

DOI: 10.36910/6775-2524-0560-2020-39-27

УДК: 004.4

Мельник Катерина Вікторівна, к.т.н., доцент

<https://orcid.org/0000-0002-9991-582X>

Костючко Сергій Миколайович, к.т.н., доцент

<https://orcid.org/0000-0002-1262-6268>

Мельник Дмитро Сергійович, студент

Луцький національний технічний університет

ОПТИМІЗАЦІЯ ВЕДЕННЯ ТА АНАЛІТИКИ ФІНАНСІВ ЗА ДОПОМОГОЮ МОБІЛЬНОГО ДОДАТКУ НА ОСНОВІ ОС ANDROID

Мельник К.В., Костючко С.М., Мельник Д.С. Оптимізація ведення та аналітики фінансів за допомогою мобільного додатку на основі ОС Android. Фінансова грамотність один з найнеобхідніших навиків для сучасної людини. Зростання матеріальних статків у значній мірі залежить від того, на скільки розумно людина використовує ресурси та розпоряджається грошима. Ведення та аналітика доходів і витрат перший крок до успіху у керуванні фінансами. У даній статті розглядається розробка додатку в IDE Android Studio для ведення та аналітики фінансів.

Ключові слова: смартфон, OS Android, фінансовий облік, фінансова аналітика.

Мельник К.В., Костючко С.Н., Мельник Д.С. Оптимизация учета и аналитики финансов с помощью мобильного приложения на основе операционной системы Android. Финансовая грамотность один из самых необходимых навыков в жизни современного человека. Увеличение материального состояния зависит от того, насколько разумно человек использует ресурсы и распоряжается деньгами. В данной статье рассматривается разработка приложения в IDE Android Studio для учета и аналитики финансов.

Ключевые слова: смартфон, OS Android, финансовый учет, финансовая аналитика.

Melnyk K.V., Kostiuchko S.M., Melnyk D.S. Optimization of accounting and financial analytics using a mobile application based on the Android operating system. Financial literacy is one of the most necessary skills in the life of a modern person. The increase in material condition depends on how rationally a person uses resources and manages money. This article discusses application development in the Android Studio IDE for accounting and financial analytics.

Keywords: smartphone, OS Android, financial accounting, financial analytic.

1. Постановка проблеми. Фінансова грамотність один з найнеобхідніших навиків для сучасної людини. Зростання матеріальних статків у значній мірі залежить від того, на скільки розумно людина використовує ресурси та розпоряджається грошима. Ведення та аналітика доходів і витрат перший крок до успіху у керуванні фінансами.[1]

Найзручнішим засобом для обліку грошей є мобільний додаток. Оскільки виходячи з дому людина завжди бере з собою смартфон, а тому може моментально записати будь-які свої фінансові операції. До того ж такий додаток може проводити аналіз даних, що також необхідно для фінансово грамотної людини. Сучасні гаджети дозволяють робити це в автоматичному режимі, позбавляючи необхідності механічно віднімати та додавати суми. Додатки для контролю фінансів пропонують лише вводити дані про свої доходи та витрати і робити висновки про свою економічну ситуацію на основі доступних та зрозумілих графіків.[2]

Отже, мобільний додаток найзручніший та найкращий спосіб для ведення та аналітики фінансів.

2. Аналіз останніх досліджень і публікацій.

Аналізуючи Play Market можна зробити висновок, що подібні додатки мають більше 100 тисяч завантажень, що підтверджує те, що все більше людей користується мобільними додатками для обліку фінансів, тому і виникає необхідність в зручних інструментах для полегшення процесу запису та аналізу. Існують як безкоштовні так і платні програми. Також є додатки з обмеженим функціоналом, щоб отримати повний доступ до всіх можливостей програми необхідно оплатити підписку. Найчастіше якраз такі найбільш необхідний функціонал є платним.

Топ-п'ятірку бестселерів у категорії «Фінанси» Google Play визначив так: CoinKeeper, Wallet, Monefy, «Дзен-мані» та «Домашня бухгалтерія». При цьому всі додатки в топі виявилися з безкоштовним основним функціоналом.[3]

Порівнюючи ці додатки можна зробити такий висновок – інтерфейс програм складний та перевантажений. Процес запису фінансової операції займає близько 5 кліків, що погіршує досвід користування додатком. Також не у всіх додатках є можливість експорту/імпорту даних. Аналітика пропонує стандартний набір інструментів та не завжди надає достатньо інформації для аналізу.

3. Виділення невирішених раніше частин загальної проблеми. Отже, актуальним завданням є розробка мобільного додатку для обліку фінансів та аналітики даних. З простим та зручним інтерфейсом для швидкого запису фінансових операцій, інструментами, які допоможуть аналізувати, експортувати та імпортувати дані. Та привчити користувача до запису своїх витрат та доходів.

4. Формулювання мети дослідження. Мета дослідження полягає в проведенні аналізу існуючих аналогів для ведення та аналітики грошових потоків та розробка власного додатку обліку фінансів з включенням платних функцій.

5. Виклад основного матеріалу дослідження. Створюючи проект розробник може обрати найновішу версію Android, використовувати усі найновіші функції. Але у даному випадку у користувачів, які мають нижчі версії Android можуть бути проблеми у використанні додатку, або взагалі неможливість його використання у зв'язку з не підтримкою певних функцій. Даний вибір є не рентабельним з комерційної точки зору, оскільки зазвичай найновіші версії Android встановлено на 1-10% смартфонів. До прикладу Android 7.0 (Nougat), згідно з інформацією наданою в Android Studio встановлено 8.1% девайсів, старішу версію Android 6.0 (Marshmallow) - 39.3%. В той же час обравши нижчу версію ОС розробник не має можливості використовувати всі переваги нових версій, але підтримувати нормальну роботу додатку буде більша кількість смартфонів.

Даний проект створений для Android 5.0, оскільки 71% девайсів підтримуються з цієї версії Android, незважаючи на те, що дана версія не є найновішою, але ще доволі популярна у використанні.

Середовищем розробки для цього було обрано Android Studio, на мові Kotlin, використовуючи MVP (Model-View-Presenter) шаблон проектування, для чистоти коду та розділення відповідальності.

Android Studio є офіційним середовищем розробки від компанії Google, з'явилася ще в травні 2013. Це заміна інструментів розробки для Android Eclipse (ADT), як основного середовища розробки для власних розробників додатків Android. Пакет інсталяції цього середовища можна абсолютно безкоштовно завантажити з сайту <http://developer.android.com/sdk/index.html>.

В MVP шаблоні View компонент має містити посилання на Presenter, для цього застосуємо ще один паттерн Dependency Injection (DI), для цього використаємо бібліотеку Dagger 2, яка вирішить проблему впровадження залежностей. Передавати залежності клієнту замість дозволити йому створювати сервіс є фундаментальною вимогою до MVP шаблону проектування. Для уникнення проблем з життєвим циклом, шаблонним кодом та простішої реалізації шаблону MVP використаємо бібліотеку Моху.

Якщо додаток неможливо зробити цілковито синхронним, то наявність єдиного асинхронного ресурсу повністю позбавляє всіх переваг традиційного імперативного стилю програмування. Якби написаний код виконувався синхронно та в одному потоці, то він би виконувався послідовно та без будь-яких проблем, проте коли є необхідність в асинхронності, наприклад при виконанні запиту до бази даних.

Всі данні про фінансові операції зберігаються локально, тобто в пам'яті пристрою, тому потрібно створити БД (базу даних) та організувати роботу по збереженню та читанню даних. Для цього використаємо Room.

Room – це високорівневий інтерфейс для низькорівневих прив'язок SQLite, вистроєних в Android. Даний інтерфейс виконує більшу частину своєї роботи під час компіляції, створюючи API-інтерфейс надбудову над SQLite API. Це спрощує роботу програміста та виключає необхідність використання Cursor та ContentResolver. В даному додатку використовується три таблиці бази даних:

- Доходи.
- Витрати.
- Категорії.

Ось так виглядає інтерфейс таблиці витрат (рис. 1). В ньому описані методи для збереження та читання даних таблиці.

```
@Dao
interface CostsDao {

    @Query("SELECT * FROM ${DatabaseConfiguration.Costs.TABLE_NAME}")
    fun getAll(): Single<List<CostsEntity>>

    @Query("SELECT * FROM ${DatabaseConfiguration.Costs.TABLE_NAME} WHERE ${DatabaseConfiguration.Costs.Columns.DATE} BETWEEN :from AND :to")
    fun getBetweenDates(from: Date, to: Date): Single<List<CostsEntity>>

    @Query("SELECT * FROM ${DatabaseConfiguration.Costs.TABLE_NAME} WHERE ${DatabaseConfiguration.Costs.Columns.CATEGORY_ID} = :categoryId")
    fun getByCategory(categoryId: Int): Single<List<CostsEntity>>

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    fun insert(costsEntity: CostsEntity): Long

    @Query("DELETE FROM ${DatabaseConfiguration.Costs.TABLE_NAME}")
    fun clear()
}
```

Рис. 1 Інтерфейс таблиці витрат

На головному екрані додатку відображається навігаційне меню та круговий графік з місячним фінансовим оборот у відсотковому співвідношенні (рис. 2). На задньому фоні розміщено view елемент з плаваючими хвилями, це надає користувачу відчуття спокою та врівноваженості, згідно з

дослідженнями, це сприяє кращому враженню від користування додатком, збільшує ймовірність повторного користування та зменшує ймовірність видалення.

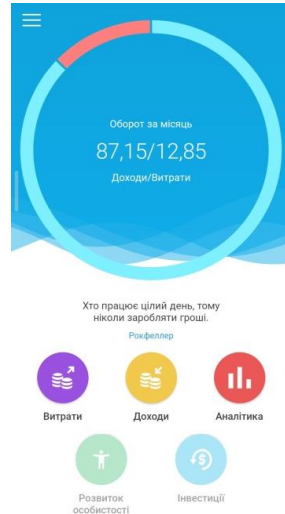


Рис. 2. Головний екран

Користувачу доступний стандартний набір функцій. Для відкриття додаткового функціоналу необхідно оплатити підписку. Доступні підписки на місяць та рік. Механізм покупки та перевірки активних підписок реалізовано за допомогою стандартного сервісу від Google - In-app Billing Subscriptions. Даний інструмент дозволяє автоматизувати списання коштів з рахунку користувача при наявності активної підписки та дозволяє перевіряти статус підписки.[4]

До платних функцій додатку входять:

- Створення кастомних категорій.
- Фільтрація по датах.
- Імпорт/експорт даних для відновлення.
- Експорт даних у файл Excel.

Однією з головних функцій додатку є аналітика. В аналітиці відображаються графіки доходів і витрат (рис. 3) та графік різниці доходів/витрат. Графік різниці у фінансово грамотних користувачів повинен бути вище 0.

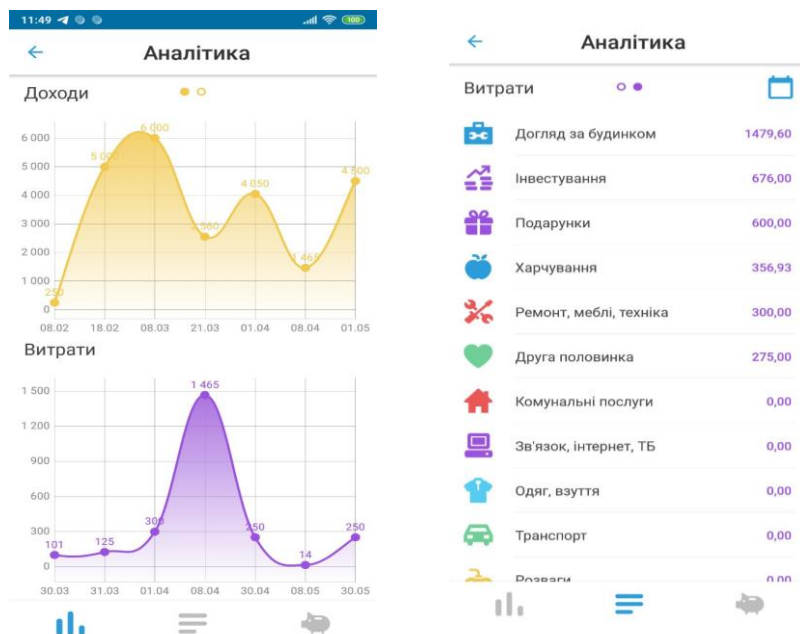


Рис. 3 Графіки доходів і витрат та список витрат по категоріям

Особливою функцією в аналітиці є розрахунок вартості години роботи, яка розраховується з даних про витрати за попередні місяці. І показує необхідний мінімальний дохід за годину для покриття витрат. Це дозволяє оцінити та порівняти свій рівень доходу.

Додаток дозволяє розрахувати дату фінансової незалежності, тобто коли пасивні доходу приносять прибутку більше витрат. Для цього є спеціальна категорія Скарбничка, яку користувач поповнює раз в місяць. Для розрахунку дати фінансової незалежності необхідно ввести відсоток доходу, який користувач отримує з грошей накопичених в скарбничці. Після чого відбуваються певні обчислення і коли дохід з відсотків перевищує місячні витрати – користувач вважається фінансово незалежним.

Вище перераховані інструменти аналітики є унікальними і не мають аналогів у інших додатках.

Також в аналітиці відображаються дані витрат та доходів по категоріям з можливістю сортувати дані по даті (рис. 3). Даний інструмент присутній не у всіх додатках, але за допомогою нього можна з легкістю проаналізувати на що витрачається найбільше коштів. Це дає можливість користувачеві оптимізувати свої витрати у майбутньому. Дрібні та необдумані покупки часто залишаються непоміченими для людини, але займають значну частину витрат, а постійний облік фінансів та дані аналітичні інструменти дозволять контролювати свої фінанси та зменшити витрати в майбутньому.

Для того, щоб користувач не забував записувати свої фінансові операції у додатку є функція сповіщень, увімкнути та вискнути які можна в налаштуваннях. Сповіщення приходять два рази на день з нагадуванням про необхідність записати свої доходи/витрати. Оскільки додаток працює без взаємодії з сервером, тому і сповіщення налаштовуються локально. Реалізовано це з допомогою AlarmManager, в якому задається таймер до наступного пробудження та задається PendingIntent. PendingIntent – це свого роду контейнер, в який можна запакувати дані. При пробудженні додатку у фоні від AlarmManager, відкривається цей контейнер з даними, після чого запускається сервіс, який вказали у Intent, а об'єкт класу Intent записали у PendingIntent. Приклад реалізацій планувальника сповіщень (рис. 4).

```
private fun scheduleNotification() {
    val notificationIntent = Intent( packageContext this, NotificationPublisher::class.java)

    val pendingIntent :PendingIntent! = PendingIntent.getBroadcast( context this, requestCode 0, notificationIntent, PendingIntent.FLAG_UPDATE_CURRENT)

    val alarmManager :AlarmManager = getSystemService(Context.ALARM_SERVICE) as AlarmManager
    alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, System.currentTimeMillis(), intervalMillis: 60 * 60 * 1000, pendingIntent)
}
```

Рис. 4 Метод для планування сповіщень

Купуючи підписку користувач отримує можливість імпортувати та експортувати дані для збереження, а також зберігати дані у файлі Excel. Ця функція також є унікальною, та немає аналогів у інших додатках. Вона буде корисною для просунутих користувачів, у програмі Excel за допомогою різних інструментів та формул можна проводити будь-який аналіз, проводити вибірки, будувати графіки та діаграми, а також просто переглянути таблицю з даними, оскільки не зважаючи на зручність та популярність смартфонів, переглядати великі таблиці даних зручніше на великому екрані. Завдяки даній функції користувач отримує таку можливість. Для реалізації створення *.xlsx файлів використовується Apache POI API. Дане API має потужний функціонал, та дозволяє створювати листи заповнювати клітинки, стилізувати їх, записувати формули, а також читати ці файли. Це свого роду конструктор *.xlsx файлів. Ось так виглядає заповнення даних у клітинки таблиці:

```
for(int i=0; i<yourArrayList.size(); i++){
    Row row = sheet.createRow(i);
    row.createCell(CELL_INDEX_0).setCellValue(VALUE_YOU_WANT_TO_KEEP_ON_1ST_COLUMN);
    row.createCell(CELL_INDEX_1).setCellValue(VALUE_YOU_WANT_TO_KEEP_ON_2ND_COLUMN);
}[5]
```

Створюючи резервну копію даних, створюється файл формату *.json. Це простий, зручний та популярний формат передачі масивів інформації. Конвертація даних у json формат не займає багато часу та зусиль. Для цього необхідно створити об'єкт класу Gson, після чого викликати метод цього класу toJson(), параметром цього методу може бути об'єкт будь-якого типу, що є доволі зручним та не вимагає додаткових конвертацій.

```
val gson = GsonBuilder().setPrettyPrinting().create()
```

```
jsonSubCategoryExportList = gson.toJson(subCategoryList)
```

6. Висновки. Мною проаналізовано аналогічні додатки на ринку та виявлено переваги та недоліки. Розроблено та реалізовано архітектуру та концепт роботи і дизайн додатку. Створено унікальні аналітичні інструменти та експорт даних у файл *.xlsx.

Перспективи подальших досліджень. В перспективі подальших досліджень є розробка взаємодії з сервером для кращої синхронізації даних між пристроями. Створення онлайн сервісу для аналізу та редагування даних. Підключення взаємодії з банкінгом для автоматичного збору інформації про фінансові операції та створення iOS версії.

Список бібліографічного опису

1. Ареф'єва І. Фінансова грамотність – мистецтво багатіти [Електронний ресурс] / Ірина Ареф'єва. – 30.04.2020. – Режим доступу до ресурсу: <https://financer.com/ua/finansova-gramotnist/>.
2. Гроші люблять облік: топ-5 додатків для контролю фінансів [Електронний ресурс] / Розробник Expensify, – 2019. – Режим доступу до ресурсу: <https://cutt.ly/rybPu7u>.
3. Гордієнко Т. Як завести здорові фінансові звички? П'ять безкоштовних додатків для обліку грошей [Електронний ресурс] / Тетяна Гордієнко // 2018 – Режим доступу до ресурсу: <https://cutt.ly/lybAcOH>.
4. In-app Billing Subscriptions со стороны сервера [Електронний ресурс] // 2013 – Режим доступу до ресурсу: <https://habr.com/ru/post/160911/>.
5. Shaki S. H. Create an excel file programmatically in Android [Електронний ресурс] / Shahadat Hossain Shaki // Mar 21, 2019 – Режим доступу до ресурсу: <https://cutt.ly/rybAYut>.

References

1. Arefieva I. Financial literacy - the art of getting rich [Electronic resource] / Irina Arefieva. – 30.04.2020 - Mode of access to the resource: <https://financer.com/ua/finansova-gramotnist/>.
2. Money loves accounting: top 5 applications for financial control [Electronic resource] / Developer Expensify, - 2019. - Resource access mode: <https://cutt.ly/rybPu7u>.
3. Gordienko T. How to start healthy financial habits? Five free applications for accounting money [Electronic resource] / Tatiana Gordienko // 2018 - Resource access mode: <https://cutt.ly/lybAcOH>.
4. In-app Billing Subscriptions from the server [Electronic resource] // 2013 - Resource access mode: <https://habr.com/ru/post/160911/>.
5. Shaki S. H. Create an excel file programmatically in Android [Electronic resource] / Shahadat Hossain Shaki // Mar 21, 2019 - Resource access mode: <https://cutt.ly/rybAYut>.