електронних гаманців та рахунків у різних валютах. Ви можете накопичити кошти на будь-яку свою мрію.

Для зручності відстеження ваших фінансів рекомендуємо встановити ліміт витрат (бюджет) на місяць/день/період. Це дозволить вам знати, на скільки ви перевищили ліміт витрат. Ви можете зменшити свої витрати, якщо є якісь перевитрати.

Money Manager



Рисунок 4. Логотип додатку Money Manager

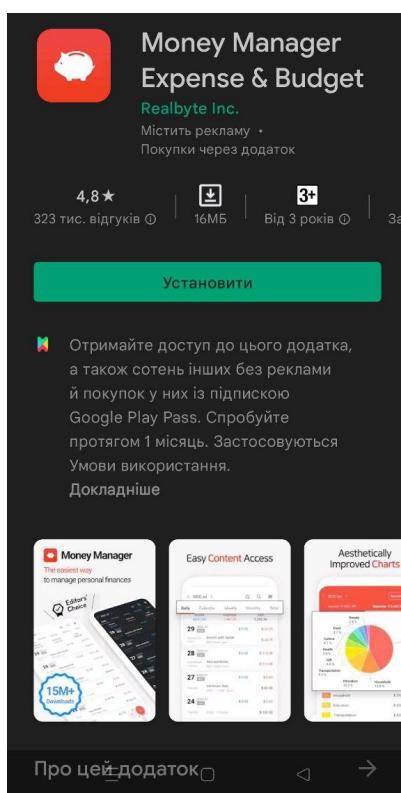


Рисунок 5. Сторінка додатку Money Manager в PlayMarket

Застосування системи подвійного бухгалтерського обліку

Money Manager сприяє ефективному управлінню активами та обліку. Він не просто реєструє ваші гроші, які надходять і виходять з вашого рахунку, але і вносить ваші гроші на ваш рахунок, щойно ваш дохід буде введено, і знімає гроші з вашого рахунку, як тільки ваші витрати будуть введені.

Основними функціями додатку є:

- Функція управління бюджетом і витратами - Money Manager показує ваш бюджет і витрати за допомогою графіка, щоб ви могли швидко побачити суму своїх витрат у порівнянні з вашим бюджетом і зробити відповідні фінансові висновки
- Функція управління кредитною / дебетовою карткою - вводячи дату розрахунку, ви можете побачити суму платежу та заборгованість на вкладці активів. Ви можете організувати автоматичне дебетування, просто підключивши свою дебетову картку до свого облікового запису.
- Код доступу - можна перевірити код доступу, щоб безпечно керувати своєю книгою фінансового огляду за допомогою Money Manager

Bluecoins



Рисунок 6. Логотип додатку Bluecoins

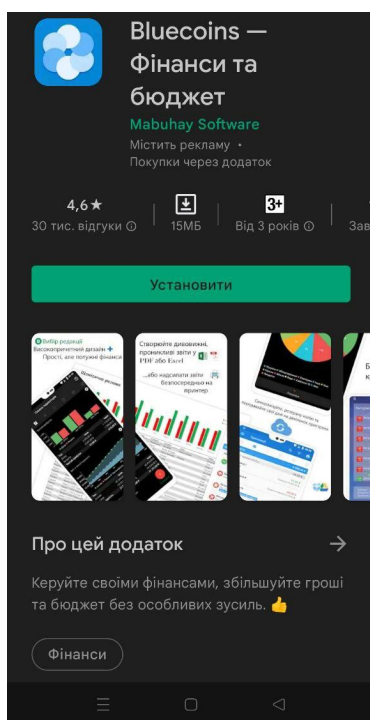


Рисунок 7. Сторінка додатку Bluecoin в PlayMarket

Він має набір інструментів звітності та аналізу для ефективного управління вашими грошима, витратами, доходами та бюджетом. Використовуйте це для своїх особистих фінансів, контролю сімейного бюджету або навіть для власного малого бізнесу. Створюйте щомісячні звіти у форматі PDF про свої витрати, керуйте своїм бюджетом разом із партнерами або експортуйте свої дані в електронні таблиці.

Alzex finance



Рисунок 8. Логотип додатку Alzex finance

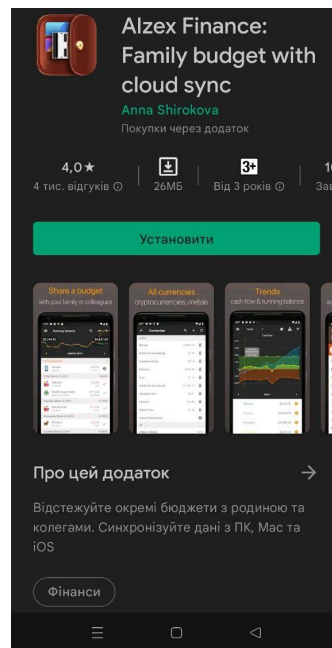


Рисунок 9. Сторінка додатку Bluecoin в PlayMarket

Ви можете відразу побачити, як були витрачені ваші гроші, загальну суму та структуру відсотків по категоріях, залишок на рахунках, а також загальний баланс.

Ключовими рисами даного додатку є:

- Валюти - програма підтримує необмежену кількість валют. Спочатку в списку вже є майже всі валюти світу, плюс дорогоцінні метали. Також ви можете додавати власні валюти, в тому числі «віртуальні», тобто ви можете відстежувати гроші та інші речі (метали, бензин, одиниці виміру). Курси валют можна завантажити з Інтернету.
- Рахунки - це основний спосіб розділення транзакцій у програмі. Ви можете створити будь-яку кількість рахунків, наприклад, кредитні картки, банківські рахунки, готівка, електронні гроші тощо. Рахунки можна групувати.

1.4. Висновки

За результатами проведених досліджень було виявлено основні переваги та недоліки програм-конкурентів на ринку мобільних додатків, що мають на меті допомогти користувачу відслідковувати рух грошових коштів.

За результатами дослідження складемо таблицю, в якій відобразимо всі переваги та недоліки даного ПО.

Таблиця 1. Огляд існуючих аналогів (переваги та недоліки)

| Назва додатку | Переваги | Недоліки |
|---------------|--|--|
| Finance PM | Наявне планування та нагадування | Немає автоматичного оновлення курсу валют, розпізнавання смс, інтеграції з банками, лише один рівень категорій |
| Money Manager | Наявні групи рахунків, бекап в гугл диск, sso | Немає розпізнавання смс, інтеграції з банками, лише один рівень категорій |
| Bluecoins | В преміум акаунті доступні розпізнавання смс, прогноз погоди, бекап в гугл диск | Лише один рівень категорій, більшість функцій доступні лише с преміум підпискою, лише один рівень категорій |
| Alzex finance | В платній версії є сканер чеків (тестування даної функції до покупки преміум аккаунта неможливе) | Немає імпорта смс, автоматичного розрахунку відсотків, лише один рівень категорій |

Основними перевагами проаналізованих додатків є функціонал, що дозволяє тримати бюджет «в хмарі», планувати фінансову діяльність на довгострокову перспективу, синхронізувати додаток з картою, розпізнавання смс, прогноз погоди, бекап в гугл диск. Серед переваг також слід відзначити такі реалізовані функції інтерфейсу, як бічна «завіса» та графіки, які є круговими діаграмами для більш наочної презентації.

Основними недоліками проаналізований вище додатків є їх складність та занадто завантажений інтерфейс, які були максимально спрощені в програмі, розробленій за темою даної роботи. Також значний відсоток користувачів відзначав, що наявність надлишкової реклами та сервісів, які не потрібні для

роботи з цими додатками значно знижують рівень задоволення від використання даних додатків.

Таким чином, аналіз конкурентних додатків дозволить вам врахувати всі зазначені нюанси при розробці власного додатка.

2. ВИМОГИ ТА ОЦІНКА ЯКОСТІ СИСТЕМИ

2.1. Авторизація

Для користування застосунком, користувач повинен мати змогу авторизуватись у системі для подальшого збереження введених даних. а також мати змогу увійти в свій обліковий запис. Для створення запису обов'язково повинні використовуватися електронна пошта і пароль.

2.2. Обліковий запис користувача

Зареєстрований користувач повинен мати можливість підтвердити свою електронну поштову адресу, а також бачити та змінювати своє ім'я. За допомогою використання облікового запису користувач має мати змогу переглядати інформацію про рух грошових коштів з будь-якого свого девайсу.

2.3. Внесення та збереження даних про грошові надходження

Користувач має мати змогу внести в програму дані щодо своїх доходів за певний період. Сутність доходу має мати таку інформацію, як сума доходу, джерело походження грошових коштів (наприклад, заробітна плата, дивіденди, роялті, дохід за відсотками свого вкладу в банку), дата отримання грошових надходжень.

2.4. Внесення та збереження даних про грошові витрати

Користувач має мати змогу внести в програму дані щодо своїх витрат за певний період. Сутність витрат має мати таку інформацію, як сума витрат, категорія витрачання коштів, дата здійснення операції.

2.5. Перегляд даних про рух грошових коштів користувача

Користувач має мати змогу переглядати статистичні дані про внесені суми доходів та витрат за допомогою діаграми.

2.6. Діаграми використання

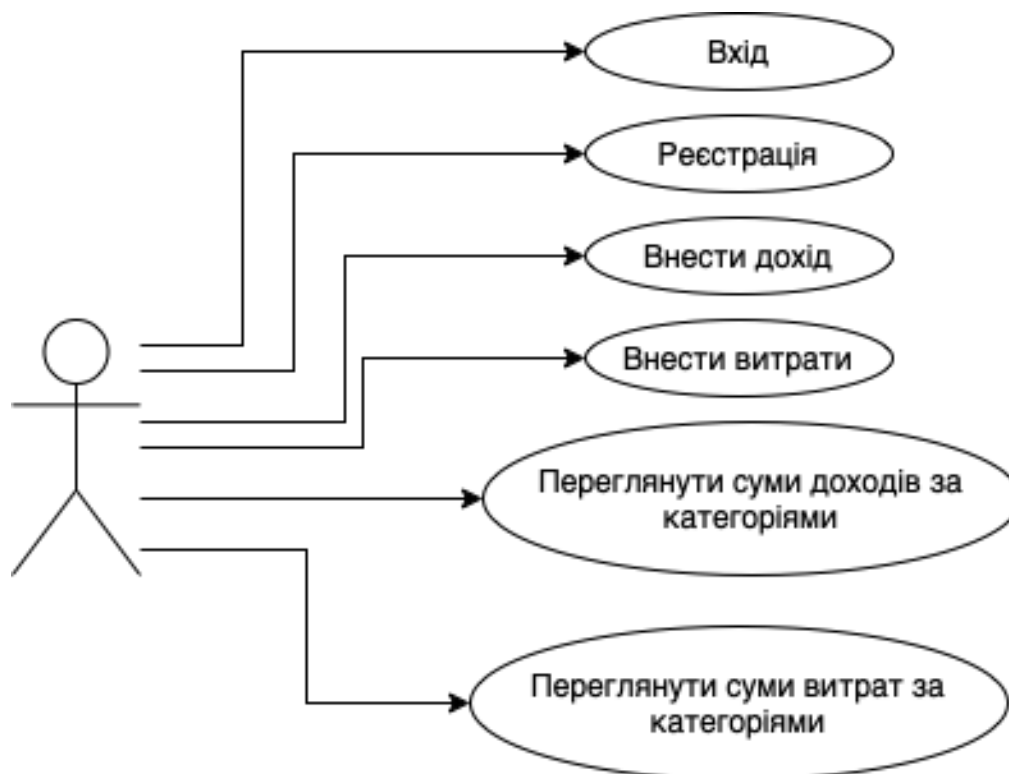


Рисунок 10. Діаграма прецедентів

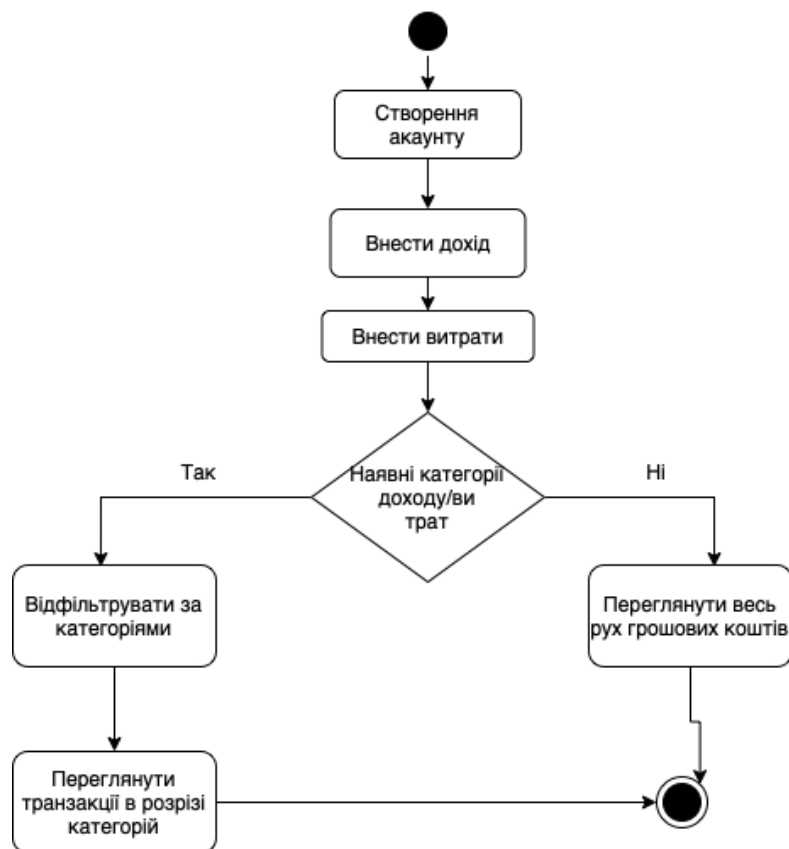


Рисунок 11. Діаграма діяльності

3. Проектування та реалізація

3.1 Інструменти для розробки

3.1.1. Опис середовища розробки Android Studio

Android Studio — інтегрована среда розробки виробництва Google, за допомогою якої розробникам стають доступні інструменти для створення додатків на платформі ОС Android. Android Studio можна встановити на Windows, Mac і Linux. Учетна запис розробника додатків у Google Play App Store коштує \$25. Android Studio створена на базі IntelliJ IDEA.



Рисунок 12. Логотип середовища розробки Android Studio

IDE можна завантажити і користуватися безкоштовно. У неї присутні макети для створення UI, з чого зазвичай починається робота над додатком. У Studio містяться інструменти для розробки рішень для смартфонів і планшетів, а також нові технологічні рішення для Android TV, Android Wear, Android Auto, Glass та додаткові контекстні модулі.

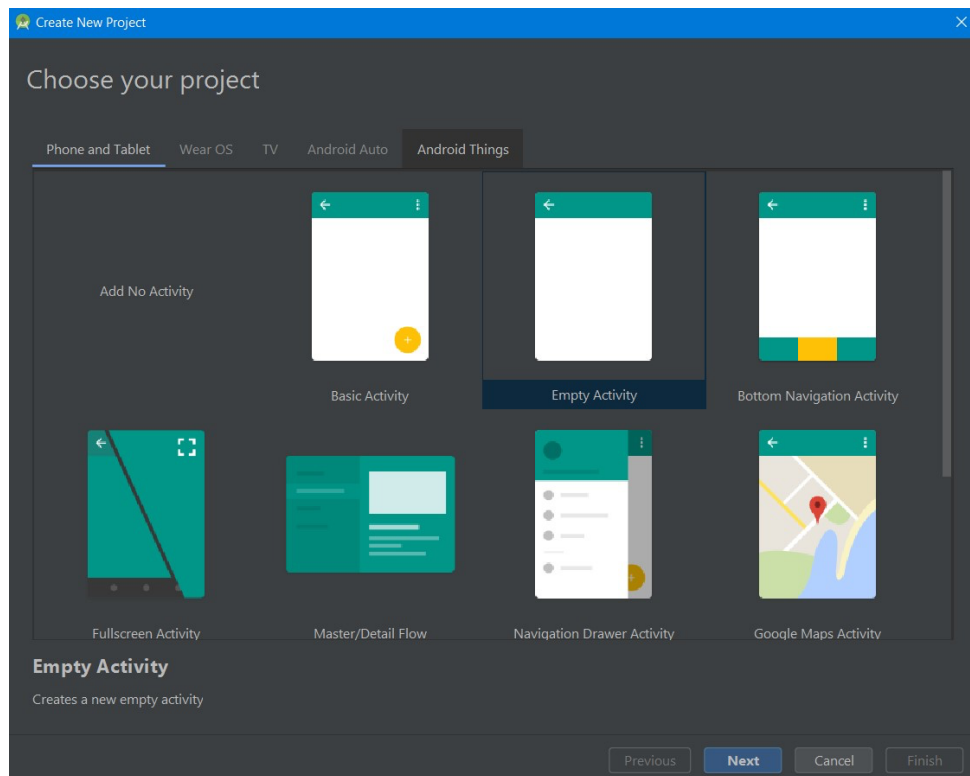


Рисунок 13. Створення нового проекту в Android Studio

Середовище Android Studio призначено як для невеликої команди розробників мобільних додатків (навіть для 1 людини), так і для великих міжнародних організацій з GIT чи іншими подібними системами керування версіями. Опитні розробники зможуть вибрати інструменти, які більше підходять для масштабних проектів. Решення для Android розробляються в Android Studio з використанням Java або C++. В основі робочого процесу Android Studio містить концепцію неперервної інтеграції, що дозволяє відразу ж виявити іменні проблеми. Продовжлива перевірка коду забезпечує можливість ефективної зворотної зв'язку з розробниками. Така опція дозволяє швидше опублікувати версію мобільного додатка в Google Play App Store. Для цього доступна також підтримка інструментів LINT, Pro-Guard і App Signing.

3.1.2 Git

Git — це безкоштовна розподілена система контролю версій з відкритим вихідним кодом, основне призначення якої - швидка та ефективна робота як з малими, так і з великими проектами.

Git простий у освоєнні та має невеликий розмір із блискавичною продуктивністю. Він перевершує інструменти SCM, такі як Subversion, CVS, Perforce і ClearCase, завдяки таким функціям, як дешеве локальне розгалуження, зручні області переходу та кілька робочих процесів.

Функція Git, яка дійсно виділяє його серед майже всіх інших SCM, — це його модель розгалуження.

Git дозволяє і заохочує вас мати кілька локальних гілок, які можуть бути повністю незалежними одна від одної. Створення, об'єднання та видалення цих ліній розробки займають секунди.

Це означає, що ви можете робити такі дії, як:

- **Перемикання контексту.** Створіть гілку, щоб випробувати ідею, зафіксуйте кілька разів, перейдіть до того місця, звідки ви розгалужувалися, застосуйте виправлення, перейдіть до місця, де ви експериментуєте, і об'єднайте його.
- **Кодові рядки на основі ролей.** Майте гілку, яка завжди містить лише те, що йде на виробництво, іншу, в яку ви об'єднуєте роботу для тестування, і кілька менших для щоденної роботи.
- **Робочий процес на основі функцій.** Створюйте нові гілки для кожної нової функції, над якою ви працюєте, щоб ви могли легко перемикатися між ними, а потім видаляйте кожну гілку, коли ця функція буде об'єднана у ваш основний рядок.
- **Одноразове експериментування.** Створіть гілку, щоб поекспериментувати, зрозумійте, що вона не спрацює, і просто видаліть її, покинувши роботу, і ніхто більше її не побачить (навіть якщо ви тим часом натиснули інші гілки).

3.1.3 Firebase Realtime Database

Firestore Realtime Database — це рішення, яке зберігає дані в хмарі та забезпечує простий спосіб синхронізації ваших даних між різними пристроями. Він працює на платформі Google Firebase і є єдиною частиною великої головоломки.

Firestore Realtime Database — це хмарна база даних, яка підтримує платформи iOS, Android, Web, C++ та Unity.

У реальному часі будь-які зміни в даних відразу відображаються на всіх платформах і пристроях протягом мілісекунд. Більшість традиційних баз даних змушують вас працювати з моделлю запиту/відповіді, але база даних реального часу використовує механізми синхронізації даних і підписки замість типових запитів HTTP, що дозволяє створювати більш гнучкі програми в режимі реального часу, легко, з меншими зусиллями і без потреби турбуватися про мережевий код.

Багато програм перестають реагувати, коли ви втрачаєте з'єднання з мережею. База даних реального часу забезпечує чудову підтримку в автономному режимі, оскільки зберігає внутрішній кеш усіх даних, які ви запитували. Коли немає з'єднання з Інтернетом, програма використовує дані з кешу, дозволяючи програмам реагувати. Коли пристрій підключається до Інтернету, база даних реального часу синхронізує локальні зміни даних з віддаленими оновленнями, які відбулися, коли клієнт перебував у автономному режимі, автоматично вирішуючи будь-які конфлікти.

3.2 Архітектура системи

3.2.1. Загальні принципи побудови андроїд додатків

Архітектура андроїд додатку складається з наступних основних частин:

- Каркас програми
- Механізм Binder Inter-Process Communication (IPC) дозволяє платформі програми перетинати межі процесів та викликати код системних служб Android. Це дозволяє високорівневим API-інтерфейсам платформи взаємодіяти із системними службами Android.

- Системні служби. Системні служби — це спеціалізовані модульні компоненти, такі як диспетчер вікон, служба пошуку або диспетчер повідомлень. Функціональні можливості, що надаються API-інтерфейсами платформи додатків, взаємодіють із системними службами для доступу до базового обладнання. Android включає дві групи служб: системні (такі як диспетчер вікон і диспетчер повідомлень) і медіа (служби, задіяні у відтворенні та запису медіафайлів).
- Рівень апаратної абстракції (HAL). HAL визначає стандартний інтерфейс реалізації постачальниками устаткування, що дозволяє Android не залежати від реалізації драйверів нижчого рівня. Використання HAL дозволяє реалізувати функціональність, не торкаючись і змінюючи систему вищого рівня. Реалізації HAL упаковуються в модулі та завантажуються системою Android у потрібний час.

3.2.2. Шаблон проектування Model-View-ViewModel

Для розробки даного програмного забезпечення було обрано Шаблон проектування Model-View-ViewModel. З MVVM процес розробки графічного інтерфейсу для користувачів поділяється на дві частини. Перша — це робота з мовою розмітки або GUI. Друга – розробка бізнес-логіки або логіки бекенда (модель даних). Частина View model у MVVM – це конвертер значень. Це означає, що view model відповідає за конвертування об'єктів даних із моделі в такий вид, щоб з об'єктами було легко працювати. Якщо дивитися з цього боку, то view model це швидше модель, ніж уявлення. Вона контролює більшу частину логіки відображення. Модель представлення може реалізовувати патерн медіатор. Для цього організується доступ до логіки бекенда навколо набору юз-кейсів, що підтримуються представленням.

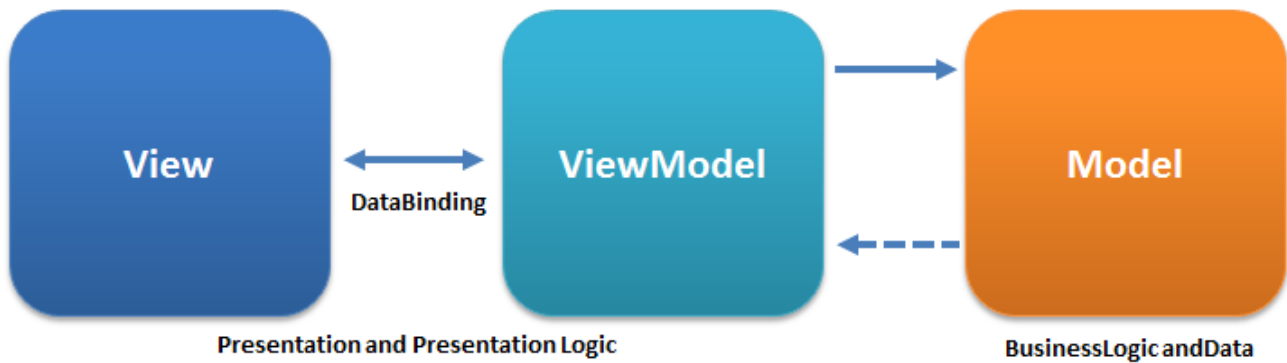


Рисунок 14. Взаємодія між компонентами MVV

Модель в даному шаблоні проектування - це рівень бізнес-даних і він не пов'язаний з жодним особливим графічним уявленням. В Android, згідно з “чистою” архітектурою, модель може містити базу даних, репозиторій та клас бізнес-логіки.

Спочатку створимо моделі сутностей, що використовуватимуться в даному Андрюїд додатку: User, Expense, Income.

```

data class User(
    var email: String = "",
    var name: String? = "",
    var phone: String? = "",
    var income: MutableList<Income>? = null,
    var expenses: MutableList<Expense>? = null,
) : Serializable

```

Рисунок 15. Модель “User”

У моделі User будуть міститися дані про користувача, зокрема його адресу електронної пошти, яку він використовує для входу у свій обліковий запис, ім'я, яке відобразатиметься в системі, номер телефона, списки, в яких містяться дані про всі доходи та витрати, що були збережені в системі.

```

@IgnoreExtraProperties
data class Expense(
    val name: String? = "",
    val value: Double? = null
) : Serializable

```

Рисунок 16. Модель "Expense"

В моделі Expense містяться дані про суму та назву витрат, що введені користувачем.

```

@IgnoreExtraProperties
data class Income(
    val name: String? = "",
    val value: Double? = null
) : Serializable

```

Рисунок 17. Модель "Income"

Клас Income подібний до вищезгаданого класу Expense, алу дана модель відповідає за збереження даних про дохід користувача. В ньому також містяться поля, що відповідають за інформацію про суму надходжень, а також їх джерело.

Наступний компонент даного шаблону проектування - View - містить структурне визначення, що користувачі отримують на екранах. Для нашого випадку у View будуть активності та фрагменти.

Зокрема, окремими активностями в додатку будуть MainActivity - головний екран додатку, NewActivity - екран введення даних про нове надходження або витрати грошових коштів, SettingsActivity - екран за налаштуваннями та екрани входу та реєстрації відповідно - LoginActivity та RegisterActivity.


```

class MainActivity : AppCompatActivity(), View.OnClickListener {

    private lateinit var auth: FirebaseAuth
    private lateinit var database: FirebaseDatabase
    private lateinit var usersRef: DatabaseReference
    private lateinit var user: User

    private lateinit var goToCreateExpense: Button
    private lateinit var goToCreateIncome: Button
    private lateinit var incomeRecyclerView: RecyclerView
    private lateinit var incomeList: List<Income>
    private lateinit var expensesRecyclerView: RecyclerView
    private lateinit var expensesList: List<Expense>

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        //FINDING VIEWS
        goToCreateExpense = findViewById(R.id.goTo_create_expense_button)
        goToCreateIncome = findViewById(R.id.goTo_create_income_button)

        goToCreateExpense.setOnClickListener(this)
        goToCreateIncome.setOnClickListener(this)
    }

    override fun onStart() {
        super.onStart()
        auth = Firebase.auth
    }
}

```

Рисунок 18. Приклад компонента View на прикладі головного екрану додатка

Кожна така активність має власний життєвий цикл - методи та стани, які відпрацьовують в певні моменти її функціонування. При створенні кожної активності першим викликається метод onCreate, єдиним параметром якого виступає змінна savedInstanceState типу Bundle. В даній змінній зберігається стан поточної активності після її минулого знищення, наприклад при повороті екрана девайсу, зміні мови локалі або при примусовому закриття її системою через нестачу ресурсів чи занадто довгий час відповіді. Метод onCreate викликається за першою ініціалізацією активності. В ньому також завдяки методу setContentView вказується, з яким саме XML-файлом буде працювати дана активність, а також відшукуються потрібні для роботи View та відбувається ініціалізація всіх змінних.

Наступним викликається метод onStart(). Він стає активним в той час, коли активність вже готова для показу на екрані.

Наступним є `onResume()` - викликається тоді, коли активність переходить у стан взаємодії з користувачем. Активність продовжує працювати в цьому стані, доки що-небудь не введе фокус від програми або активності (наприклад, вхідний дзвінок). Якщо це станеться, буде викликаний метод `onPause()`.

Також в життєвому циклі є широко використовуваний метод `onPause()` - цей метод використовується для припинення дій, які не повинні відбуватися, поки активність на стадії паузи. Виклик цього методу вказує на те, що користувач залишив програму. Наприклад, вхідний дзвінок може перевести програвач музики в стан паузи. Це повинно заглушити або зупинити музику, що відтворюється.

Коли користувач повертається до програми, викликається метод `onResume()`.

`onStop()`: цей метод викликається, якщо `Activity` більше не видно у додатку. Таке може статися, якщо підвантажено іншу `Activity` і вона займає весь екран пристрою. Коли цей метод викликає, `Activity` повідомляється перейти в стан припинення роботи. У цьому стані система або викликає `onRestart()` для повернення взаємодії з `Activity`, або викликає метод `onDestroy()`, щоб прибрати `Activity`.

`onDestroy()`: цей метод викликається перед тим, як `Activity` буде завершено. Система викликає цей метод, коли користувач завершує `Activity` або якщо система тимчасово прибирає процес, що містить `Activity`, для вивільнення місця. З цим методом, обов'язково звільніть будь-які ресурси, створені вашою `Activity`, інакше ваша програма матиме витік пам'яті.

`onRestart()`: це викликається, якщо `Activity` перезапускається після того, як було зупинено.

Наступним елементом шаблону є `ViewModel`. Цей компонент пов'язує модель та `View`. Відповідає за керування посиланнями даних та можливих конверсій. Тут з'являється біндінг. В `Android` ми не турбуємося про це, тому що можна використовувати клас `AndroidViewModel` або `ViewModel`.

```

class MainActivityViewModel(application: Application) : AndroidViewModel(application) {

    private val repository: UserDataRepository = UserDataRepository(application)

    fun archiveAllSpendNode(name : String) {
        repository.archiveAllSpendNode(name)
    }
}

```

Рисунок 19. Приклад компонента *ViewModel* на прикладі *ViewModel* головного екрану додатка

3.2.3. Зберігання даних користувача за допомогою **Firebase Realtime Database** та автентифікація за допомогою **Firebase Auth**

Firebase — це платформа для розробки програмного забезпечення, розроблена Google. У 2014 році його купила Google. Зараз Firebase має 18 програмних продуктів, які використовують 1,5 мільйона програмних додатків.

Найвідоміші програми – Firebase Analytics, Firebase Cloud Messaging, Firebase Auth, Firebase Realtime, Firebase Hosting. Крім того, створено інфраструктуру для тестування програмних додатків для мобільних операційних систем Android та iOS. Усі необхідні матеріали про результати тестування доступні в зручній формі. Є можливість автоматичного тестування, якщо розробник програмного забезпечення не надав тестовий код для розробленої системи.

Firebase Realtime — це послуга, що надається базою даних, заснована на хмарній технології. Дані зберігаються у форматі JSON і синхронізуються в режимі реального часу. Завдяки Firebase SDK для мобільної розробки розробники автоматично оновлюються новими даними. Дані залишаються доступними, коли ви переходите в режим офлайн.

Основні особливості Firebase:

- Синхронізація даних у режимі реального часу;
- швидкий доступ до документів і колекцій, що зберігаються в базі даних; -
- Простота використання, яка є пріоритетом для всіх продуктів Firebase;

- Використання колекцій і документів для структурування та обробки даних.

При такому підході продуктивність запиту не залежить від обсягу даних, а результат роботи з базою даних на 100 і 100 мільйонів документів однаково ефективний. – Розгортання без розробки сервера. Cloud Firestore Client SDK займається функцією аутентифікації користувача. Бекенд надає набір утиліт, які контролюють доступ до даних. Повна інтеграція з платформою Firebase.

Firebase Realtime Database - база даних, створена для зберігання та синхронізації даних у реальному часі. Це документоорієнтована база даних, головною перевагою якої є універсальність роботи з різними пристроями і відсутність необхідності готувати таблиці дані в цей формат.

Firebase – це платформа для розробки мобільних і веб-додатків, що дає змогу швидко створювати високоякісні додатки, залучати лояльних користувачів і збільшувати прибутки. До платформи включено комплекс інтегрованих функцій, які можна поєднувати й суміщати, зокрема мобільний сервер, засоби аналітики, а також інструменти вдосконалення й монетизації для максимальної успішності додатків.

Для початку в консолі розробника firebase, що знаходиться за посиланням <https://console.firebase.google.com/u/1> потрібно створити новий проект. Для цього потрібно казати ім'я проекту та погодитися з умовами використання firebase.

Після створення потрібно перейти в налаштування проекту та натиснути на вкладку «сервісні акаунти» та згенерувати закритий ключ, який потрібно помістити в корінь нашого проекту. Щоб можна було використовувати всі можливості firebase, потрібно підключити в gradle на рівні окремого модуля потрібні бібліотеки.

Для зчитування даних з файлу призначений клас `FileInputStream`, який є спадкоємцем класу `InputStream` і тому реалізує всі його методи. Для створення об'єкта `FileInputStream` ми можемо використовувати конструктор, який приймає вхідним параметром шлях до файлу. Тому ми вказуємо шлях до файлу з ключом.

В налаштуванні для підключення до бази вказуємо ключ та шлях до бази, який можна знайти в консолі розробника на вкладці Database. Після ініціалізації налаштувань, можна використовувати базу через `DatabaseReference`. Функція `child()`

приймає в себе значення ключа, якому будуть присвоєні якісь значення, але якщо ключа не існує, то firebase його створить.

Щоб зчитати данні з бази використовується одна з трьох функцій: `addValueEventListener`, `addListenerForSingleValueEvent`, `addChildEventListener`.

Ці функції приймає вхідним параметром інтерфейс, який реалізовує два методи: `onDataChange` та `onCancelled`. Перший метод заповниться об'єктами із бази даних в `dataSnapshot`, який ми можемо використовувати. Отримавши всі дочірні об'єкти з `dataSnapshot`, за допомогою циклу ми можемо вивести їх ключі та значення. Якщо дані більше не потрібні, їх можна видалити функцією `removeValueAsync()`.

```
dependencies {
    implementation fileTree(dir: "libs", include: ["*.jar"])
    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
    implementation 'androidx.core:core-ktx:1.3.2'
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation platform('com.google.firebase:firebase-bom:26.1.0')
    implementation 'com.google.firebase:firebase-auth-ktx'
    implementation 'com.google.firebase:firebase-database-ktx'
    implementation 'androidx.recyclerview:recyclerview:1.1.0'
    implementation 'androidx.cardview:cardview:1.0.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
}
```

Рисунок 19. Імплементування бібліотек для підключення Firebase сервісів

Після імплементування бібліотек потрібно створити інстанс Firebase автентифікації та Firebase БД. Для цього в класі `MainActivity` ініціалізуємо змінні типів `DatabaseReference`, `FirebaseAuth`, `FirebaseDatabase`.

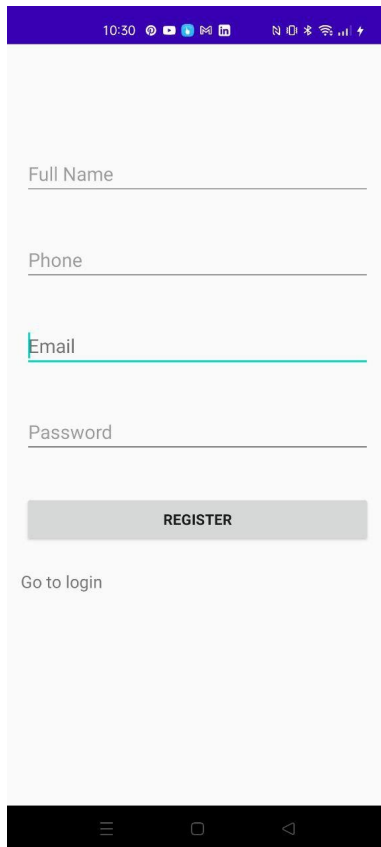
```
// FIREBASE
auth = FirebaseAuth
database = FirebaseDatabase
usersRef = database.reference.child( pathString: "users")
```

Рисунок 20. Імплементування бібліотек для підключення Firebase сервісів

Після додавання в проект всіх необхідний налаштувань БД готова до роботи.

Практичний результат дослідження

В результаті проведеної роботи було створено додаток, який допомагає користувачу у моніторингу руху власних грошових ресурсів.



The image shows a mobile application registration screen. At the top, there is a status bar with the time 10:30 and various system icons. Below the status bar, the registration form consists of four input fields: "Full Name", "Phone", "Email", and "Password". Each field has a horizontal line indicating the input area. Below the "Email" field, there is a "REGISTER" button. At the bottom of the form, there is a link that says "Go to login". The screen is framed by a dark purple header and a black footer with navigation icons.

Рисунок 21. Екранна форма реєстрації користувача

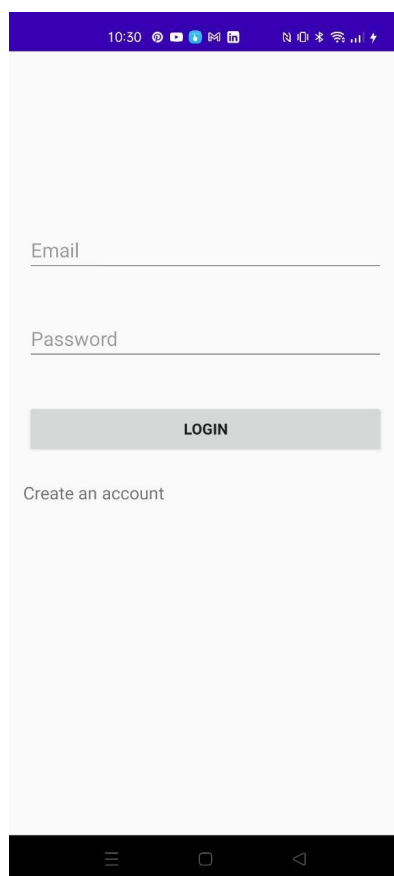


Рисунок 22. Екранна форма входу користувача

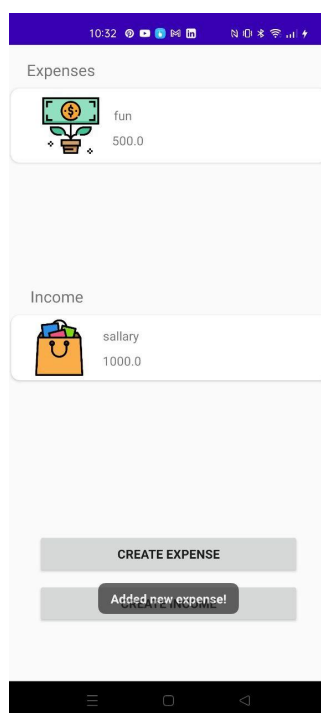
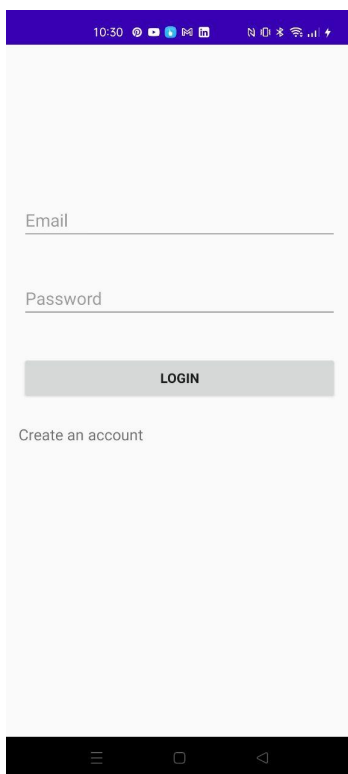


Рисунок 23. Екранна форма головного екрана



10:30

Email

Password

LOGIN

Create an account

Рисунок 24. Екранна форма внесення інформації про грошову операцію

Висновки

Дана робота була спрямована на проектування програмного забезпечення для фінансового моніторингу.

1. Проведено аналіз актуальності розробленої системи завдяки дослідженням аналогів.
2. Був обраний шаблон проектування MVVM. В якості мови програмування було обрано Kotlin, AndroidStudio.
3. Було розроблено програмне забезпечення для фінансового моніторингу мовою Kotlin.
4. Розроблено програмні модулі для роботи з базою даних Firebase Realtime Database.
5. Було досліджено автентифікація на основі Google Auth.
6. Результат дослідження бакалаврської роботи апробовані на всеукраїнських науково-технічній конференції: «IV Міжнародна студентська наукова конференція «НАУКА СЬОГОДЕННЯ: ВІД ДОСЛІДЖЕНЬ ДО СТРАТЕГІЧНИХ РІШЕНЬ»» з темою «ВАЖЛИВІСТЬ ВИКОРИСТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ “ФІНАНСОВИЙ МОНІТОРИНГ” ДЛЯ СУЧАСНОГО КОРИСТУВАЧА ТА АНАЛІЗ НАЙБІЛЬШ ПОПУЛЯРНИХ ДОДАТКІВ».

Список використаної літератури

1. Эккель Б. Философия Java. 4-е полное изд. – СПб.: Питер, 2018 – 1168 с.: ил. – (Серия «Классика computer science»)
2. Efficient Android Threading: Asynchronous Processing Techniques for Android Applications. By Anders Goransson. O'Reilly Media; 1 edition June 13, 2014; 280 pages;
3. Android Programming: The Big Nerd Ranch Guide. By Bill Phillips, Chris Stewart & Kristin Marsicano. Big Nerd Ranch Guides; 3 edition February 9, 2017; 624 pages;
4. Mastering Android Development with Kotlin: Deep dive into the world of Android to create robust applications with Kotlin By Milos Vasic. Packt Publishing — ebooks Account, November 8, 2017; 378 pages;
5. Дэвид Гриффитс, Дон Гриффитс «Head First. Программирование для Android» Питер, 2018 год, 912 стр, О’Reilly, 2-е изд.,
6. Java 8. Полное руководство. 9-е издание 89480К, 1377 с.
7. Android Studio [электронный ресурс] // Режим доступа: https://en.wikipedia.org/wiki/Android_Studio
8. Gartner [электронный ресурс] // Режим доступа: <https://www.gartner.com/en/newsroom/press-releases/2018-02-22-gartner-says-world-wide-sales-of-smartphones-recorded-first-ever-decline-during-the-fourth-quarter-of-2017>
9. Android [электронный ресурс] Режим доступа: <https://uk.wikipedia.org/wiki/Android>
10. Firebase Cookbook: Over 70 recipes to help you create real-time web and mobile applications with Firebase (English Edition).
11. Поняття про додатки для Android і їх компоненти [электронный ресурс] // Режим доступа: <https://fans-android.com/ponyatiya-o-prilozheniyax-dlya-android-i-ix-komponenty/>
12. Firebase [электронный ресурс] // Режим доступа: <https://uk.wikipedia.org/wiki/Firebase>

Додаток



Державний університет телекомунікацій
 Навчально-науковий інститут ІНФОРМАЦІЙНИХ
 ТЕХНОЛОГІЙ
 Кафедра інженерії програмного забезпечення



Розробка програмного забезпечення «Фінансовий моніторинг» мовою Kotlin

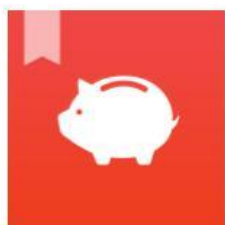
Виконала студентка 4 курсу
 Групи ПД-43
 Шibaєва Тамара Дмитрівна
 Керівник роботи
 доц. кафедри ІПЗ Поперешняк
 Світлана Володимирівна

Київ – 2022

Наявні аналоги на ринку



Finance PM



Money Manager



Bluecoins



Alzex finance

Огляд існуючих аналогів (переваги та недоліки)

| Назва додатку | Переваги | Недоліки |
|---------------|---|--|
| Finance PM | Наявне планування та нагадування | Немає автоматичного оновлення курсу валют, розпізнавання смс, інтеграції з банками, лише один рівень категорій |
| Money Manager | Наявні групи рахунків, бекап в гугл диск, sso | Немає розпізнавання смс, інтеграції з банками, лише один рівень категорій |
| Bluecoins | В преміум акаунті доступні розпізнавання смс, прогноз погоди, бекап в гугл диск | Лише один рівень категорій, більшість функцій доступні лише с преміум підпискою, лише один рівень категорій |
| Alzex finance | В платній версії є сканер чеків (тестування даної функції до покупки преміум акаунта неможливе) | Немає імпорта смс, автоматичного розрахунку відсотків, лише один рівень категорій |

3

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета - розробка та підтримка протягом всього життєвого циклу мобільного програмного забезпечення для здійснення фінансового менеджменту.

Об'єкт дослідження - розробка мобільного додатку для фінансового менеджменту.

Предмет дослідження - мобільне програмне забезпечення для фінансового менеджменту.

4

Технічне завдання

1. Реалізувати базу даних, в якій зберігатимуться дані про суми грошових надходжень та витрат користувача
2. Реалізувати можливість входу та реєстрації користувача за допомогою облікового запису Гугл
3. Реалізувати можливість групувати суми надходжень та витрат за категоріями
4. Реалізувати можливість графічного відображення місткості категорій за допомогою діаграм

Програмні засоби реалізації



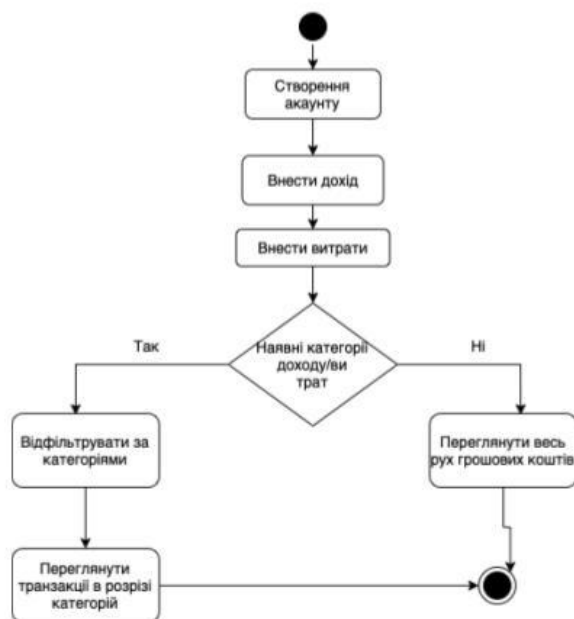
6

Діаграма прецедентів



7

Діаграма діяльності



8

Екранні форми



Екран реєстрації



Екран входу



Екран внесення доходу



Екран головної сторінки

9

Висновки

1. Проаналізовано наявні на ринку програми, що призначені для фінансового моніторингу, виявлено їх недоліки та можливості для їх усунення
2. Спроектовано програмне забезпечення на основі проведеного аналізу
3. Розроблено додаток із врахуванням вищезгаданих вимог
4. Протестовано додаток

10

ДЯКУЮ ЗА УВАГУ!

11