

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ
ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА ПРИКЛАДНОЇ ІНФОРМАТИКИ**

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

_____ Гамаюн В.П.
(підпис) (ПІБ)

“ _____ ” _____ 2021р.

**ДИПЛОМНИЙ ПРОЕКТ
(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ “БАКАЛАВР”

Тема: Android-додаток для ведення особистих фінансів

Виконавець: _____ Трухан Денис Сергійович
(підпис) (ПІБ)

Керівник: _____ Ходаков Данііл Вікторович
(підпис) (ПІБ)

Нормоконтролер: _____ Боровик Володимир Миколайович
(підпис) (ПІБ)

Київ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра прикладної інформатики

Освітній ступень Бакалавр

Спеціальність 122 "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Гамаюн В.П.
(підпис) (ПІБ)
" ____ " _____ 2021р.

ЗАВДАННЯ

на виконання дипломного проекту

Трухана Дениса Сергійовича

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема дипломного проекту: Android-додаток для ведення особистих фінансів
затверджена додатком до наказу ректора від " ____ " _____ 2021 р., № ____ /ст _____
2. Термін виконання проекту: з " ____ " _____ 2021 р. по " ____ " _____ 2021 р.
3. Вихідні дані до проекту: 1. Зробити аналіз існуючих засобів для керування власним бюджетом 2. Розробити інтерфейс і структуру мобільного додатку "Менеджер фінансів". 3. Розробити програмну реалізацію мобільного додатку "Менеджер фінансів"
4. Зміст пояснювальної записки: 1. Аналіз існуючих засобів для керування власним бюджетом. 2. Проектування та вимоги мобільного додатку "Менеджер фінансів". 3. Розробка функціональної частини мобільного додатку "Менеджер фінансів".
5. Перелік обов'язкового графічного (ілюстративного) матеріалу: 1. Дерево ресурсів ОС "Android". 2. Загальна структурна схема додатку. 3. Створення макету додатку. 4. Кругова діаграма "PieChart" графіку витрат.
5. Структура бази даних додатку. 6. Верстка фрагменту «Рахунки».
7. Результат роботи готового додатку.

ВСТУП

Бюджет і відповідно процес бюджетування – більш вузькі поняття, ніж фінансовий план та фінансове планування. План включає весь спектр дій, спрямованих на досягнення сформульованих цілей, які можуть бути описані як з допомогою формалізованих кількісних оцінок, так і шляхом опису якісних показників. Бюджет є робочим інструментом фінансового планування, який полегшує реалізацію намічених планів і робить більш обґрунтованим їх коригування.

Тобто, для складання бюджету потрібно визначити, по-перше, час, наприклад, помісячно. Далі, базуючись на досвіді попереднього місяця скласти список і підрахувати очікувані витрати на поточний місяць. Порівняти отриману суму із власним сукупним місячним доходом. Це порівняння дозволить зрозуміти, чи можливо забезпечити всі свої потреби з таким доходом, також покаже, в чому можна обмежитись і т.д.

Саме через цю причину бюджетування є дуже популярною процедурою серед людей усього світу. Багато людей проводить його в блокнотах чи спеціально відведених поліграфічних журналах. Але на сьогоднішній день технології не стоять на місці, тому існують і інші методи складання власного бюджету.

Для великої кількості людей є зручним використання програми *Excel* для складання власного бюджету та контролю за власними фінансами. В цьому програмному забезпеченні є дуже велика кількість потрібних засобів для бюджетування, але *Excel* також і перенасичений ними, тому і важкий в користуванні. Через цю проблему були створені спеціалізовані програмні засоби для контролю власних доходів та витрат, для складання бюджету. У більшості з них також перенасичений функціонал.

Актуальність теми полягає в тому, щоб автоматизувати операцію ведення обліку фінансів за допомогою мобільного телефона чи планшета. Також в теперішній час з'явилася тенденція до спрощення графічного інтерфейсу користувача, тобто створення меншої кількості засобів керування (кнопок, чек-

боксів і. т.д.). Ця ідея заключається в тому, щоб не давати користувачу лишній функціонал, не перенасичувати інтерфейс програми та інкапсулювати дані, які користувачу не потрібно змінювати. І в зв'язку з цим, тема створення додатку "Менеджер фінансів" є актуальною.

Додаток "Менеджер фінансів" орієнтований на будь-якого користувача, який не планує проводити багато часу за плануванням власного чи сімейного бюджету. Додаток може використовуватися як і в особистих цілях, так і в цілях контролю бюджету невеликого підприємства.

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ ЗАСОБІВ ДЛЯ КЕРУВАННЯ ВЛАСНИМ БЮДЖЕТОМ

1.1. Основні концепції ведення власного бюджету

Фінансове планування повинне ґрунтуватися на реалістичних прогнозах і вирішувати грошові проблеми і давати зрозумілі і чіткі часові рамки.

Важливо також, що саме бюджетування – це досить цікаве заняття і після деякого часу звикання перетворюється на захоплююче проведення часу.

Багато хто, провівши за обліком витрат і доходів декілька місяців, настільки захоплюються процесом, що забувають, для чого це робиться. Бюджет сам по собі цінності не має, якщо він не вирішує описаних вище завдань. Результатом ведення обліку повинні стати важливі дані, які слід регулярно аналізувати. На основі цього аналізу приймаються конкретні обґрунтовані рішення.

Головні кроки для ведення власного бюджету.

Крок 1. Найпростіший спосіб розпочати – згадати, де зберігаються гроші. Практично у кожного є гаманець, куди ці гроші регулярно або хоча б іноді потрапляють. Тому першим записом в списку місць зберігання грошей будуть "готівкові". Хтось на цьому і закінчить, а інший продовжить, записавши, наприклад, "депозит в банку", зарплатну картку і т.д.. У програмах обліку особистих фінансів цей крок називається "створенням рахунків". Дійсно, у сучасної людини все менше грошей зберігається в готівці і все більше – на рахунках у банках. Очевидно, що кредитні карти теж слід записати в рахунки, різниця тільки в тому, що баланс на них буде негативним. Іншим видом рахунків можна вважати боргові.

При записі рахунків корисно вказувати, в якій валюті ці рахунки ведуться. Наприклад, в одному банку може бути два вкладу – гривневий і доларовий, тоді слід записати два рахунки і в назві відобразити тип валюти.

Крок 2. Створити власний список категорій витрат і доходів. Якщо рахунки дозволяють просто розуміти, де знаходиться та чи інша сума, то

категорії дають можливість розібрати, на що гроші витрачаються і звідки вони надходять. Багато програм пропонують вже готовий список категорій витрат і доходів. Але найкраще – це скласти свій, тому що у кожного своя ситуація і універсального списку не існує. Чужий список категорій можна використовувати в якості підказки для створення власного, додаючи, прибираючи або видозмінюючи категорії. Типовий список:

– Доходи:

- зарплата;
- банківські вклади;
- стипендія;
- пенсія.

– Витрати:

- автомобіль (ремонт, обслуговування, мийка);
- благодійність;
- банківське обслуговування;
- гігієна (перукарні, косметика, засоби особистої гігієни тощо);
- квартира (електрика, газ, вода);
- медицина;
- непередбачене;
- громадське харчування (харчування поза домом);
- одяг, взуття, аксесуари;
- подарунки;
- поїздки (відпустка, поїздки у вихідні і т.д.);
- допомога родичам;
- продукти;
- розваги (ресторани, кіно і т.д.);
- зв'язок (інтернет, мобільні телефони);
- спорт;
- транспорт (бензин, громадський транспорт).

Всі типи витрат складно передбачити в категоріях, а заради деяких типів витрат просто недоцільно створювати категорію, якщо такі витрати

відбуваються дуже рідко і в невеликих розмірах. У такому випадку витрату заносять в категорію "непередбачене". Також в цю категорію потрапляють всі помилки у веденні обліку, якщо такі виникають.

Крок 3. Далі слід розпочати вести облік витрат і доходів. Дії по витрачання коштів, їх отриманню чи перекладу між рахунками прийнято також називати транзакцією. Кожній транзакції повинен відповідати певний рахунок, сума грошей, категорія витрат чи доходів або перевід (якщо гроші переводяться між власними рахунками). Також для кожної транзакції можна вводити примітки. Вони бувають корисні для подальшого аналізу.

1.2. Аналіз додатків для керування власним бюджетом

Деякі користувачі почали використовувати програми для домашньої бухгалтерії ще задовго до появи сучасних мобільних пристроїв. Як тоді, так і зараз на ринку є дуже прості додатки, в які можна вносити свої доходи і витрати, а також сортувати всі операції по категоріях.

Але є і великі фінансові "комбайни". Їх функціонал значно ширше. За допомогою подібних додатків користувачі зможуть відстежувати коливання курсу акцій на ринку, вести облік витрат за складними схемами, стежити за своїми інвестиціями і т.д. При бажанні можна буде навіть подивитися докладний звіт про витрати на той чи інший аспект.

Для таких великих інструментів існує ряд недоліків:

- не існує безкоштовних версій
- дуже рідко вони націлені на пересічну людину
- перенасичений функціонал
- не є зручними в користуванні на мобільних платформах

Для огляду та аналізу існуючих додатків потрібно визначити категорію такого типу. Обрана категорія – "Фінанси" (рис.1.5).

Andromoney – додаток, який є одним з лідерів за кількістю скачувань в *Google Play* з високою середньою оцінкою. Досить простий і злегка недоладний інтерфейс програми ніяк не позначається на її функціональності (рис. 1.6, 1.7). Стартове вікно при запуску програми показує фінансові операції за обраним

користувачем проміжок часу (поточний день, тиждень, місяць або рік). Класичне бічне меню, яке викликається свайпом, містить такі розділи, як рахунки, бюджет, категорії та звіти. Незважаючи на вже наявні шаблони, рахунки та категорії піддаються детальному налаштуванню, а в якості звіту додаток пропонує вивести різні типи діаграм.

AndroMoney відображає баланс грошових коштів і має функцію транзакцій між рахунками. Додавати фінансові операції не складає труднощів. Бюджет в програмі можна створити на рік, місяць, тиждень і навіть день. Крім того, є можливість визначати бюджет для окремих категорій, а також включити попередження при перевищенні місячного бюджету.

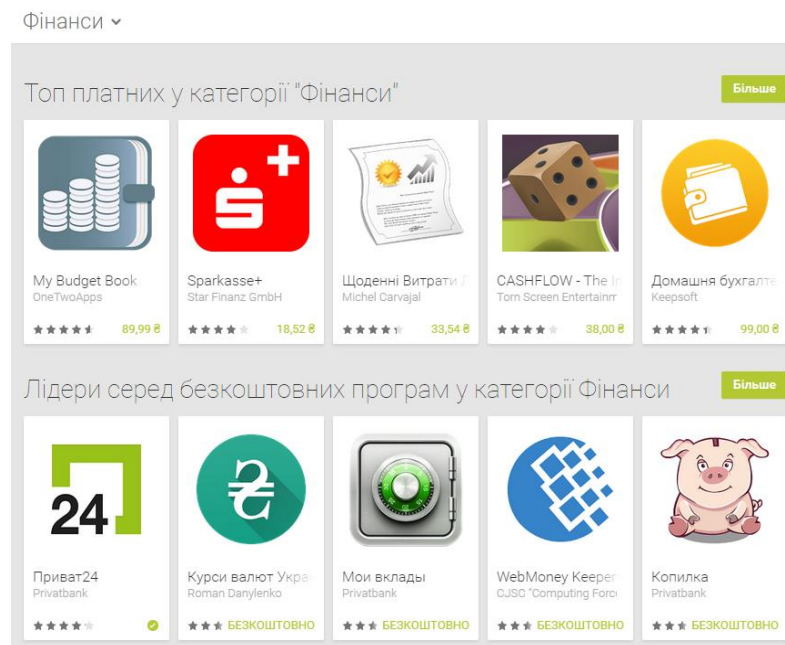


Рис. 1.5 Категорія "Фінанси" на сайті [Play Google](#)

У налаштуваннях програми слід вибрати основну валюту, до якої будуть прив'язані всі грошові маніпуляції. Важливою особливістю *AndroMoney* є експорт / імпорт баз даних. Зробити це можна кількома способами, серед яких є *Dropbox* і *Google Drive*. Характерні недоліки у додатку: місцями не зовсім коректний російський переклад, українського взагалі немає, також наявність рекламних банерів, за відсутність яких доведеться заплатити, застарілий інтерфейс, який не є дуже зручним, надлишкова функціональність.

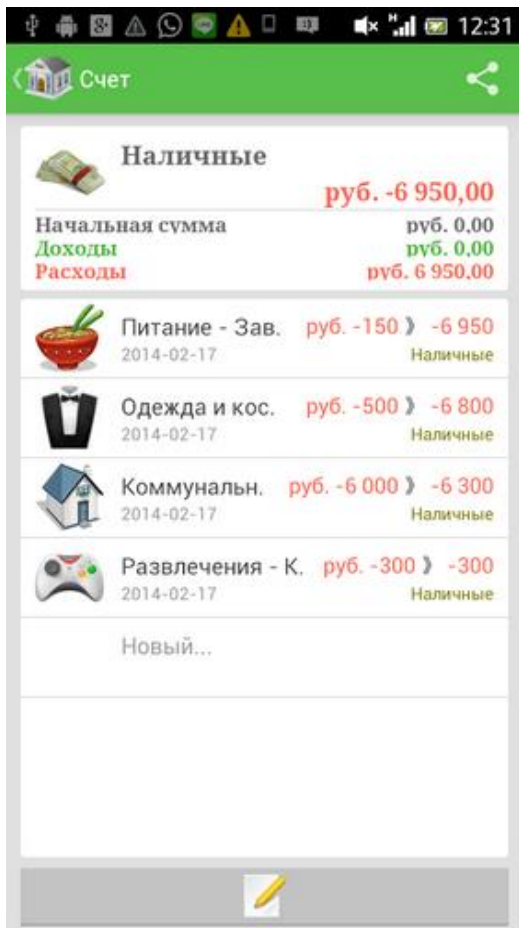


Рис 1.6 Додаток *Andromoney*
Счет

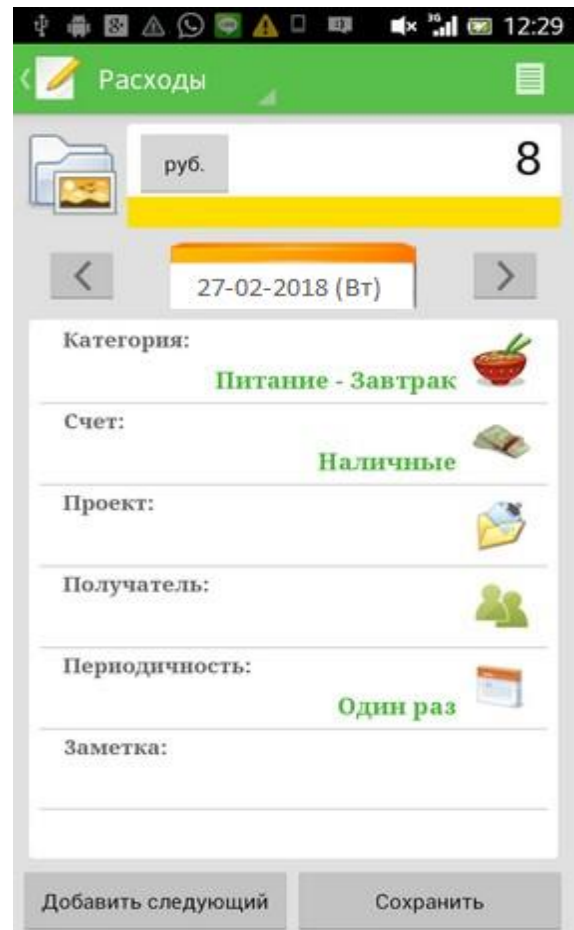


Рис.1.7 Додаток *Andromoney*
Расходы

Expense Manager (рис. 1.8). Незважаючи на велику кількість завантажень (близько одного мільйона), додаток є досить простим планувальником фінансів. Розробники заздалегідь поставили кілька основних пунктів витрат (їх можна змінити в настройках). Крім цього, користувачі можуть налаштувати повторювані платежі або подивитися графік витрат за тиждень, місяць і рік.

На жаль, повна версія програми є платною. Також серед мінусів додатку варто відмітити його не дуже зручний інтерфейс та відсутність перекладу на українську мову.

"Витрати" (рис. 1.9, 1.10). Додаток дозволяє вести облік щоденних сімейних витрат за різними категоріями і тегами, створювати і змінювати існуючі категорії витрат, а також переглядати статистику витрат (за день, тиждень, місяць, рік і будь-який період) за категоріями і тегами. У кожній категорії доступна повна історія витрат з можливістю вносити зміни. Додаток

дозволяє настроїти резервне копіювання даних в *Dropbox* папку користувача за розкладом.

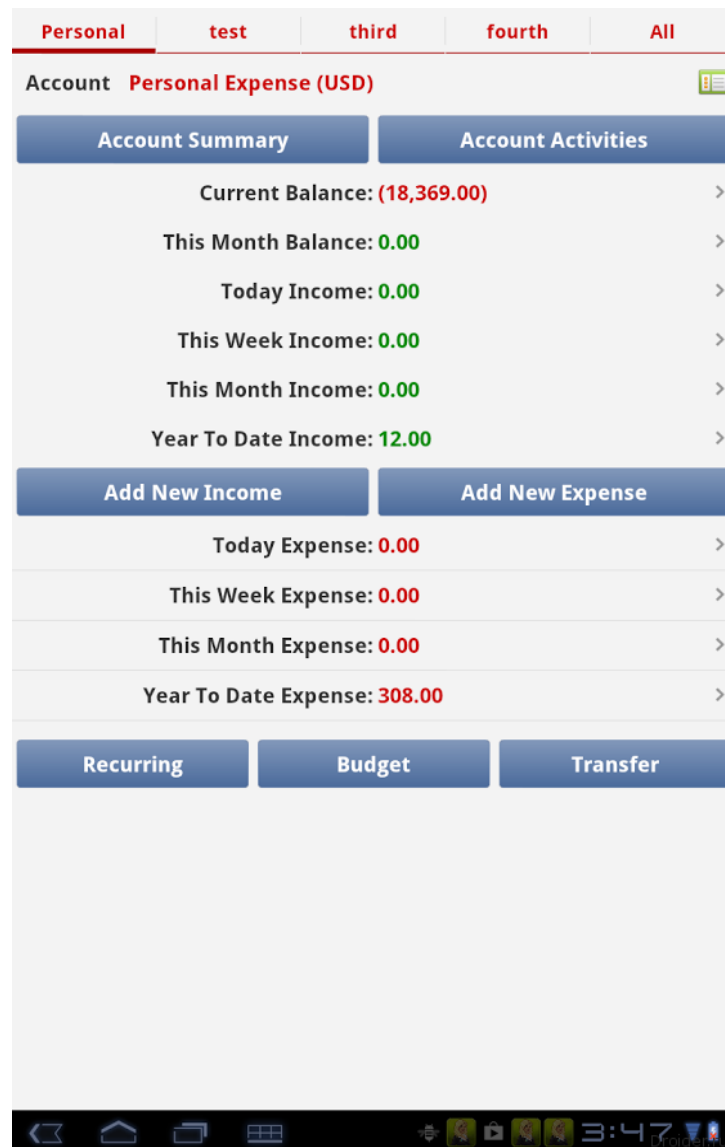


Рис. 1.8 Додаток *Expense Manager*

Серед мінусів програми "Витрати":

- рахує лише витрати;
- не має українського перекладу;
- застарілий інтерфейс;
- не зручний в користуванні;
- розробники вже більше року не оновлюють додаток;
- немає пункту "заощадження"
- дуже дрібний шрифт в пункті "статистика"

– немає графіків

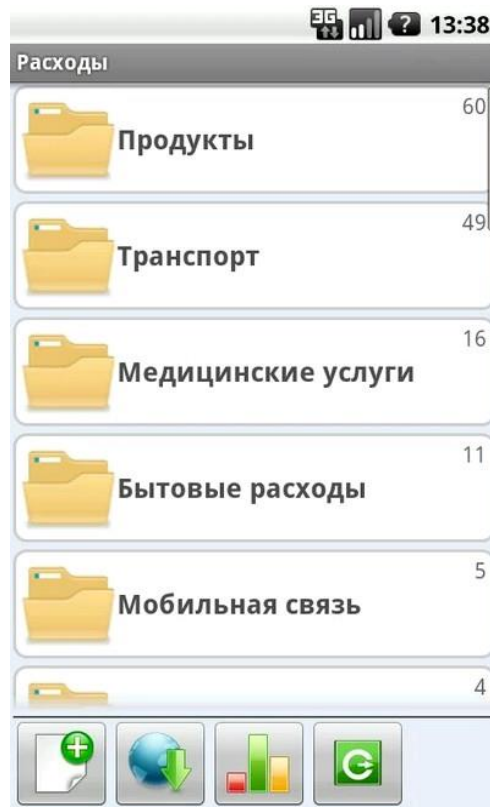


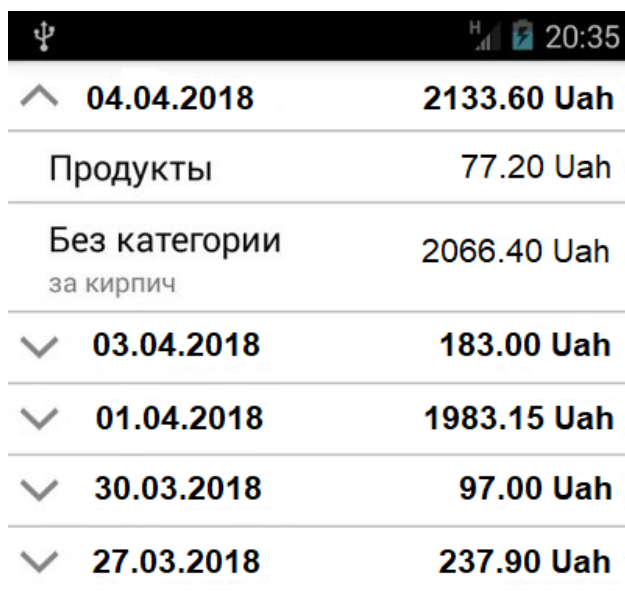
Рис. 1.9 Додаток "Витрати" Расходы



Рис. 1.10 Додаток "Витрати" Статистика

"5 копеек – учет расходов" (рис. 1.11). Додаток рахує лише витрати. Занадто мало можливостей. Немає української мови.

Finance PM – Expense Manager (рис. 1.12, 1.13). Крім обліку руху грошових коштів, програма пропонує управління боргами, бюджетом та майбутніми транзакціями



↑	04.04.2018	2133.60 Uah
	Продукты	77.20 Uah
	Без категории за кирпич	2066.40 Uah
↓	03.04.2018	183.00 Uah
↓	01.04.2018	1983.15 Uah
↓	30.03.2018	97.00 Uah
↓	27.03.2018	237.90 Uah

Рис. 1.11 Додаток "5 копеек – учет расходов"

В додатку відсутні обмеження, що, безсумнівно, радує. Так, наприклад, можна створити необмежену кількість гаманців з різною валютою або задати ліміти (бюджет) не тільки на часовий інтервал, а й на певні категорії. *Finance PM* дозволяє запланувати операції за витратами, доходами або переказу грошових коштів на певну дату з можливістю щоденного, щотижневого, щомісячного повторів і навіть у певні дні тижня чи місяця. Переглянути очікувані транзакції можна в розділі "Найближчі операції", а всі минулі – у розділі "Грошові переміщення". Відмінною особливістю програми є функція створення шаблонів за різними типами фінансових операцій. Це досить корисна

опція, яка значно скорочує час, якщо, приміром, є часто повторюваний дохід, витрата або переклад з однаковою сумою грошей. Ще одна цікава особливість – можливість налаштувати зовнішній вигляд і елементи інтерфейсу на головному вікні. *Finance PM* підтримує синхронізацію і може зберігати дані в *Excel*.

Недоліки:

- не дуже зручний інтерфейс з дрібним шрифтом та елементами;
- додаток не може вивести підсумковий загальний баланс за всіма джерелами доходів (хоча тут ситуацію трохи рятує наочна картина по залишку на всіх гаманцях);
- в пункті "звіти" можна вивести дані або про доходи, або про витрати, без сумарного результату між ними;
- реклама, яка найбільш настирливо себе проявляє при перегляді звітів.

У підсумку, *Finance PM* ніби балансує між непоганою функціональністю і не зовсім вдалою реалізацією.

Toshl Finance (рис. 1.14, 1.15). Кожний календарний день у додатку являє собою закладку з фінансовими операціями, а помічником в програмі виступає одноокий монстр. Все це додає інтерактивності додатком і мотивує ним користуватися.

Щоб почати роботу, необхідно зареєструватися, вибрати основну валюту та синхронізуватися з сервером.

У той час як багато розробників намагаються функціонально наповнити свої додатки по максимуму, *Toshl Finance* має тільки базові функції і включає такі розділи як витрата, дохід, статистика і бюджет. Замість звичних категорій витрат і доходів тут використовуються мітки. *Toshl Finance* платний – підписка \$1,99 / міс., \$19,99 / рік, яка дозволяє додавати необмежену кількість доходів і бюджетів, експортувати звіти в *PDF*, *Excel* і *Google Docs*. Також не має української мови та використовується доволі застарілий інтерфейс.

Загалом по дизайну *Toshl Finance* схожа на дитячу програму, а додатки фінансової категорії – це інструменти, які створені для серйозних цілей, таких

як бізнес, інвестиції тощо. Тому додаток, створений у вигляді дитячої гри не створює потрібного враження.



Рис 1.12 Додаток *Finance PM*

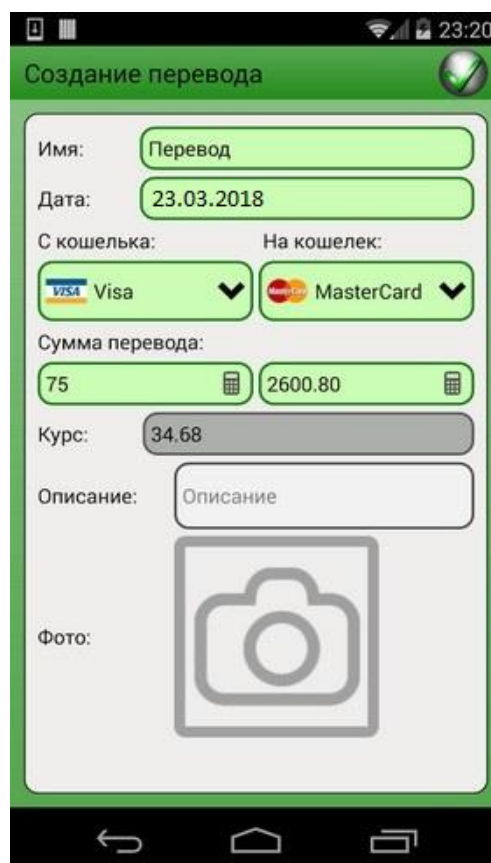
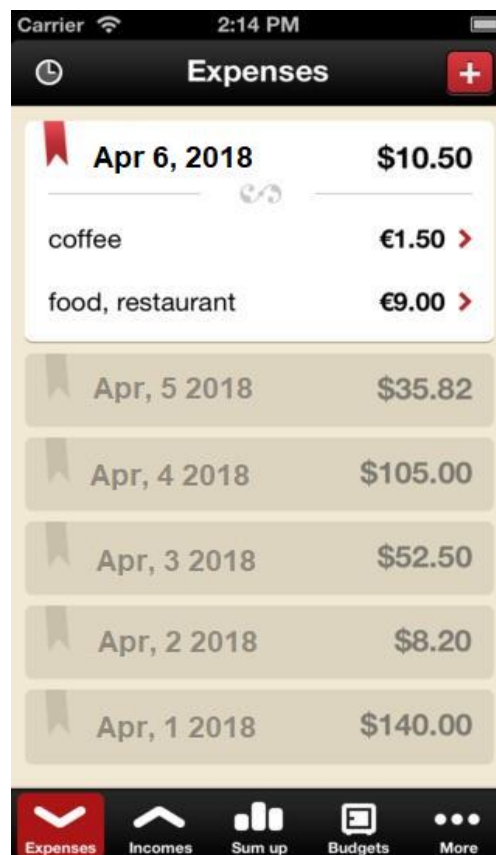
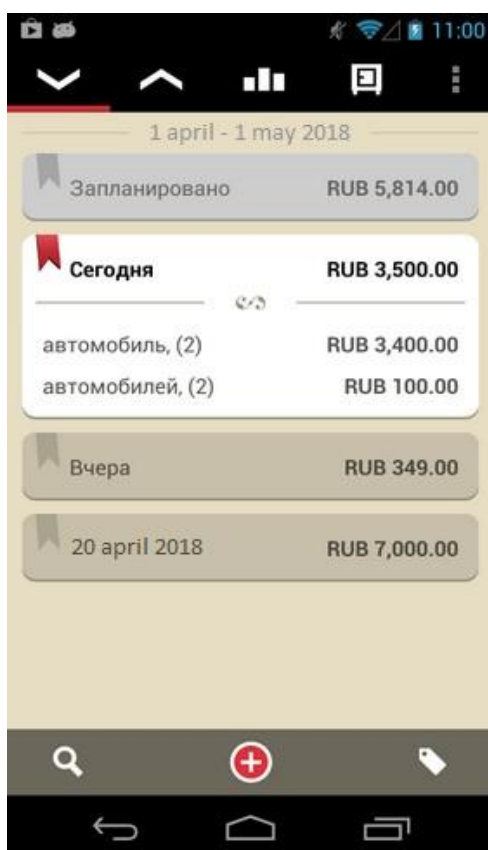


Рис.1.13 Додаток *Finance PM*



Висновки до розділу

В цьому розділі розглянуті основні поліграфічні видання, які дозволяють організувати власні фінанси. Виявлений ряд недоліків такого типу контролю доходів та витрат:

- Користувачу необхідно проводити всі розрахунки власноручно, а це займає немало часу.
- Незручно проводити аналіз доходів і витрат, вести статистику, виявляти відсоткове співвідношення витрат і т.д., особливо якщо потрібно проаналізувати дані за тривалий період.
- Обмежені можливості ведення обліку.
- Немає можливості додавання власних категорій витрат.

Також розглянуті додатки під ОС *Android* для керування власним бюджетом, в яких також виділено ряд недоліків:

- Немає українського перекладу.
- Деякі додатки є платними (наприклад, щоб користуватися повною версією додатка *Expense Manager* потрібно за нього платити).
- Немає можливості використовувати зразу декілька валют.
- Надлишкова функціональність.
- Застарілий інтерфейс.

В зв'язку з визначеними недоліками вирішено розробити автоматизовану систему для керування особистими доходами та витратами на мобільну операційну систему *Android*.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА ВИМОГИ МОБІЛЬНОГО ДОДАТКУ "МЕНЕДЖЕР ФІНАНСІВ"

2.1. Розробка структури додатку

При розробці дизайну додатка потрібно врахувати безліч чинників. В додатку повинні бути правильно розставлені елементи управління, з урахуванням правил юзабіліті та інших технічних особливостей, необхідних для швидкої і якісної роботи. Дизайн мобільних додатків має ряд особливостей і кардинально відрізняється від всіх інших напрямків візуального оформлення. Для розробки якісного дизайну, обов'язково потрібно знати основні мови, що використовуються при написанні даного типу програм, тільки при дотриманні всіх правил можна розраховувати на успішний розвиток і якісну роботу мобільного додатка.

2.1.1. Загальна схема додатку

Для розробки графічного інтерфейсу створено, перш за все, схему архітектури додатку, тобто схематично зображено всі його вікна (рис. 2.1).

Для створення загальної схеми додатку "Менеджер фінансів" були використані можливості програмного пакету Microsoft Office Word. А саме передбачені ним вставки схематичних фігур.

На схемі архітектури додатку "Менеджер фінансів" зображені основні елементи інтерфейсу:

- бокова панель дає зручний доступ до усіх фрагментів вікон додатку;
- рахунки, доходи, витрати, заощадження, шаблони, статистика, налаштування – це фрагменти вікон програми, які виконують функції відповідні до назв елементів;

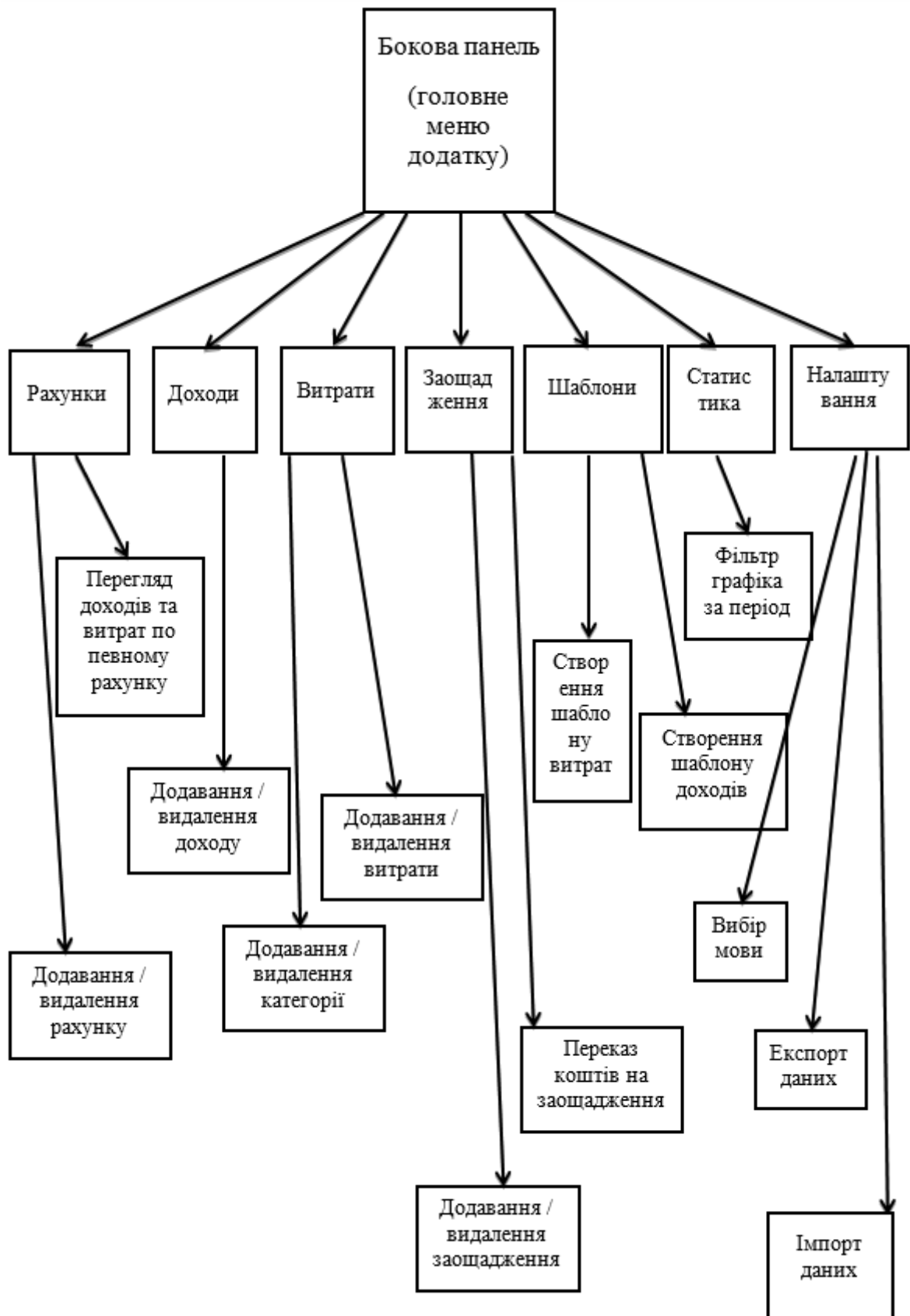


Рис 2.1 Загальна структурна схема додатку

– засоби керування, наприклад, кнопка, список, меню тощо.

Схема розроблена виходячи з того, що додаток розробляється під мобільну платформу Android. Тобто використовуються нові програмні бібліотеки та засоби для зручності в використанні додатку. Наприклад, основний елемент керування можна відкрити з будь-якого вікна додатку, не закриваючи його. Це дуже зручно, оскільки існує велика кількість різних розширень екранів мобільних телефонів та планшетів, а бічна панель є універсальним рішенням як для малих розширень, так і для великих.

2.1.2. Функціональні можливості графічного інтерфейсу додатку

У зв'язку з сучасною тенденцією до спрощення, головне вікно привітання не є обов'язковим, а навпаки, ускладнює користування додатком. Тому вирішено, що запускатися програма буде зразу у вікно "Рахунки" (рис. 2.2).

Для малювання елементів інтерфейсу додатку використовується програмне забезпечення Adobe Photoshop CS6 (рис. 2.3). Для створення скріншотів використовується програма "Ножиці" (рис. 2.4).

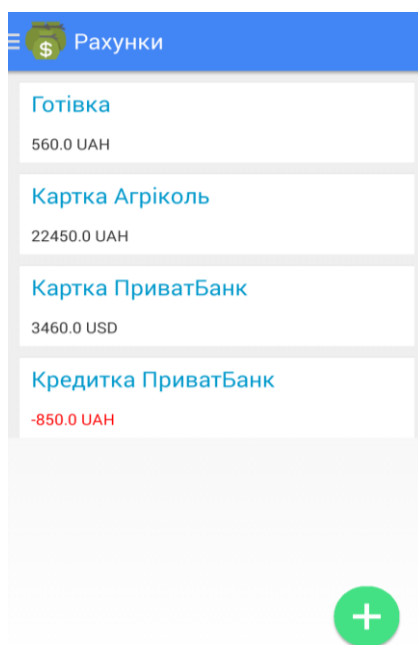


Рис. 2.2 Форма "Рахунки"

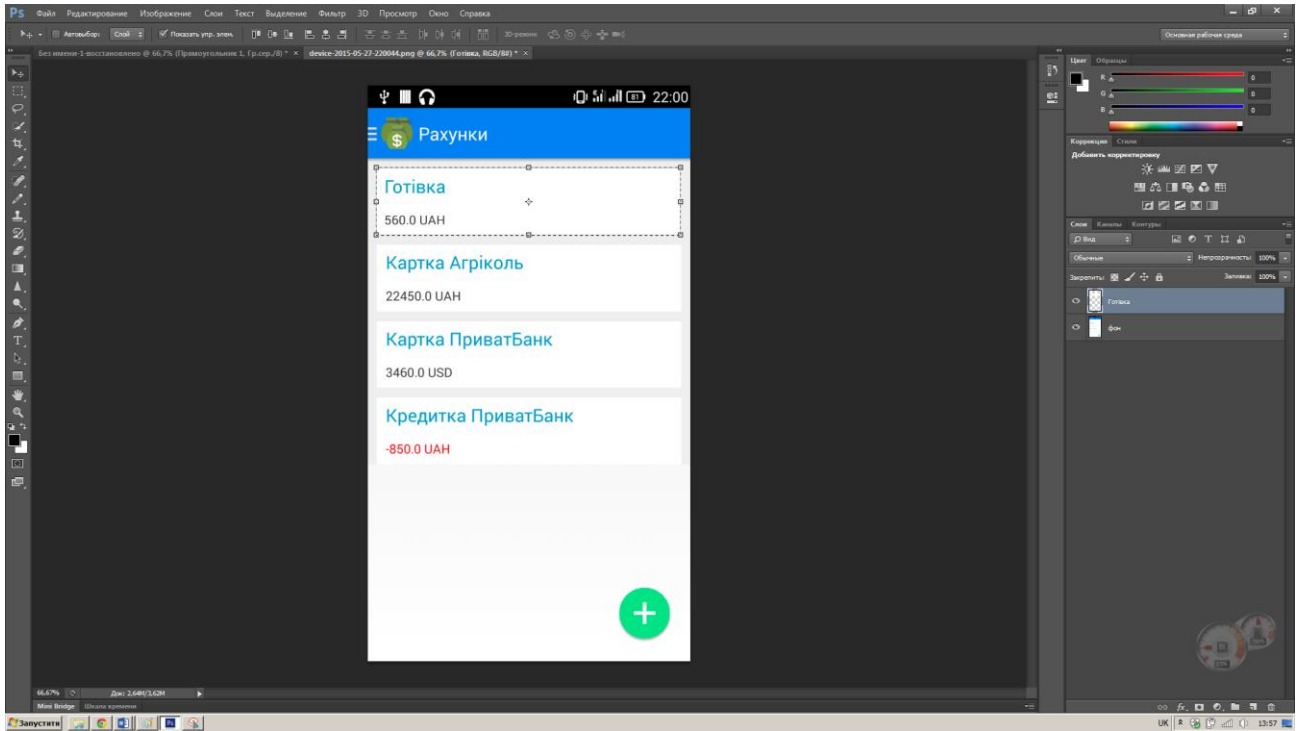


Рис. 2.3 Створення макету додатку в Adobe Photoshop CS6

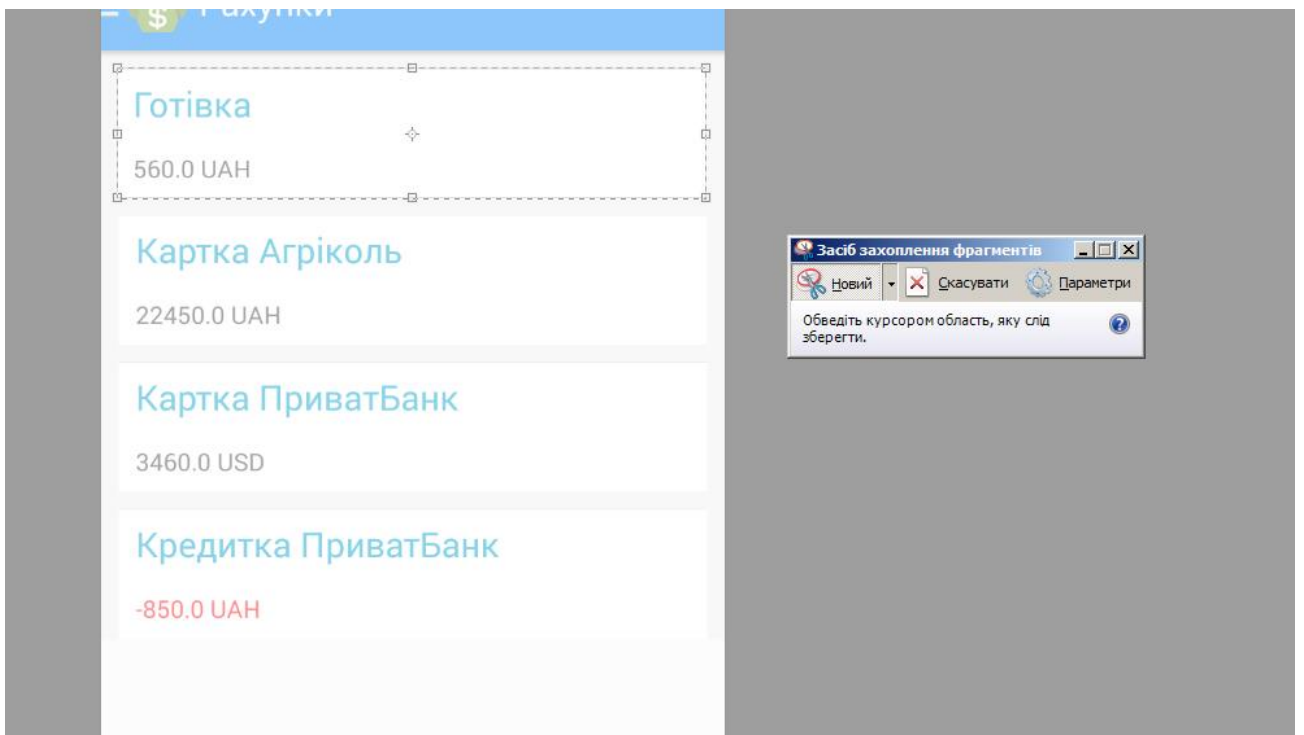


Рис. 2.4 Програма "Ножиці" для створення знімків екрану

За допомогою бокової панелі (рис. 2.5) буде здійснюватися вибір всіх інших вікон. Для мобільно додатку потрібно якомога більше розділити

елементи керування, і ні в якому разі не створювати багато з них на одному вікні. Це призведе до незручності в користуванні.

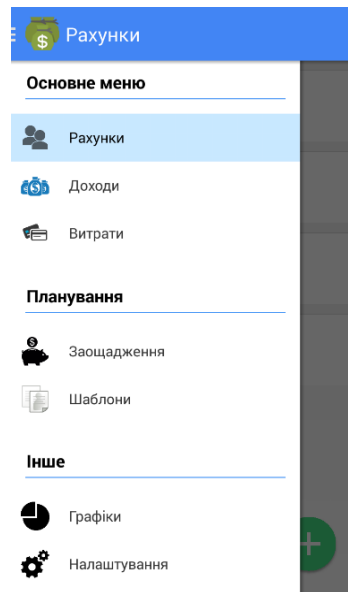


Рис. 2.5 Бокова панель додатку (головне меню)

У вікнах "Рахунки", "Доходи", "Витрати" та "Заощадження" створені кнопки для додавання відповідних елементів (рис. 2.6).

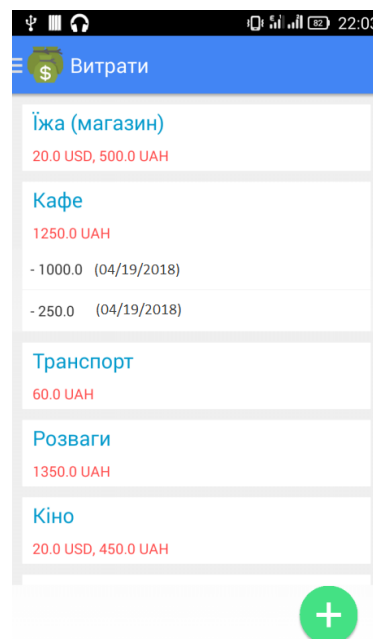


Рис. 2.6 Форма "Витрати" з кнопкою додавання витрат в загальний список

Для видалення та редагування того чи іншого елемента в списку використовується довге натискання на ньому та запускається контекстне меню

(рис. 2.7). Відповідно до вибраного пункту контекстного меню виконується певна дія.

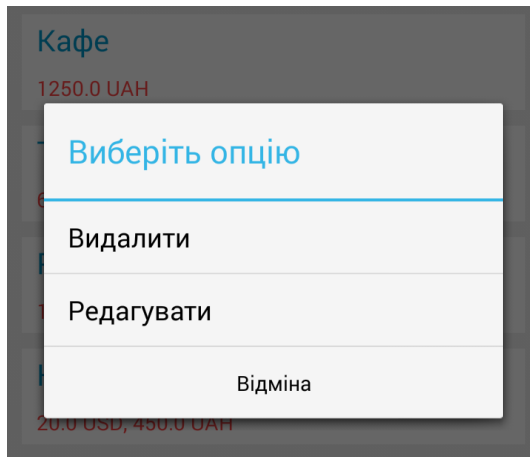


Рис. 2.7 Контекстне меню

Також необхідне вікно – "Статистика". В ньому є можливість переглянути усі доходи чи витрати за певний період часу (рис. 2.8, 2.9).

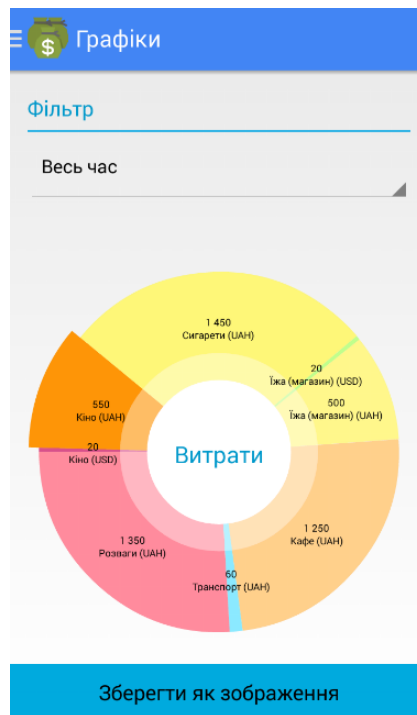


Рис. 2.8 Діаграма форми "Статистика за весь час"

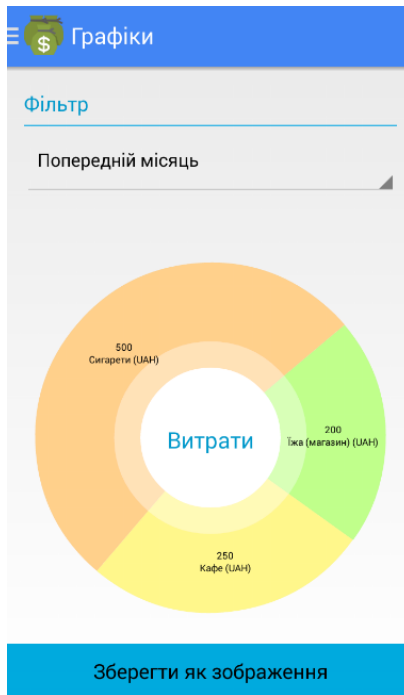


Рис. 2.9 Діаграма форми "Статистика за період"

В вікні шаблони є можливість створювати операції, які часто повторюються (наприклад, дохід "зарплата", або навпаки, витрати "оренда квартири"). Це вікно створено для зручності, щоб користувач не вводив одні і ті ж самі дані кожного разу, коли це необхідно (рис. 2.10, 2.11).

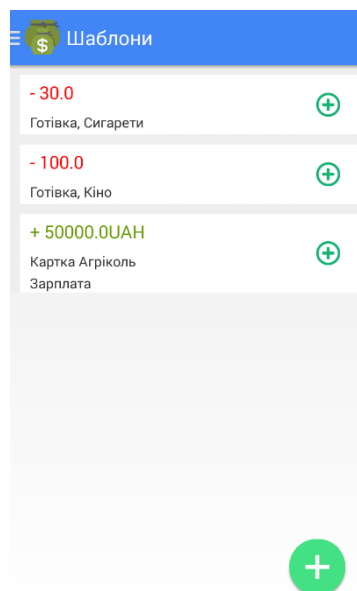


Рис. 2.10 Форма "Шаблони"

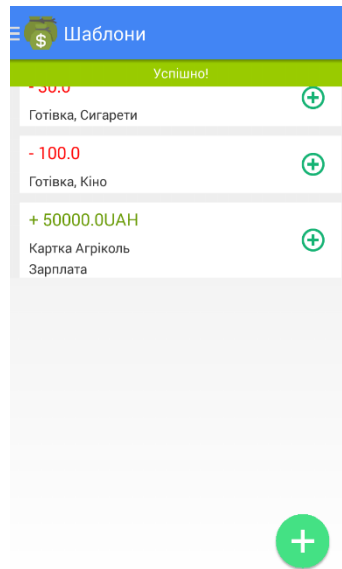


Рис. 2.11 Успішне використання шаблону

2.2. Ергономіка та дизайн додатку

Загальний дизайн та ергономіка додатку передбачає безліч факторів, таких як вибір кольорів, вибір шрифтів та їх характеристик, макет додатку, правильний вибір категорії користувачів, які користуватимуться ним. В залежності від виду додатку та цих користувачів і підбираються весь його дизайн.

На рахунок ергономіки, оскільки додаток розробляється для мобільної платформи, то елементів на кожному вікні має бути мінімум, тільки ті, без яких не обійтись.

2.2.1. Вибір кольорової гамми для додатку

Колір, безумовно, є важливим джерелом емоції. Кольори можуть встановлювати правильний тон і передавати необхідні емоції користувачу, можуть схвилювати, викликати безліч почуттів і стимулювати до дій. Він є надзвичайно потужним фактором впливу на користувачів.

При виборі колірної гами для додатку, важливо зробити це правильно, керуючись основними принципами теорії кольору.

Кольоровий круг (рис. 2.12) є неодмінним атрибутом багатьох дизайнерів і художників по всьому світу. Це ідеальні докази теорії, що геніальне завжди

просто. Коло дозволяє вибрати кольори, які гармоніювали б разом. Він складається з 6 основних кольорів: червоний, оранжевий, жовтий, зелений, синій, фіолетовий і додаткових кольорів.

Щоб знайти правильну кольорову схему, необхідно використовувати будь-які два кольори один навпроти одного, будь-які три кольори на рівній відстані при формуванні трикутника або будь-який з чотирьох кольорів, що утворюють прямокутник (дві пари кольору один навпроти одного). Колірні схеми залишаються правильними незалежно від кута повороту.

Основні кольори.

Є три основні кольори: червоний (# ff0000), жовтий (# FFFF00) і блакитний (# 0000FF). Їх не можна отримати шляхом змішування інших кольорів. Додаткові кольори можуть бути сформовані шляхом об'єднання цих трьох кольорів. Основним кольором додатку "Менеджер фінансів" обрано блакитний, адже у всьому світі він символізує безпечність.

Також, якщо користувач помилився у своїх діях, то зображується вікно помилки червоного кольору. Саме червоного тому, що він найбільш помітний, судячи з діаграми кольорів. Також він символізує ненависть та пристрасть, тому цей колір є ідеальним для такого роду повідомлень. Ним же ж зображений текст суми залишку на рахунку, якщо та є від'ємною.



Рис. 2.12 Кольоровий круг

Третинні кольори.

Щоб отримати один з третинних кольорів, необхідно змішати один основний колір і один вторинний колір. Можливості для третинних кольорів безмежні.

Додаткові кольори.

Додаткові кольори розташовані прямо навпроти один одного на колірному колі: червоний і зелений, синій і помаранчевий, фіолетовий і жовтий. У поєднанні один з одним, вони складають різкий контраст. Такі поєднання, як правило, використовуються для виділення деяких елементів, проте, не є бажаним їх надмірне використання.

Аналогічні кольори.

Ці кольори розташовані поруч один з одним на колірному колі. Вони зазвичай дивляться дуже добре разом. Використання таких кольорових сполучень викликає почуття комфорту у користувача.

Існують декілька корисних правил, які допомагають у виборі правильної колірної схеми для мобільного додатку. Ці правила широко використовують професійні дизайнери мобільних додатків та web-дизайнери:

- щоб текстовий вміст, було легко читаним, вибирайте контрастні кольори;
- оптимальна кількість кольорів;
- якщо потрібно залучити користувача, потрібно застосовувати інтенсивні кольори.

2.2.2. Вибір використовуваних шрифтів у додатку

Підбираючи шрифти завжди потрібно опиратися об свою основну мету. Оскільки категорія користувачів – середньостатистична людина, то потрібно використовувати більш стандартні шрифти, які є дуже зручними при читанні.

Скільки використовувати шрифтів повністю залежить основної мети додатку, але завжди потрібно мати на увазі який ефект потрібно отримати.

Немає ніякого правила в тому яка максимальна кількість шрифтів повинна бути на вікні додатку, але чим більше число шрифтів тим важче досягти гармонії. Однак якщо це вдається то результат виходить декоративно-яскравим.

Проте для фінансової тематики в додатку "Менеджер фінансів" потрібно використовувати мінімальне число шрифтів. Один з крайніх варіантів використовувати одну гарнітуру, але з декількома розмірами і вагами, тоді можна бути впевненим що шрифти добре доповнять один одного.

Успішне поєднання досягається згодою і контрастом, але не конфліктом. Шрифти можуть добре працювати шляхом поєднання певних зовнішніх якостей або їх повної відмінності, але будуть конфліктувати якщо занадто подібні один одному і розрізняються тільки в дрібних деталях.

Згода (узгодженість) двох шрифтів передбачає наявність однієї ознаки в них обох. Це може бути кернінг, пропорції, вага і інші ознаки.

Один із способів досягнення згоди це вибір шрифтів з одного сімейства.

Приклад: сімейство Droid розроблене Стівом Матесоном для використання в перших версіях Android від Google (рис. 3.13). Шрифт із зарубками для заголовків і без для основного тексту.

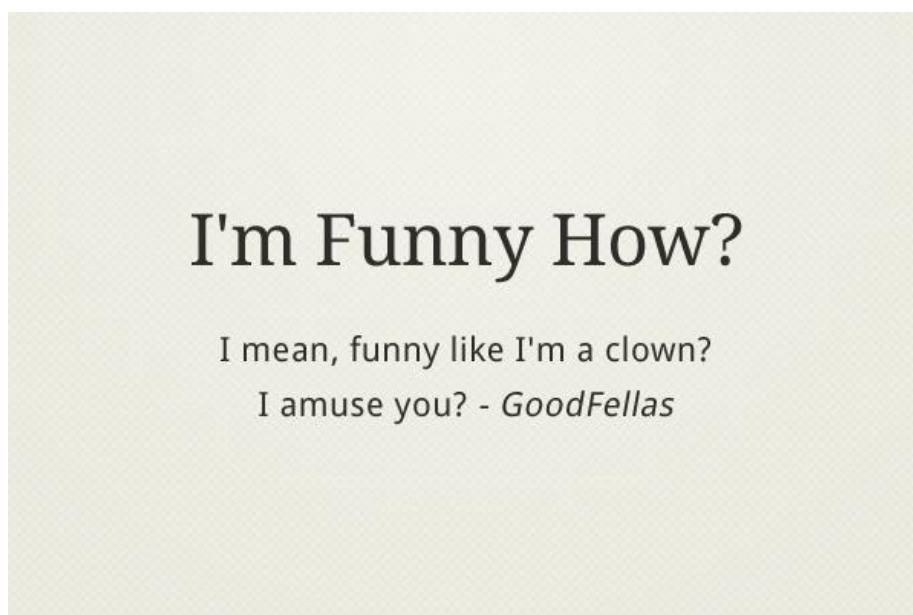


Рис. 2.13 Шрифт сімейства Droid в перших версіях Android

Шрифти чисті і сучасні, хоча і не так сильно пристосовані для екранів з великою щільністю екрану через що Google і перейшов на шрифт Roboto, але добре підходять для інтернету через свою велику висоту рядкових символів, що допомагає читабельності.

Так само шрифти одного і того ж дизайнера часто поділяють загальні якості і сприяють досягненню гармонії.

Roboto (рис. 2.14) – шрифтова гарнітура без зарубок представлена разом з операційною системою Android 4.0 "Ice Cream Sandwich". Google описує шрифт як "сучасний, але доступний" і "хвилюючий". Гарнітура ліцензована під ліцензією Apache. Весь набір шрифтів офіційно доступний для безкоштовного завантаження з 12 січня 2012 року, на недавно запущеному сайті Android Design. Гарнітура включає в себе пряме і похиле накреслення для шрифтів різної ширини (Thin, Light, Regular, Medium, Bold, Black, Condensed).

Roboto
SUNGLASSES
Self-driving robot lollipop truck
Fudgesicles only 25¢
ICE CREAM
Marshmallows & almonds
#9876543210
Music around the block
Summer heat rising up from the boardwalk

Рис. 2.14 Основний шрифт в Android 4.0+ Roboto

Тому для додатку "Менеджер фінансів" вирішено використовувати саме цей шрифт, запропонований розробником мобільної операційної системи Android –Google Inc.

Висновки до розділу

Розроблена архітектура додатку у вигляді схеми. Після чого визначені функціональні можливості графічного інтерфейсу додатку. За допомогою цих даних були підібрані шрифти та кольори, які зображуються в додатку. Основним кольором вибрано синій, а основним шрифтом – Roboto, який представила корпорація Google разом з Android.

РОЗДІЛ 3

РОЗРОБКА ФУНКЦІОНАЛЬНОЇ ЧАСТИНИ МОБІЛЬНОГО ДОДАТКУ

"МЕНЕДЖЕР ФІНАНСІВ"

3.1. Розробка структури бази даних

Оскільки в операційній системі Android використовуються бази даних SQLite, то потрібно засобами AndroidSDK створити чотири таблиці, які будуть з'єднані між собою на програмному рівні.

- Рахунки – для окремих рахунків, наприклад, "готівка", "зарплатна картка" тощо.
- Доходи – це гроші, які надходять на певні рахунки.
- Витрати – це витрачені гроші на якийсь товар чи послугу.
- Резерви – це заощадження, тобто певні суми грошей, які користувач бажає заощадити.
- Групи витрат (наприклад: одяг, їжа, бензин тощо).

Всі таблиці мають бути залежні одна від іншої. Тобто таблиця "Доходи" додає в таблицю "Рахунки" певні суми грошей, які користувач вніс до неї. Аналогічно працюватиме таблиця "Витрати", яка, в свою чергу, мінусуватиме значення кількості грошей на певному рахунку.

Таблиця "Резерви" має одну ціль – в неї можна перекладати гроші з певного рахунку з таблиці "Рахунки". Тобто в таблиці "Рахунки" сума віднімається, і ця ж сума перекладається в "Резерви".

В таблицях є також колонки для більш детальної інформації про них.

Наприклад, в таблиці "Рахунки" є такі колонки:

- Назва рахунку.
- Валюта, в якій відкрито рахунок.
- Сума, яка є на цьому рахунку в даний момент часу.

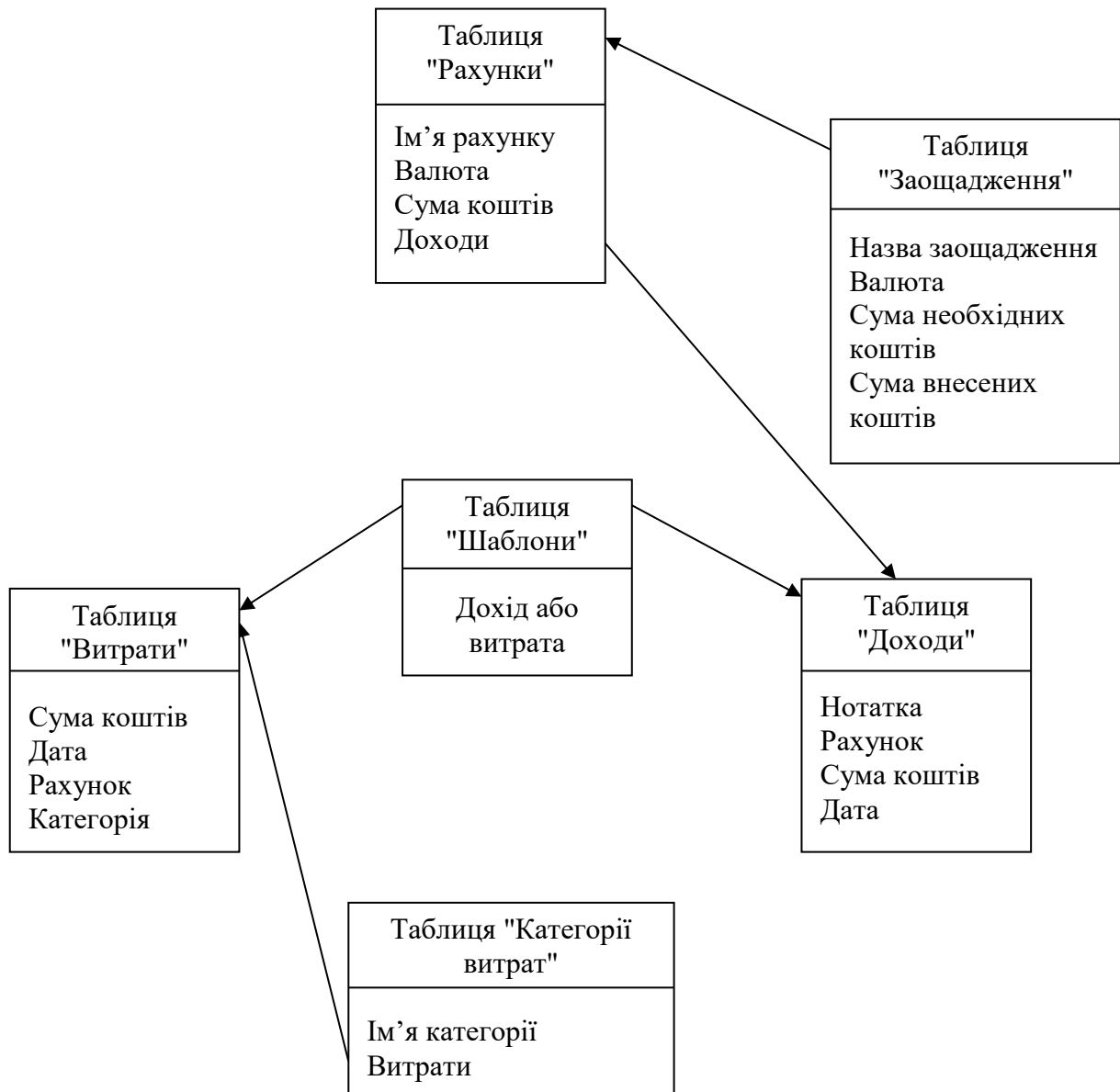
В таблиці "Доходи" є такі колонки:

- Назва доходу (наприклад: зарплата, відсотки з депозиту тощо).
- Номер рахунку в таблиці "Рахунки" – для з'єднання з таблицею

"Рахунки".

- Сума доходу.
- Дата, коли був отриманий дохід.

Розробимо схематичну структуру бази даних (рис. 3.1):



Якщо база даних **Рис. 3.1 Структура бази даних** ікається. Цей метод є перевизначеним з класу `SQLiteOpenHelper` (рис. 3.2), від якого є наслідуваним клас додатка "Менеджер фінансів" `DataBaseFinances`:

```
public class DataBaseFinances extends SQLiteOpenHelper
```

Для додавання даних в таблицю використовуються стандартні SQL транзакції. Для полегшення їх використання в мові програмування Java є спеціальні класи, наприклад, `ContentValues` .

Метод для додавання даних в таблицю "Рахунки":

```
public void putPay (String name, long time, float amount, int counter, int  
acc_indicator, boolean update)  
{
```

```

ContentValues cv = new ContentValues();
cv.put(DataBaseFinances.COLUMN_PAY_NAME, name);
cv.put(DataBaseFinances.COLUMN_PAY_DATE, time);
cv.put(DataBaseFinances.COLUMN_PAY_AMOUNT, amount);
cv.put(DataBaseFinances.COLUMN_INEXPENSES_COUNTER,
counter);

cv.put(DataBaseFinances.COLUMN_PAY_ACC, acc_indicator);
if (!update) {
    sqdb.insert(DataBaseFinances.TABLE_PAY,
DataBaseFinances.COLUMN_PAY_NAME, cv);
} else {
    sqdb.update(DataBaseFinances.TABLE_PAY, cv,
DataBaseFinances.COLUMN_INEXPENSES_COUNTER+"=?",
    new String []{String.valueOf(counter)});
}
cv.clear();
cv = null;
}

```

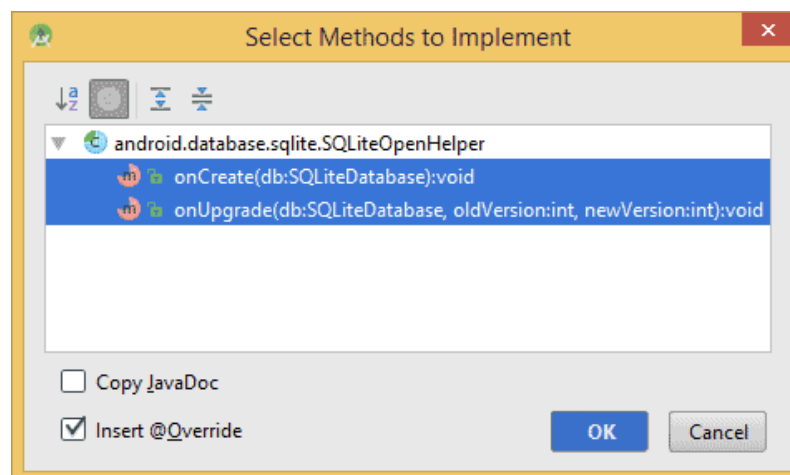


Рис. 3.2 Додавання наслідника класу SQLiteOpenHelper

Методи для додавання даних в інші таблиці аналогічні вищенаведеному. Змінюються лише назви колонок та самої таблиці.

Метод для вибірки елементу з комірки таблиці в базі даних:

```
public String getElementString (int ID, String table, String column)
{
    Cursor cursor = sqdb.query(table, new String[] {
        DataBaseFinances.COLUMN_INEXPENSES_COUNTER,column }, null, null,
null, null, null );

    String name = new String();
    while (cursor.moveToNext()) {
        int id = cursor.getInt( cursor.getColumnIndex
(DataBaseFinances.COLUMN_INEXPENSES_COUNTER) );
        if (ID == id)
return name =
cursor.getString(cursor.getColumnIndex(column));
    }
    cursor.close();
    return name;
}
```

Наведений вище метод виконується, якщо потрібно дістати з таблиці об'єкт типу даних String. Якщо потрібно дістати інший тип даних, то використовується аналогічний метод, але з іншими параметрами.

Наступним кроком є створення методу, який видаляє рядки з таблиці по порядковому номеру елемента:

```
public void deleteRow (String table, int id){
    sqdb.delete (table,
        DataBaseFinances.COLUMN_INEXPENSES_COUNTER
        + " = ?",
        new String[] { String.valueOf(id) }); }
```

Також потрібні методи для керування усіма таблицями в базі даних. Наприклад, підрахунок суми грошей в категорії витрат чи рахунків, заміна певних даних в таблиці тощо.

3.2. Верстка та розробка графічного інтерфейсу користувача

3.2.1. Опис ресурсів використовуваних в додатку

В середовищі AndroidStudio для верстки додатків використовується розширена мова розмітки XML. Є різні типи файлів XML для ОС Android:

- анімація;
- ресурси;
- малюнки;
- меню;
- інші XML файли.

На зображенні (рис. 3.3) показана таблиця з усіма видами ресурсів.

На схемі представлена тільки частина ресурсів. Система постійно оновлюється і додаються нові типи.

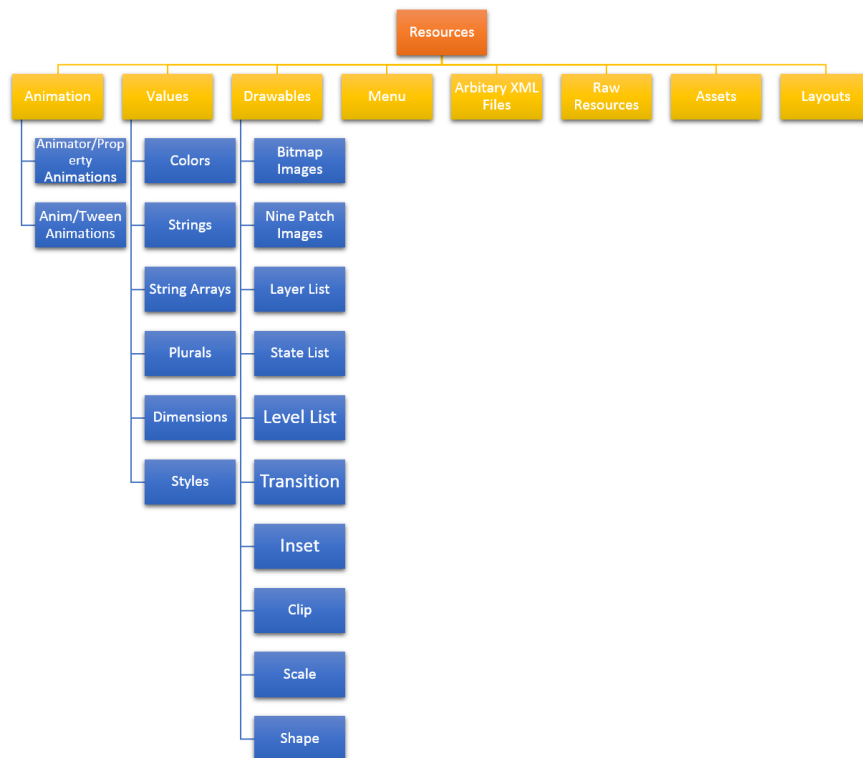


Рис. 3.3 Дерево ресурсів в ОС Android

Ресурси в Android є декларативними. В основному ресурси зберігаються у вигляді XML-файлів в каталозі `res` з підкаталогами `values`, `drawable-ldpi`, `drawable-mdpi`, `drawable-hdpi`, `layout`, але також бувають і інші типи ресурсів.

Для зручності система створює ідентифікатори ресурсів і використовує їх у файлі `R.java` (клас `R`, який містить посилання на всі ресурси проекту), що дозволяє посилатися на ресурси всередині коду програми. Статичний клас `R` генерується на основі заданих ресурсів і створюється під час компіляції проекту. При створенні клас містить статичні підкласи для всіх типів ресурсів, для яких був описаний хоча б один екземпляр.

Найпоширенішими ресурсами є, рядки (`string`), кольори (`color`) і графічні малюнки (`bitmap`). У додатку не рекомендується застосовувати жорстко написані рядки коду – замість них слід використовувати відповідні ідентифікатори, що дозволяє змінювати текст строкового ресурсу, не змінюючи вихідного коду.

Ресурси меню можна використовувати як для опису звичайного і контекстного меню. Меню, описане у форматі XML, завантажується у вигляді програмного об'єкта за допомогою методу `inflate`, що належить сервісу `MenuInflater`. Як правило, це відбувається всередині обробника `onCreateOptionsMenu`.

3.2.2. Верстка та розробка додатку “Менеджер фінансів”

Існує п'ять стандартних типів верстки:

- `LinearLayout`;
- `RelativeLayout`;
- `TableLayout`;
- `FrameLayout`;
- `AbsoluteLayout`;

AbsoluteLayout (користуватися не рекомендується, `deprecated`) – означає що кожен елемент верстки матиме абсолютну позицію щодо верхнього лівого кута екрана задається за допомогою координат x і y . Тобто верхній лівий кут екрану при `AbsoluteLayout` має координати $x = 0$, $y = 0$.

Позиція вказується в атрибутах елемента `android: layout_x` і `android: layout_y`.

У разі якщо елемент повинен займати кілька комірок використовується атрибут `android: layout_span`.

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

Alternate Layouts – альтернативна верстка. Дозволяє використовувати різну верстку для різних орієнтацій екрану.

XML для альтернативної верстки поміщається в папки проекту:

- *res / layout-land* - альтернативна верстка для *landscape UI*.
- *res / layout-port* -альтернативная верстка для *portrait UI*.
- *res / layout-square* - альтернативна верстка для *square UI*.

В проекті "Менеджер фінансів" використовуються три стандартних типи верстки:

- *FrameLayout*;
- *LinearLayout*;
- *RelativeLayout*;

Розробка панелі навігації (рис. 3.4) зображений елемент *DrawerLayout* як список елементів. При виборі певного елемента відбувається зміна інтерфейсу головного Activity (рис. 3.5).

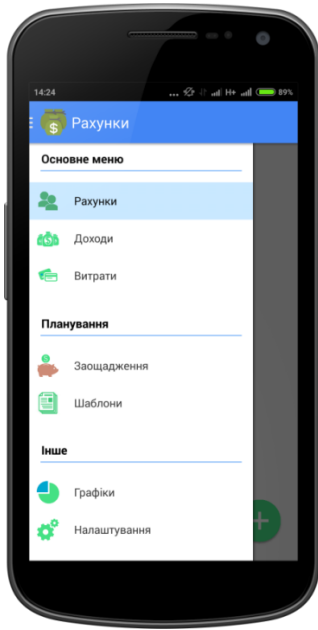


Рис. 3.4 Елемент *DrawerLayout*

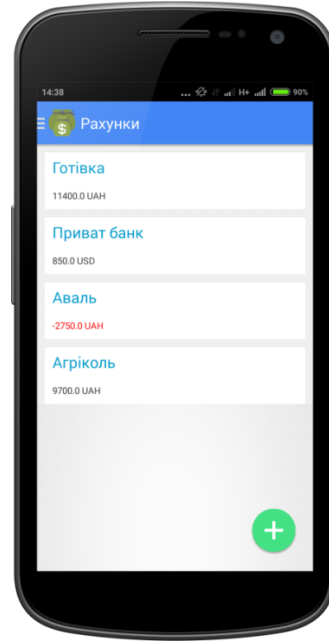


Рис. 3.5 Зміна інтерфейсу при натисканні на *DrawerLayout*

DrawerLayout входить до складу бібліотеки *Support Library v4* (release 13), який вийшов у травні 2013 року і використовується за умовчанням в проєктах, створених в ADT 22.0.

Створено стандартну розмітку, що складається з *FrameLayout* і *ScrollView*, і обертається додатково новим батьківським елементом `android.support.v4.widget.DrawerLayout`. На панелі інструментів даного компонента немає, тому код написано вручну.

```
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <FrameLayout
        android:id="@+id/content_frame_FL"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
```

```

<ScrollView
    android:id="@+id/lldrawercontent"
    android:layout_width="300dp"
    android:layout_height="fill_parent"
    android:layout_gravity="start"
    android:background="@android:color/background_light">
<!--Елементи навігації (див. в додаток) -->
</ScrollView>

```

Основний елемент для виведення інформації (FrameLayout) повинен бути першим дочірнім елементом DrawerLayout. Це пов'язано розташуванням елементів в Z-порядку.

Висота і ширина основного елемента повинна відповідати висоті і ширині батьківського елемента.

3.2.3. Фрагменти Activity для вікон додатку

Фрагменти були представлені в API 11, але в цілях сумісності була написана спеціальна бібліотека Android Support library для старих пристроїв. Таким чином зараз існують два класи Fragment для нових пристроїв і для старих пристроїв. Назви методів і класів дуже схожі, але не варто змішувати в одному проекті два різних класи.

Другий варіант використовує динамічне підключення фрагмента. Принцип наступний – в розмітку поміщається макет ViewGroup, який стає контейнером для фрагмента. Зазвичай, для цієї мети використовують FrameLayout, але це не обов'язкова умова. І в потрібний момент фрагмент заміщає контейнер і стає частиною розмітки. Саме цей варіант і використовується в додатку.

Приклад верстки фрагменту графічного інтерфейсу для рахунків в додатку "Менеджер фінансів":

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/RelativeLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

```

```
android:orientation="vertical" >
```

```
<ImageButton
```

```
    android:id="@+id/account_add_Button"  
    android:contentDescription="@string/about"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentRight="true"  
    android:background="@android:color/transparent"  
    android:paddingBottom="20dp"  
    android:paddingRight="20dp"  
    android:paddingLeft="20dp"  
    android:src="@drawable/btn_fab" />
```

```
<ListView
```

```
    android:id="@+id/account_list"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:divider="@null"  
    android:layout_above="@id/account_add_Button"  
    android:layout_alignParentStart="true"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentTop="true" >
```

```
</ListView>
```

```
</RelativeLayout>
```

В даному прикладі зображений фрагмент графічного ресурсу розмітки вікна. Використовується RelativeLayout в якості кореневого елемента, на якому розміщені:

- список рахунків (ListView);
- кнопка "додати рахунок" (ImageButton).

Якщо ввімкнути функцію перегляду графічного макету в середовищі AndroidStudio, то можна побачити сам макет інтерфейсу даного фрагменту "Рахунки" (рис. 3.6).

Для з'єднання цього фрагменту з основним вікном необхідно створити клас фрагменту та підключити його до контейнера основного вікна.

```
KleXFragment fragment;
```

```
FragmentManager fragmentManager = getFragmentManager();
```

```
fragment = AccountFragment.getInstance();
```

```
fragmentManager.beginTransaction()
```

```
.replace(R.id.content_frame_FL, fragment).commit();
```

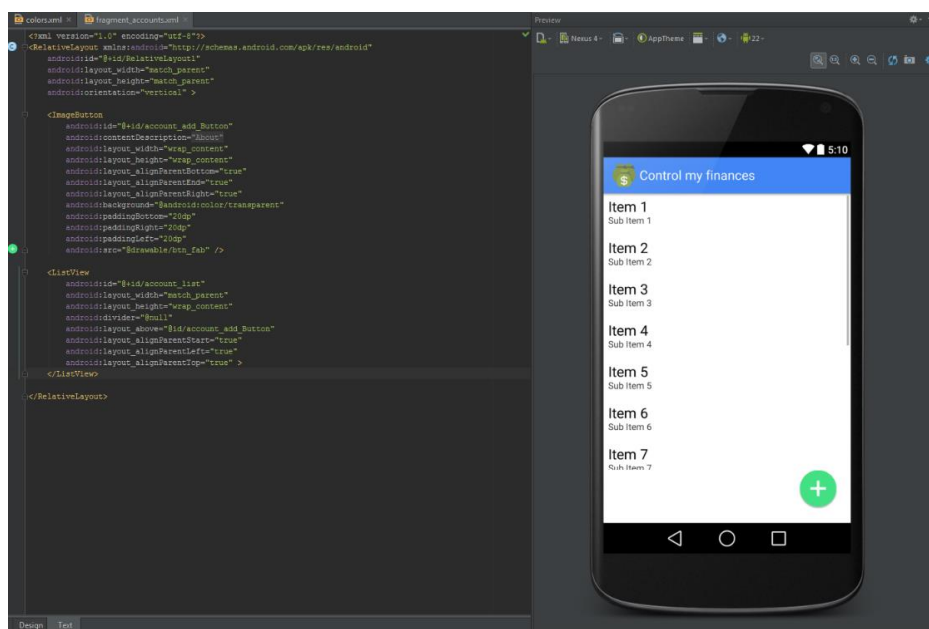


Рис. 3.6 Верстка фрагменту "Рахунки"

Аналогічним методом створені всі інші фрагменти користувацького інтерфейсу, такі як: "Витрати", "Доходи", "Заощадження", "Шаблони", "Статистика", "Налаштування".

3.2.4. Елементи списків в додатку

Елемент, який найчастіше зустрічається в додатку – це список (ListView). За допомогою нього відображається вся інформація про рахунки, витрати,

доходи тощо. Цей елемент має багато своїх особливостей. За замовчуванням він виглядає наступним чином (рис. 3.7):



Рис. 3.7 Віджет ListView

Для створення випадаючого списку, потрібно використати похідний клас від `ListView` – `ExpandableListView`. Цей клас дозволяє з'єднувати кожен елемент списку з ще одним списком. Таким чином створений список "Категорії". При натисканні на будь-який елемент цього списку, з нього випадає ще один список з витратами у цій категорії.

Для задання спеціальних стилів використовується власна розмітка окремого елемента списку. Для задання цієї розмітки для списку використовується певний адаптер – система Android налічує їх дуже багато (рис. 3.8).

Для створення списку категорій було вибрано адаптер *ExpandableListAdapter*.

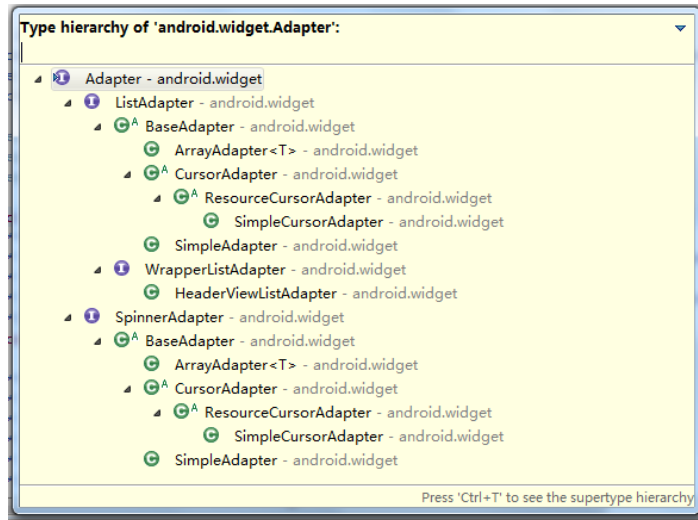


Рис. 3.8 Класи адаптерів для списку в ОС Android

Для всіх інших списків в додатку вибрано адаптер, який містить в собі масив з певними об'єктами, які необхідно відображати в елементі ListView – *android.widget.AdapterView*.

Наприклад, клас *ReservedsAdapter* унаслідуються від класу *ArrayAdapter<?>*:

```
public class ReservedsAdapter extends ArrayAdapter<Reserved>
```

Конструктор класу *ReservedsAdapter* приймає вказівник на активне вікно додатку (*Context*) та список об'єктів, необхідних для заповнення списку:

```
public ReservedsAdapter (Context context, List<Reserved> reserveds) {  
    super(context, R.layout.item_reserved_list, reserveds);  
    this.reserveds = reserveds;  
}
```

Для зв'язку деяких графічних елементів користувача з іншими (наприклад, діалогове вікно взаємодіє з Activity, або з'єднання кліку на елементі списку з певною дією) використано шаблон проектування "спостерігач" (рис. 3.9) – поведінковий шаблон проектування. Також відомий як "підлеглі" (*Dependents*), "видавець-передплатник" (*Publisher-Subscriber*). Створює механізм у класі, який дозволяє отримувати екземпляру об'єкта цього класу оповіщення від інших об'єктів про зміну їх стану, тим самим спостерігаючи за ними.

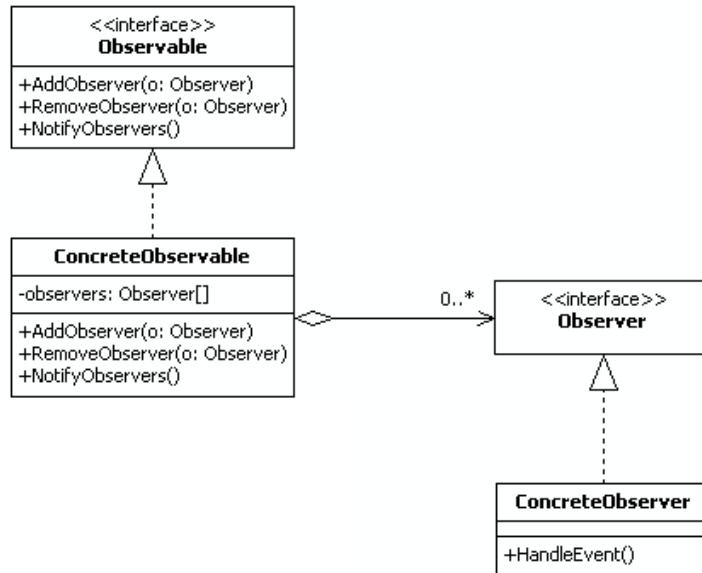


Рис. 3.9 Діаграма інтерфейсу "Спостерігач"

Він визначає залежність типу "один до багатьох" між об'єктами таким чином, що при зміні стану одного об'єкта, всі залежні від нього сповіщаються про цю подію.

При реалізації шаблону "спостерігач" зазвичай використовуються наступні класи:

- Observable – інтерфейс, що визначає методи для додавання, видалення та оповіщення спостерігачів;
- Observer – інтерфейс, за допомогою якого спостерігач отримує сповіщення;
- ConcreteObservable – конкретний клас, який реалізує інтерфейс Observable;
- ConcreteObserver – конкретний клас, який реалізує інтерфейс Observer.

Шаблон "спостерігач" застосовується в тих випадках, коли система має такі властивості:

- існує, як мінімум, один об'єкт, що розсилає повідомлення;
- є не менше одного одержувача повідомлень, причому їх кількість і склад можуть змінюватися під час роботи програми;
- немає потреби дуже сильно пов'язувати взаємодіючі об'єкти, що корисно для повторного використання.

Даний шаблон часто застосовують в ситуаціях, в яких відправника повідомлень не цікавить, що роблять одержувачі з наданої їм інформацією.

3.3. Керівництво користувача

Додаток "Менеджер фінансів" можна встановити двома способами:

1. Скачавши його з Google Play Market.
2. Встановивши його вручну з пам'яті мобільного телефону чи планшету, попередньо скопіювавши файл програми в пам'ять пристрою.

Файл додатку має розширення "*.apk" – це формат архівних файлів-додатків для ОС Android. Додаток "Менеджер фінансів" скомпільовано і упаковано в один файл, який включає в себе весь код програми (.DEX) файли), ресурси, активи і файл .manifest.

При першому запуску програми потрібно створити список власних рахунків, натиснувши на кнопку "+". Потрібно лише ввести назву рахунку та валюту (рис. 3.10), в якій відкрито рахунок та натиснути кнопку "Далі". Після створення потрібних рахунків потрібно додати список доходів, щоб рахунки були заповнені.

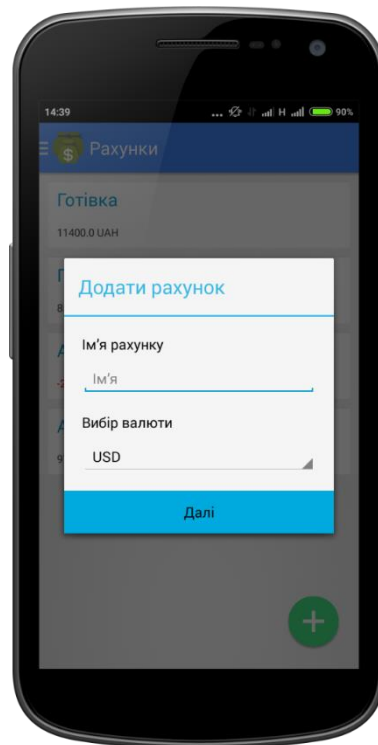


Рис. 3.10 Додавання рахунку

Для додавання списку доходів потрібно перейти в меню "Доходи" (рис. 3.11) та натиснути кнопку "+". Після цього з'являється діалогове вікно з вибором рахунку до якого належить цей дохід. Наступне вікно слугує для вводу параметрів доходу (рис. 3.12). Обов'язково потрібно заповнити лише поле "сума". Всі інші є опціональними.

Якщо не вибирати в цьому вікні дату для доходу, то вона автоматично запишеться як сьогоднішня. Також передбачено, що користувач може залишити замітку про дохід.

Коли всі дані заповнено потрібно натиснути кнопку "Далі". Після цього програма повернеться у вікно "Рахунки" та знову відобразить список усіх рахунків разом із тільки що доданим.

Якщо доходи вже заповнені і відображають всю фінансову картину користувача, то потрібно починати вести облік власних витрат.

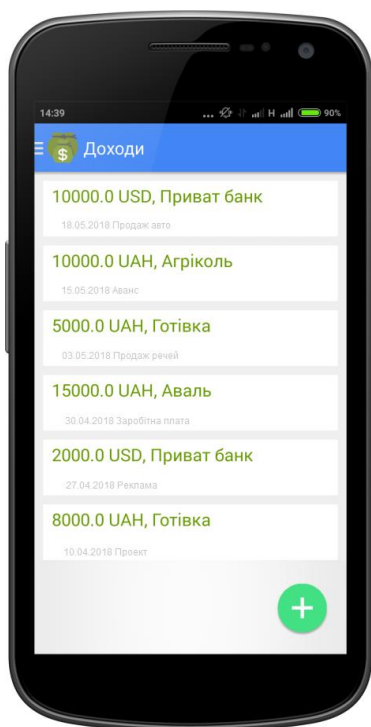


Рис. 3.11 Вікно "Доходи"

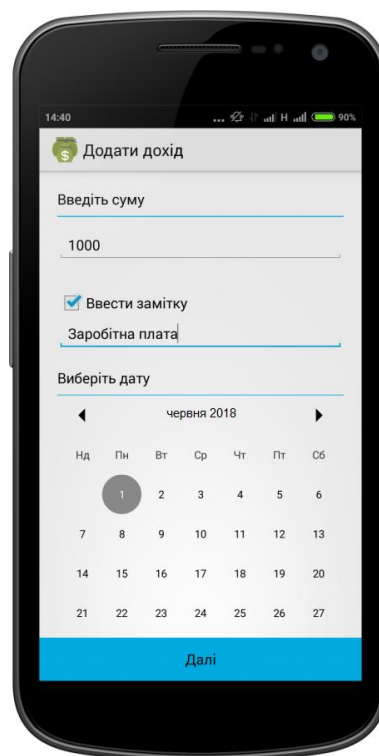


Рис. 3.12 Додавання доходу

Для додавання витрат потрібно перейти в розділ "Витрати" в боковому меню програми (рис. 3.13).

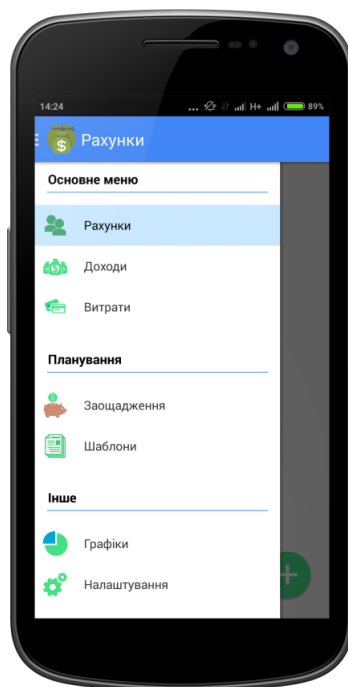


Рис. 3.13 Бокове меню додатку

Щоб створити список витрат потрібно додати категорію. Потрібно в вікні "Витрати" (рис. 3.14) натиснути кнопку "+" та вибрати пункт меню "Категорія", і відкриється вікно додавання категорії (рис. 3.15).

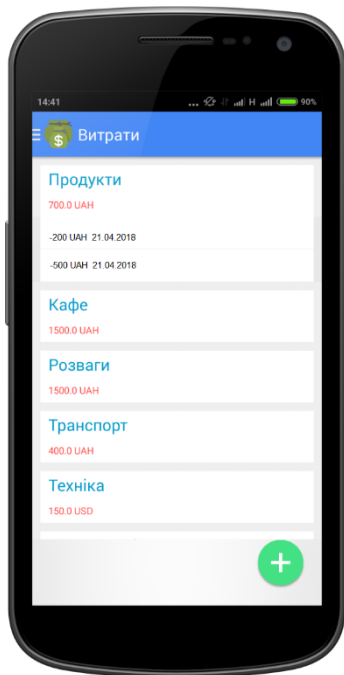


Рис. 3.14 Вікно "Витрати"

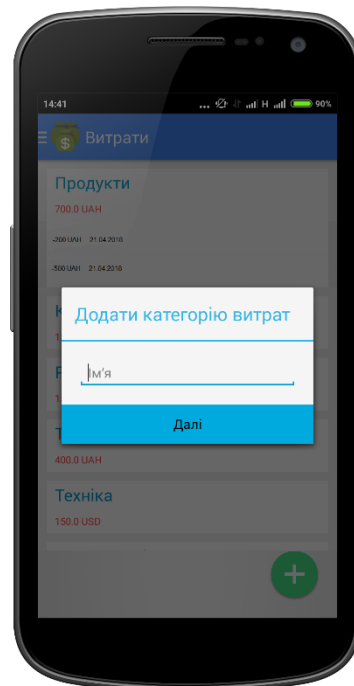


Рис. 3.15 Додавання категорії витрат

Після додавання необхідних категорій потрібно додати самі витрати. Для цього потрібно в вікні "Витрати" натиснути кнопку "+" та вибрати пункт меню "Витрата", і відкриється вікно вибору категорії. Після цього пропонується вибрати рахунок. Після вибору рахунку відкриється вікно з параметрами для створення витрати (рис. 3.16).

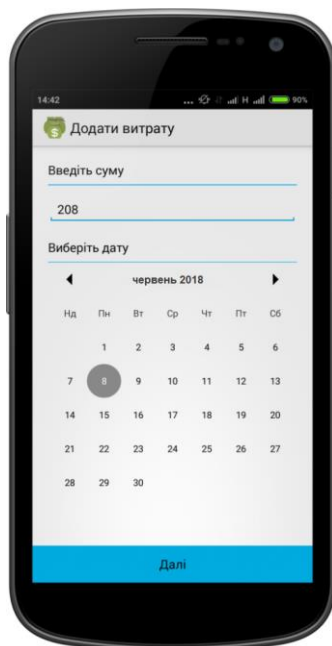


Рис. 3.16 Вікно додавання витрати

Після цих дій можна додати заощадження. Наприклад, в користувача в планах є купити новий будинок. Він вносить необхідну суму для заощадження, і по мірі надходження коштів на його рахунки, він може пересилати їх на це заощадження. Вікно "Заощадження" зображено на (рис. 3.17). Для додавання заощадження потрібно натиснути кнопку "+". Після цього відкриється вікно для вводу даних про заощадження (рис. 3.18).



Рис. 3.17 Вікно "Заощадження"

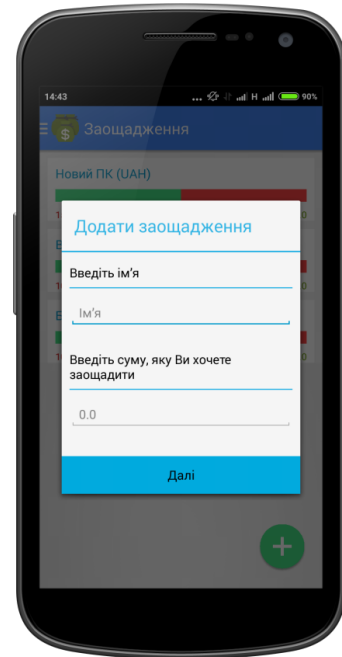


Рис. 3.18 Додавання заощадження

Також присутнє меню шаблонів (рис. 3.19). При додаванні шаблону потрібно натиснути кнопку "+". І в діалоговому меню вибрати тип шаблону – "Витрата" або "Дохід". При створенні шаблону витрати чи доходу використовується та ж інструкція, що й при додаванні звичайної витрати або доходу. Щоб додати елемент з шаблону в реальний список потрібно натиснути кнопку "+" справа від елементу списка.

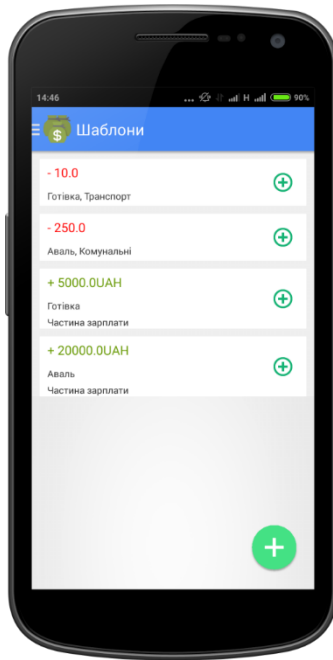


Рис. 3.19 Вікно "Шаблини"

Передостанній елемент меню – це розділ "Статистика" (рис. 3.20). В ній відображається діаграма витрат. Також передбачена можливість фільтру витрат за певний період (рис.3.21).

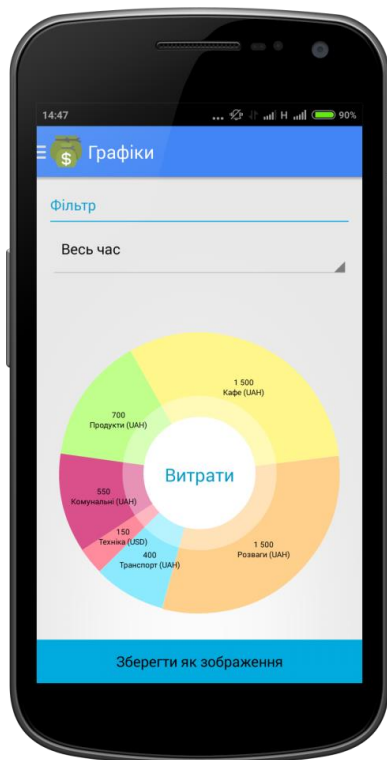


Рис. 3.20 Вікно "Статистика"

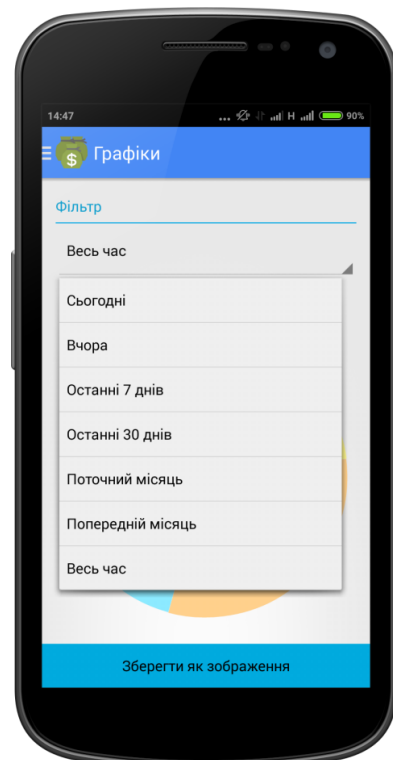


Рис. 3.21 Фільтр витрат

Останній розділ програми – "Налаштування" (рис. 3.22). В ньому є можливість експорту та імпорту даних, вибір мови. Є можливість надіслати власні дані в мережу Інтернет та вибір валют, якими користувач користується. Вибір валют створено для зручності в користуванні додатком, щоб валют, які не використовуються не було в списку.

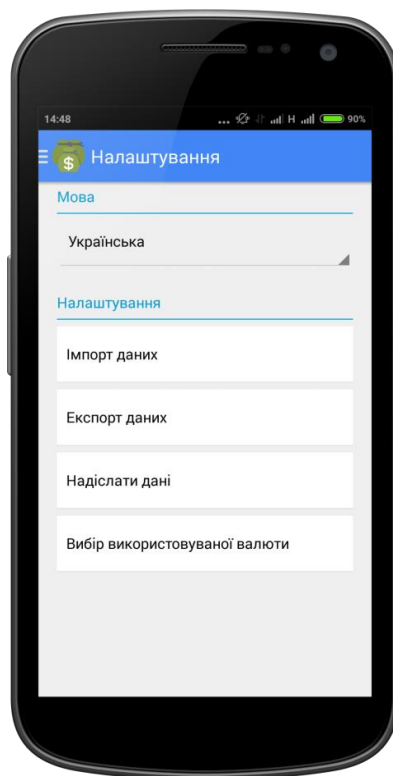


Рис. 3.22 Вікно "Налаштування"

Висновки до розділу

Розроблена структура бази даних. За її допомогою проведена розробка самої бази даних в програмному кодї додатка.

Також розроблені класи для маніпуляції з даними, які збережені в базі даних. За допомогою розширеної мови розмітки проведено верстку графічного інтерфейсу додатку. Створено усі списки, кнопки та текстові поля з відповідними відступами та розміщенням відносно границь екрану.

Наступним кроком проведено з'єднання графічного інтерфейсу з функціональною частиною. Написані класи адаптерів для наповнення списків, створені класи фрагментів тощо.

ВИСНОВКИ

При виконанні дипломного проекту було розроблено автоматизовану систему для керування особистими доходами та витратами та для організації власного бюджету в мобільному телефоні або планшеті на операційній системі *Android* – додаток “Менеджер фінансів”.

Досліджено і проаналізовані існуючі мобільні застосунки для ведення власного бюджету. Проаналізовано найпоширеніші додатки з магазину додатків *Google Play Market*. Після цього аналізу виявлено, що всі програми даної тематики мають ряд схожих недоліків:

- в більшості додатків або занадто складний і перенавантажений інтерфейс, або ж, навпаки, немає дуже необхідних функцій;
- майже у всіх додатках відсутня український переклад;
- стиль графічного інтерфейсу застарілий;
- незручність в користуванні;
- немає постійної підтримки та оновлень версій зі сторони розробників.

Обрана категорія користувачів додатком – це середньостатистична людина, яка не вміє власноручно вести свій облік фінансів. Цей проект буде цікавий також і тим, хто вже давно займається веденням власного обліку фінансів. Це досягається тим, що за допомогою "Менеджера фінансів" можна спростити саму процедуру.

Функціональна частина написана за допомогою мови програмування *Java*, а верстка графічного дизайну – за допомогою розширеної мови розмітки *XML*.

Також проаналізовано мультимедійні технології, які використовувалися на етапі розробки мобільного додатку – це графіки, діаграми, звук та графіка, *Activity*, діалоги та меню, анімація.

Розроблено загальну схему додатку "Менеджер фінансів", розроблено його дизайн та ергономіку.

Розроблена автоматизована система може бути встановленою на мобільний телефон чи планшет під ОС *Android* та використаною будь-яким користувачем для обліку особистих фінансів.