

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ
ФАКУЛЬТЕТ ЕКОНОМІЧНОЇ ІНФОРМАТИКИ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ**

Пояснювальна записка

до дипломного проекту

бакалавра

на тему: «Розроблення модуля обліку особистих фінансів»

Виконав: студент 4 курсу,
групи 6.04.51.16.01,
спеціальності 122

«Комп'ютерні науки та
інформаційні технології»

Ширков Андрій Олегович

Керівник:

к.е.н., доц.

Беседовський Олексій
Миколайович

Харків – 2020 рік

РЕФЕРАТ

Пояснювальна записка до дипломного проєкту: 67 с., 34 рис., 23 табл., 6 додатків, 31 джерело.

Об'єктами проєктування є функціональні елементи, архітектура, інформаційне і програмне забезпечення модуля обліку особистих фінансів ФОП «Шнуренко Дмитро Геннадійович».

Мета проєктування – створення модуля "Облік особистих фінансів".

Метод проєктування – використання програмних систем ARIS Toolset, Erwin, Ramus Education, Android Studio.

Створений модуль дозволяє спростити процес обліку особистих фінансів користувача.

Результати розробки можуть бути впроваджені в торговому майданчику мобільних додатків Google Play Market.

СИСТЕМА УПРАВЛІННЯ ФІНАНСАМИ, ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОЄКТУВАННЯ, CASE-ДІАГРАМИ, БАЗА ДАНИХ, БЮДЖЕТ, МОБІЛЬНИЙ ДОДАТОК, РОЗПОДІЛ БЮДЖЕТУ, ОПЕРАЦІЙНА СИСТЕМА ANDROID.

ABSTRACT

The bachelor's thesis report: 67 pages, 34 figures, 23 tables, 6 appendices, 31 sources.

The objects of design are functional elements, architecture, information and software of the personal finance accounting module.

The purpose of the design is to create a module "Accounting of personal finances".

Design method - use of software systems ARIS Toolset, Erwin, Ramus Education, Android Studio.

The created module allows to simplify process of the accounting of personal finances of the user.

The results of the development can be implemented in the trading market of mobile applications Google Play Market.

FINANCIAL MANAGEMENT SYSTEM, OBJECT-ORIENTED DESIGN, CASE-DIAGRAMS, DATABASE, BUDGET, MOBILE APP, DISTRIBUTION ANDROID.

ЗМІСТ

1	АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ «ОБЛІК ОСОБИСТИХ ФІНАНСІВ»	8
1.1	Коротка характеристика об'єкту управління «ФОП Шнуренко Дмитро Геннадійович»	8
1.2	Опис предметної області «Облік особистих фінансів».....	9
1.3	Огляд і аналіз існуючих аналогів, що реалізують функції предметної області	14
2	СПЕЦИФІКАЦІЯ ВИМОГ ДО МОДУЛЯ.....	20
2.1	Глосарій.....	20
2.2	Розроблення варіантів використання.....	21
2.2.1	Діаграма варіантів використання	21
2.2.2	Специфікація варіантів використання	22
2.3	Специфікація функціональних та нефункціональних вимог	24
2.4	Проектування інтерфейсу користувача	26
3	ПРОЄКТНІ ТА ТЕХНІЧНІ РІШЕННЯ	31
3.1	Математична та логічна постановка	31
3.2	Проектування структури бази даних	32
3.2.1	Концептуальне інфологічне проектування	32
3.2.2	Проектування логічної моделі даних.....	34
3.2.3	Проектування фізичної моделі даних	36
3.3	Проектування програмного забезпечення.....	36
3.4	Тестування програмної системи.....	37
3.5	Розгортання програмного продукту.....	45
	ВИСНОВКИ.....	46
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47
	Додаток А Код SQL запитів для бази даних	51
	Додаток Б UML- діаграма класів.....	55
	Додаток В Клас IncomeActivity.....	56
	Додаток Г Клас IncomeAdapter	61
	Додаток Д Клас Income.....	62
	Додаток Е Інтерфейс користувача.....	63

ВСТУП

На сьогоднішній день проблема контролю особистого бюджету є дуже актуальною.

Що відрізняє успішну людину? Вона завжди впевнена у собі та своєму достатку, тому що такі люди дуже прискіпливо ставляться до власних коштів. Вони дуже ретельно відстежують потік грошей: дохід та витрати. Окрім цього відстежуються також і сфери, куди ці гроші використовуються, щоб усі сфери життєдіяльності людини були профінансовані.

Процес відстеження бюджету може бути дуже складним та віднімати багато часу. Але зараз із цим не має бути проблем. Програма дипломного проєкту буде допомагати користувачеві заощаджувати гроші та час. Все, що потрібно від користувача, це вносити дохід та витрати, а все інше буде виконувати програма: розподіляти гроші за сферами, відстежувати виконання плану витрат, якщо потрібно, давати підказки з приводу розпорядження бюджетом та давати звіти за відповідні дати з доходу та витратами за сферами.

Під час написання дипломного проєкту мають бути реалізовані наступні вимоги: навести коротку характеристику об'єкту управління «ФОП Шнуренко Дмитро Геннадійович»; навести опис предметної області «Облік особистих фінансів»; провести огляд і аналіз існуючих аналогів, що реалізують функції предметної області; розробити глосарій термінів; розробити діаграму варіантів використання; навести специфікацію варіантів використання; розробити специфікацію функціональних та нефункціональних вимог; спроектувати інтерфейс користувача; навести математичну та логічну постанову завдання; спроектувати базу даних (інфологічне проєктування, фізична на логічна модель); спроектувати програмний продукт; провести тестування програмного продукту; привести розгортання програми.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ «ОБЛІК ОСОБИСТИХ ФІНАНСІВ»

1.1 Коротка характеристика об'єкту управління «ФОП Шнуренко Дмитро Геннадійович»

Організація ФОП «Шнуренко Дмитро Геннадійович» займається розробкою програмного забезпечення (веб та мобільних додатків) на замовлення українських та іноземних партнерів.

Розробка програмного забезпечення, як і в усіх подібних ІТ компаніях відбувається у декілька етапів:

- Детальне визначення потреб замовника від подальшого продукту
- Розподілення обов'язків на команду, яка бере участь у розробці
- Створення дизайну
- Розробка програмного продукту
- Тестування

Схема організаційної структури підприємства наведена на рис. 1.1.

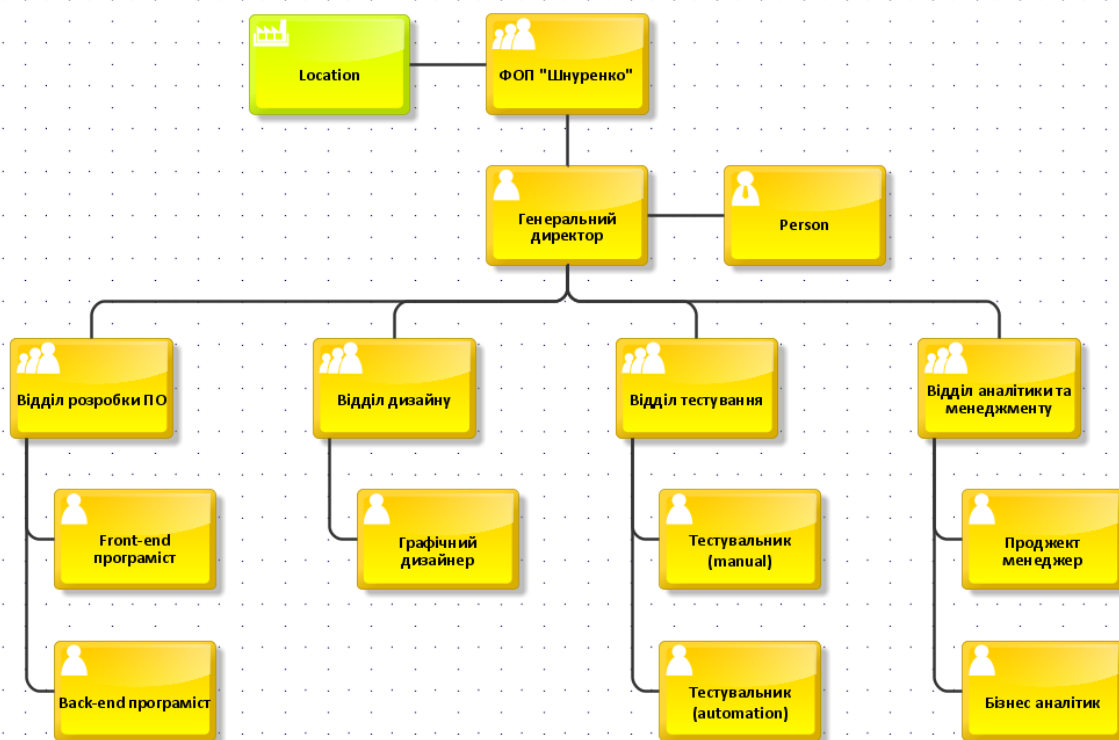


Рисунок 1.1 – Схема організаційної структури управління підприємством «ФОП Шнуренко Дмитро Геннадійович»

Підприємство «ФОП Шнуренко Дмитро Геннадійович» має наступну організаційну структуру: головну роль в організації виконує генеральний

директор, який керує відділами розробки, дизайну, тестування та аналітики і менеджменту.

Відділ розробки займається написанням програмного забезпечення і складається з команди Front-end програмістів та команди Back-end програмістів.

Відділ дизайну займається створенням мокапу майбутнього продукту для замовника та розробкою унікальних графічних рішень продукту. Складається з команди графічних дизайнерів.

Відділ тестування відповідальний за розробку тест кейсів та тестуванням продукту на всіх етапах його розробки. Складається з команди тестувальників (manual та automation).

Відділ аналітики та менеджменту займається комунікацією із замовником, визначенням та аналізом його детальних потреб від продукту та розподіленням роботи між відділами в залежності від визначених потреб замовника. До складу відділу входять бізнес аналітик та проджект менеджер.

Наразі проблему бізнесу складає розробка програмного продукту (мобільний додаток), який міг би допомогти користувачам вести облік особистих фінансів.

1.2 Опис предметної області «Облік особистих фінансів»

Програмний продукт у вигляді мобільного додатку перш за все повинен автоматизувати процес аналізу та обліку особистих фінансів користувача.

В основу додатку лягла економічна модель розподілення бюджету «Метод кувшинів». Її сенс у тому, що кожного місяця бюджет користувача розподіляється на категорії за сферами життя людини.

Перша та найбільша категорія – витрати на найнеобхідніше або щоденні (регулярні) витрати. До неї входить наступне: сплата комунальних послуг, аренди, виплати кредиту, харчування, сплата за проїзд, одягу, медичні послуги та гігієнічні товари. На ці потреби виділяється 60% усього бюджету.

Друга категорія це розваги. Сюди включають кіно, кафе, ресторани, театри, масаж, похід у парк та інше. На розваги виділяють 10% бюджету.

Третя категорія – освіта. З цих грошей ви можете придбати курси, книжки, абонемент у спортзалу, сплатити за навчання, загалом, те, що допоможе вам покращити свої вміння та навички. Складає 5% бюджету.

Четверта категорія це резерв. Його можна витратити на великі покупки, такі як нова мебелю або побутова техніка, поїздка на відпочинок або ремонт автомобіля. На це виділяють 10% бюджету.

П'ята категорія – це подарунки та благодійність – 5% бюджету.

Та остання категорія – це довгострочні накопичення – 10% бюджету.

Ця економічна модель забезпечує рівномірне забезпечення фінансами усі необхідні сфери життя звичайної людини та призначена для людей усіх поколінь та професій. Ефективність методу підтверджена шляхом тестування методу розробником мобільного додатку протягом декількох років.

Найзручніше описати роботу додатку за допомогою *idef0*. Це методологія функціонального моделювання і графічного описання процесів, призначена для формалізації і опису бізнес-процесів. Особливістю IDEF0 є її акцент на ієрархічне представлення об'єктів, що значно полегшує розуміння предметної області. В IDEF0 розглядаються логічні зв'язки між роботами, а не послідовність їх виконання в часі.

На рис. 1.2 відображений процес обліку особистих фінансів за допомогою контекстної діаграми. На вхід отримуємо інформацію про витрати, інформацію про доходи та список категорій. Процес керується економічною моделлю розподілу фінансів за категоріями. Процес виконується користувачем та інформаційною системою, та на виході маємо звіт про стан фінансів за обраним періодом.

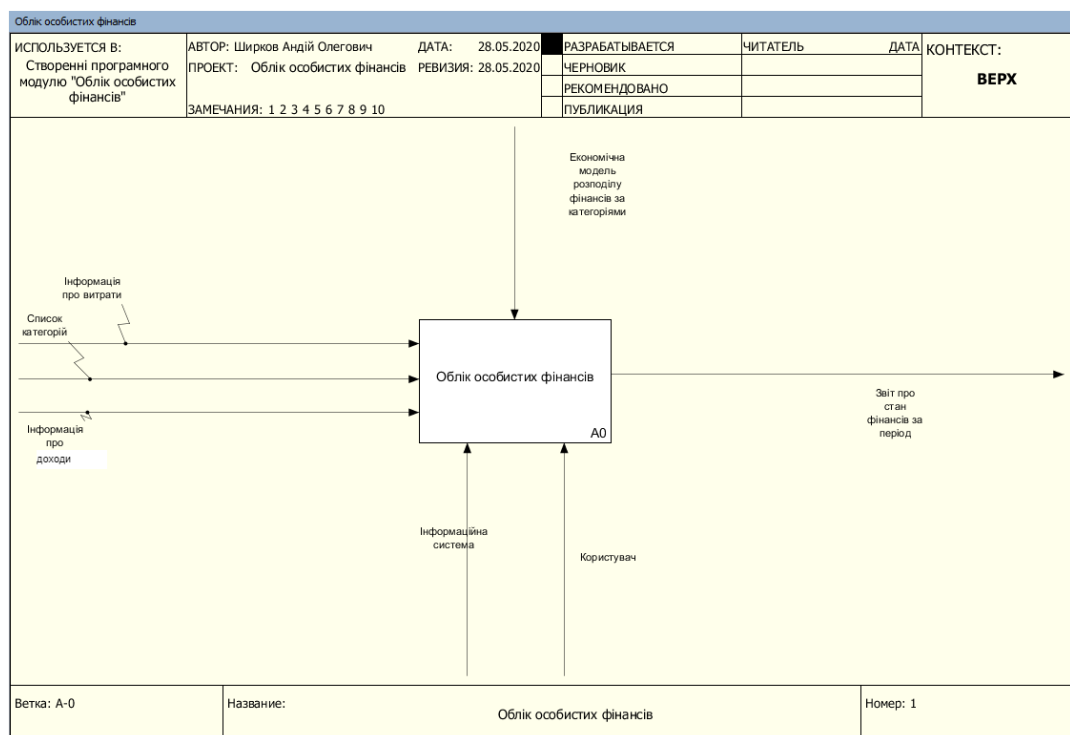


Рисунок 1.2 – Контекстна діаграми процесу

Оскільки процес обліку особистих фінансів складний, для кращого розуміння етапів його виконання маємо декомпозувати контекстну діаграму (рис. 1.3).

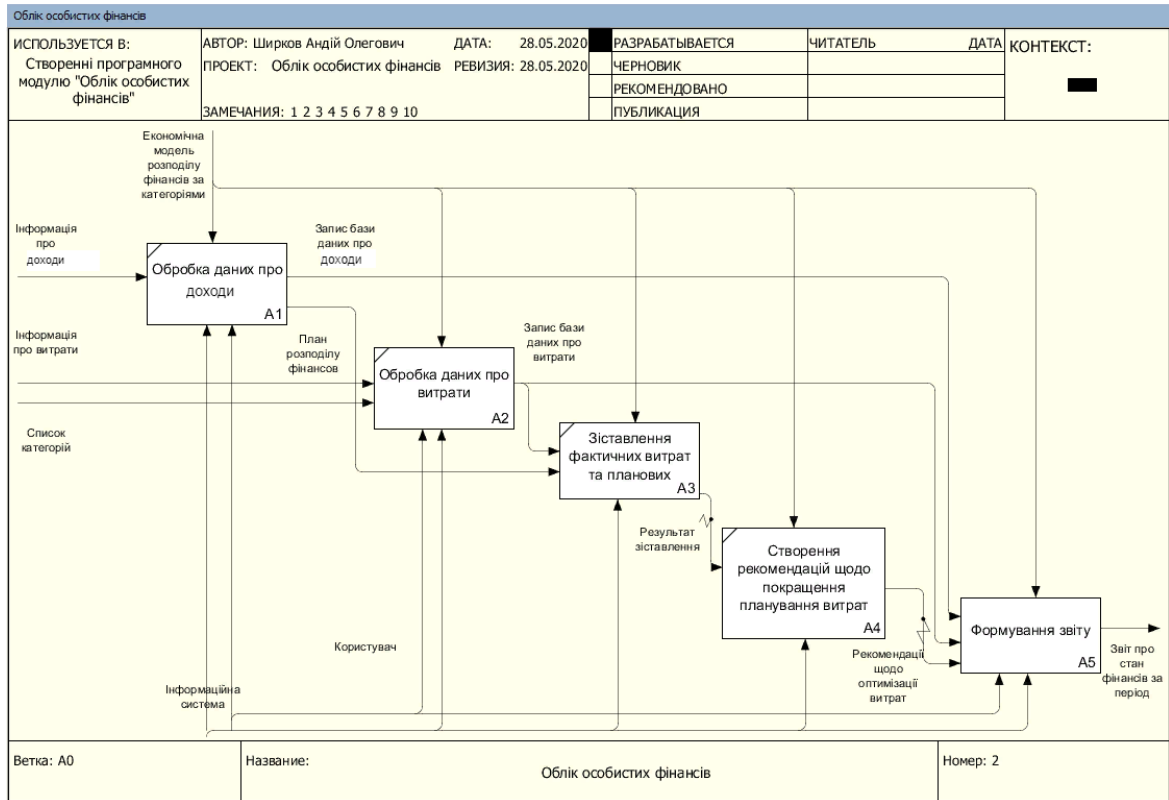


Рисунок 1.3 – Декомпозиція контекстної діаграми

Декомпозиція контекстної діаграми містить в собі наступні процеси: обробка даних про прибуток, обробка даних про витрати, зіставлення фактичних витрат та запланованих, створення рекомендацій щодо покращення планування витрат та формування звіту. Всі процеси керуються економічною моделлю розподілу фінансів за категоріями.

Процес «Обробка даних про доходи» на вхід отримує інформацію про доходи, виконується інформаційною системою та користувачем. На виході має запис у базі даних про доходи та план розподілу фінансів.

Процес «Обробка даних про витрати» на вхід отримує список категорій, інформацію про витрати. Процес виконують користувач та інформаційна система, на виході має запис у базі даних про витрати.

Процес «Зіставлення фактичних витрат та запланованих» на вхід отримує запис бази даних про витрати та план розподілу фінансів, на виході - результат зіставлення. Виконується інформаційною системою.

Процес «Створення рекомендацій щодо покращення планування витрат» отримує на вхід результат зіставлення з попереднього процесу, на виході має рекомендації щодо оптимізації витрат. Виконується інформаційною системою.

Процес «Формування звіту» на вхід отримує записи бази даних про витрати та доходи, а також рекомендації щодо оптимізації витрат. Виконується користувачем та інформаційною системою. На вихід подає звіт про стан фінансів за період.

Функції які повинен забезпечувати продукт згідно описанню предметної області наведено у таблиці 1.1.

Таблиця 1.1 – Перелік функцій бізнес процесу

Функція	Описання
Розрахунок плану витрат за системою «кувшини»	Розробка плану витрат, розподіляючи доходи на 5 категорій витрат
Реалізувати можливість внесення даних про доходи	Можливість користувача створювати записи з інформацією про доходи
Реалізувати можливість внесення даних про витрати	Можливість користувача створювати записи з інформацією про витрати
Реалізувати можливість видалення даних про доходи	Можливість користувача видалити записи з інформацією про доходи
Реалізувати можливість видалення даних про витрати	Можливість користувача видалити записи з інформацією про витрати
Реалізувати можливість створення звіту	Можливість користувача сформувати звіт за період
Реалізувати можливість видалення звіту	Можливість користувача видалити запис з інформацією про звіт
Реалізувати можливість збереження звіту на пристрій	Можливість користувача завантажити текстовий файл зі звітом про стан фінансів за період
Реалізувати можливість перегляду історії доходів	Можливість користувача переглянути історію записів про доходи
Реалізувати можливість перегляду історії витрат	Можливість користувача переглянути історію записів про витрати
Реалізувати можливість перегляду історії звітів	Можливість користувача переглянути історію записів про звіти
Реалізувати можливість перегляду історії витрат за категорією	Можливість користувача переглянути історію записів про витрати за окремими категоріями

На рис. 1.4 відображено процес обліку особистих фінансів за допомогою case-засобів. На вхід процес отримує інформацію про доходи та витрати користувача, які потрібно ввести до обліку. Ця інформація обробляється програмою, розподіленням витрат за різними категоріями та розрахуванням

плану обліку фінансів за формулами. Після обробки можна налаштувати параметри для отримання звіту про стан фінансів за період за обробленою інформацією та завантажити звіт.

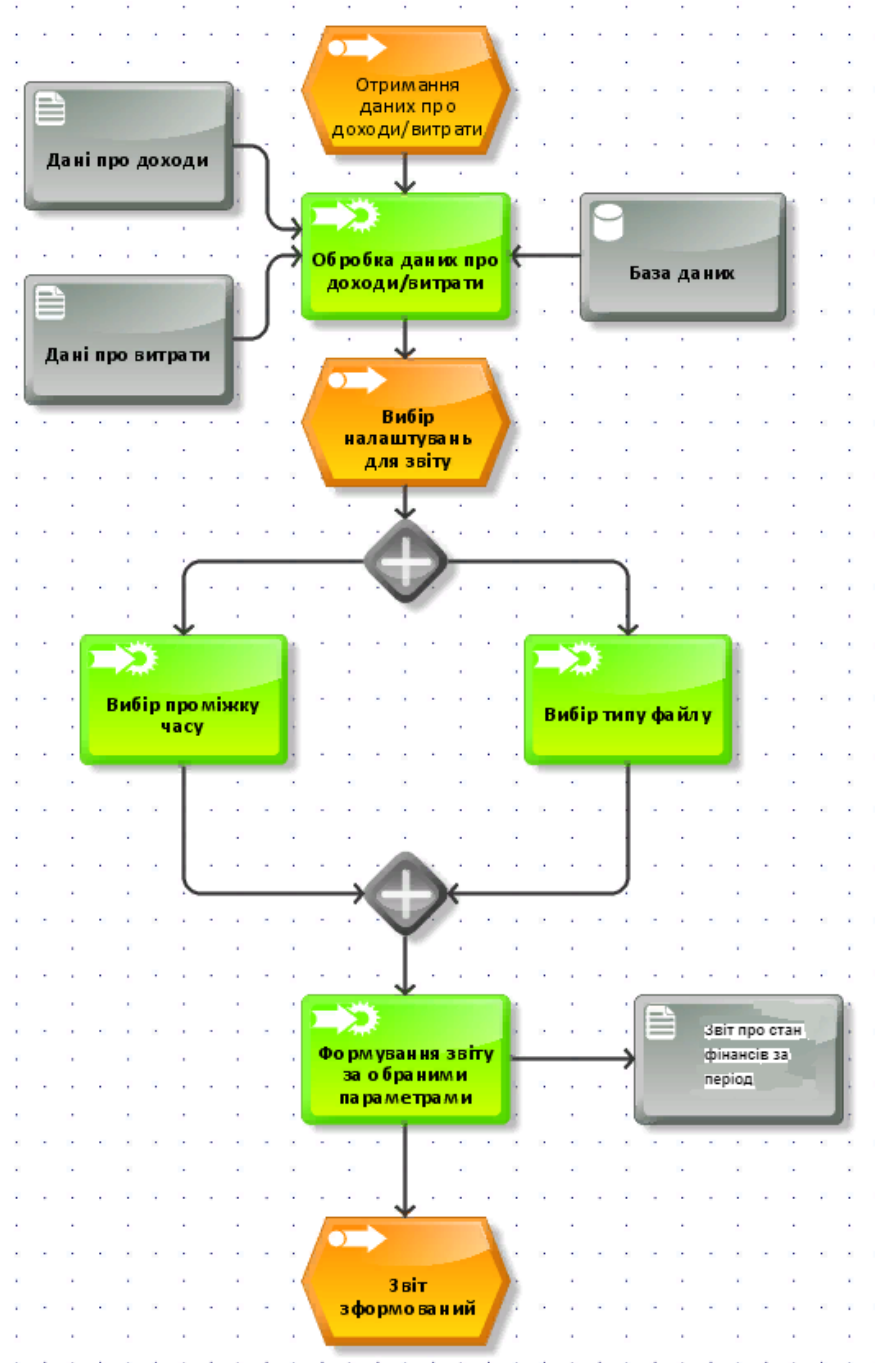


Рисунок 1.4 – Модель предметної області

У таблиці 1.2 наведено характеристику бізнес-процесу «Облік особистих фінансів»

Таблиця 1.2 – Характеристика бізнес-процесу

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Облік особистих фінансів
Основні учасники	Користувач
Вхідна подія	Отримання даних про доходи / витрати
Вхідні документи	Інформація про доходи, інформація про витрати, список категорій
Вихідна подія	Отримання звіту про стан фінансів за період
Вихідні документи	Звіт про стан фінансів за період
Клієнт бізнес-процесу	Користувач

1.3 Огляд і аналіз існуючих аналогів, що реалізують функції предметної області

Перед розробкою програмного продукту слід провести огляд та аналіз аналогів (таблиця 1.3), щоб уникнути можливих помилок під час розробки та виявити функціональні потреби, які повинна реалізувати майбутня програма.

Таблиця 1.3 – Порівняльна характеристика програмних продуктів

Фірма-розробник	«CB Mobile Ltd»	«Aimbity AS»
Назва програмного продукту	Monefy	Moneon
Версії продукту	1.3.3	6.0.3
Функціональність	Контроль витрат, створення звітів, підтримка декількох гаманців, синхронізація з хмарним сховищем	Контроль доходів та витрат, підтримка декількох гаманців та бюджетів, можливість створити власну категорію витрат, оформлення звітів
Інтерфейс користувача	Інформативний, але не є інтуїтивно зрозумілим	Інформативний, містить декілька вкладинок, що полегшує користування
Допомога користувачу	Є підказки, рекомендації до використання	Існує процес «знайомлення» з програмою, тех-підтримка розробників

Об'єктами дослідження конкурентних продуктів були два мобільних додатки: «Monefy» та «Moneon».

Monefy – відносно новий додаток, котрий функціонально відповідає потребам описаними у проблематиці предметної області.

Головне меню програми (рис. 1.5) досить просте та інформативне, тут можна побачити загальний стан фінансів (порівняння доходів та витрат) та кнопки внизу для додавання доходів та витрат. Проте інтерфейс є не зовсім інтуїтивно зрозумілим, оскільки тут є багато кнопок навколо діаграми, і не зрозуміло для чого вони призначені та за що відповідає кожна кнопка.

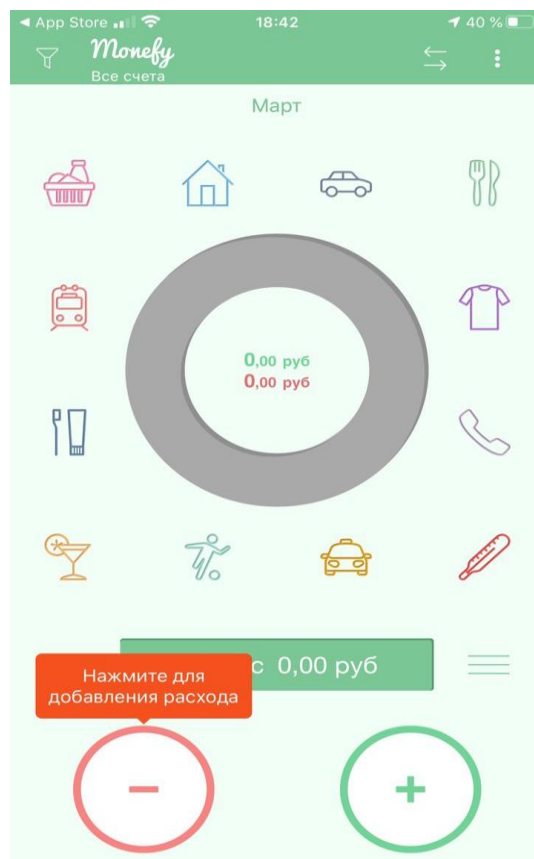


Рисунок 1.5 – Головне вікно програми Monefy

Вікна додавання записів (рис. 1.6) мають мінімалістичний дизайн, як і вся програма, та є інтуїтивно зрозумілими для користування. Тут можна ввести новий запис, обравши суму для вводу, дату (за замовчуванням – сьогоднішня) та категорію. Також можна додати короткий опис запису.

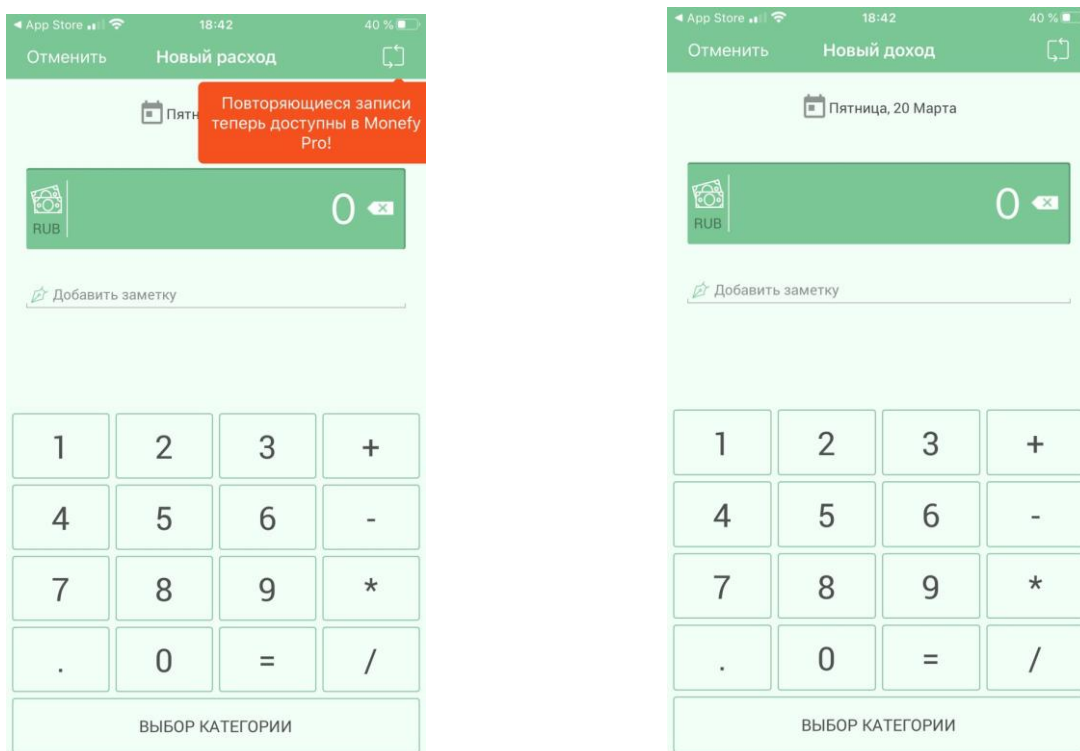


Рисунок 1.6 – Вікна додавання витрат та доходів

Також є декілька інформативних вікон, таких як перелік категорій, рахунків та статистика за обраним періодом (рис. 1.7).



Рисунок 1.7 – Інформативне вікно з категоріями

Даний програмний продукт є досить вдалим, але зі своїми недоліками, такими як не завжди зрозумілий дизайн (допомагають тільки підказки) та неможливість отримати звіт, що є основною потребою при обліку фінансів.

Moneop – мобільний додаток, який все досить давно існує на ринку та є найбільш вдалим із подібний йому продуктів. Має майже весь потрібний функціонал для обліку.

Цей програмний продукт не має заздалегідь визначеної головної сторінки, оскільки користувач може у налаштуваннях вибрати вікно, яке буде зустрічати його при вході у програму. За замовчуванням це вікно «Сводка», де відображений обраний користувачем гаманець для доходів та процес витрат по цьому гаманцю (рис. 1.8).

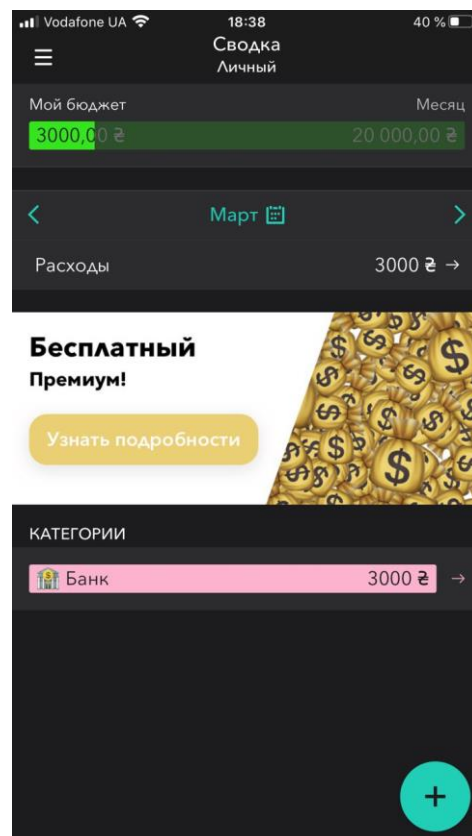


Рисунок 1.8 – Головне вікно «сводка»

Також серед вікон програми є «бюджети» та «транзакції», у котрих відображаються відповідні дані та де можна додати новий запис до цих вікон (рис. 1.9).

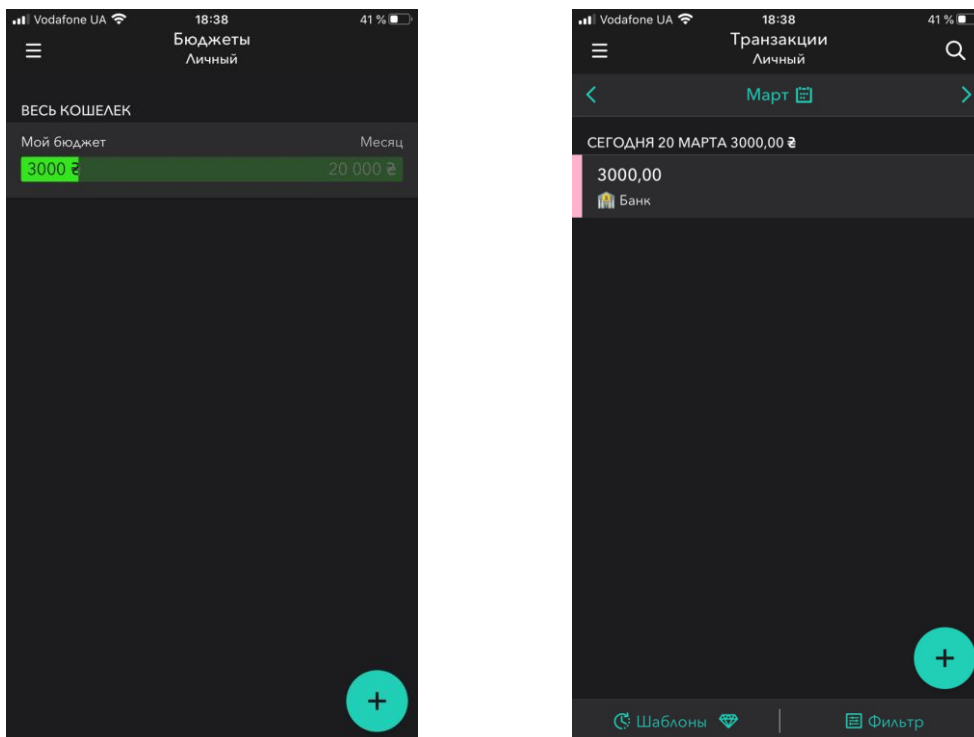


Рисунок 1.9 – Вікна «бюджети» та «транзакції»

Вікна із додаванням нових записів (рис. 1.10) є зручними та інтуїтивно зрозумілими. Тут можна обрати необхідну суму для вводу, категорію, за якою буде записано витрати (категорій багато та можна додати власну), дату коли була зроблена витрата, додати короткий опис запису та фотографію чека, котрий буде зберігатися у додатку.

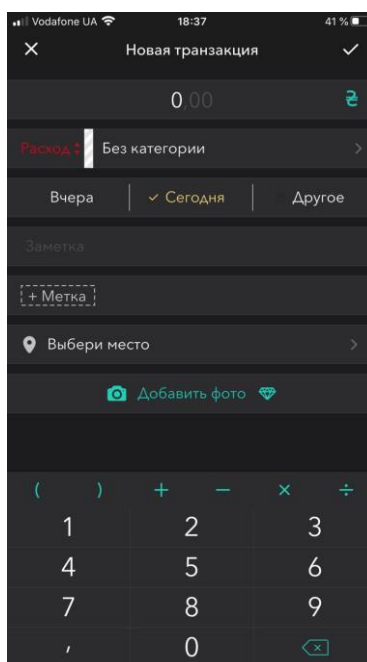


Рисунок 1.10 – Вікно додавання витрат

Також тут є дуже важлива функція отримання звіту за обраним періодом та типом файлу (рис. 1.11).

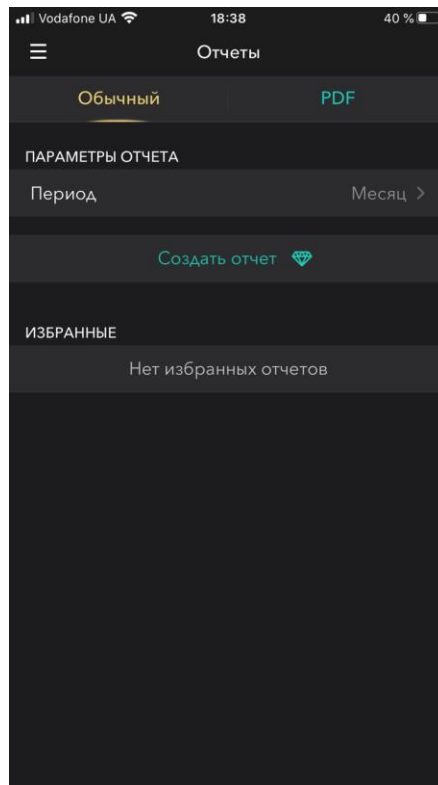


Рисунок 1.11 – Вікно оформлення звіту

Даний програмний продукт є найбільш відповідним до вимог предметної області, проте, одним із його недоліків, і найбільш значним є те, що більшість функцій цієї програми можна використовувати тільки у платній версії додатку, до того ж і з досить високою платою в місяць.

Досліджені програмні продукти можуть допомогти у розробці мобільного додатку як функціональними ідеями, так і у створенні інтуїтивно-зрозумілого та зручного інтерфейсу. Проаналізувавши програмний продукт «Монеон», було звернено увагу на структуру звіту та процес його формування та зручний інтерфейс додатку, що буде враховано при розробці програмного модуля.

2 СПЕЦИФІКАЦІЯ ВИМОГ ДО МОДУЛЯ

2.1 Глосарій

Глосарій термінів та понять предметної області наведений в таблиці 2.1.

Таблиця 2.1 – Глосарій до модуля

Термін	Опис терміну
1. Основні поняття і категорії предметної області та проєкту	
Дохід	Грошові надходження, що складаються із заробітної плати і додаткових надходжень.
Витрата	Зменшення обсягу коштів за рахунок покупки матеріальних цінностей, послуг, тощо.
Регулярні витрати	Витрати на найнеобхідніше або щоденні витрати (сплата комунальних послуг, аренди, виплати кредиту, харчування, сплата за проїзд, тощо).
Витрати на освіту	Витрати на саморозвиток (курси, книжки, абонемент у спортзалу, оплата за навчання).
Витрати на розваги	Витрати на кіно, кафе, ресторани, театри, масаж, похід у парк, тощо.
Великі витрати	Нерегулярні витрати (одежа, побутова техніка, медичні послуги).
Витрати на подарунки та благодійність	Нерегулярні витрати, які призначені для покупки подарунків, пожертвування в благодійні фонди.
Накопичення	Резерв, в який виділяються гроші на майбутні непередбачувані та великі витрати.
Транзакція	Різні операції з фінансами і рахунками.
Категорія витрат	Різновид витрат за їх характером.
2. Користувачі системи	
Користувач	Фізична особа, яка користується даним програмним продуктом.
3. Вхідні та вихідні документи	
Інформація про доходи	Інформація про прибутки користувача
Інформація про витрати	Інформація про витрати користувача
Список категорій	Таблиця бази даних з інформацією про категорії витрат
Звіт про стан фінансів за період	Текстовий файл, який містить інформацію про стан фінансів за обраний період часу.

2.2 Розроблення варіантів використання

2.2.1 Діаграма варіантів використання

Діаграма варіантів використання в UML - діаграма, на якій зображено відношення між акторами та прецедентами в системі, відображає функціональність, яка буде реалізована в програмному продукті. Варіант використання можна розглядати як функцію, що реалізується системою.

Для користувача у системі доступні наступні дії: управління витратами (видалення та внесення витрат), управління прибутком (видалення та внесення прибутку), перегляд історії(за категоріями, витрат, прибутку, звітів), та формування звіту.

Дії «Внесення даних про прибуток» та «Внесення даних про витрати» включають у себе введення суми та мають на вибір користувача вибір дати(за замовчуванням поточна дата). Дія «Внесення даних про витрати» включає вибір категорії витрат.

Дія «Перегляд історії» має чотири варіанта: перегляд історії витрат, прибутку, перегляд історії за категоріями та історії звітів.

Дія «Формування звіту» включає у себе вибір періоду, на який буде сформовано звіт та присвоєння назви.

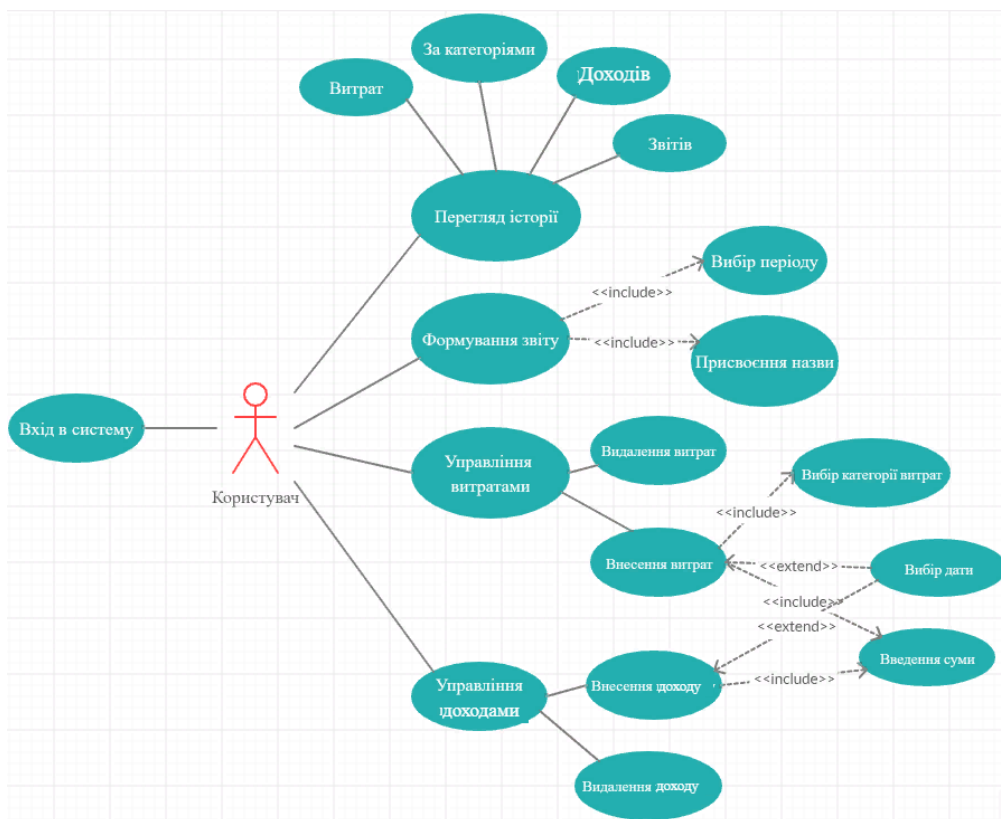


Рисунок 2.1 – Діаграма Use Case

2.2.2 Специфікація варіантів використання

Специфікацію варіантів використання наведено у таблицях 2.2. – 2.9.

Таблиця 2.2 – Варіант використання «Внесення доходу»

Контекст використання	UC-01 Внесення доходу
Діюча особа	Користувач
Передумова	Запущено додаток
Тригер	Отриманий новий дохід
Сценарій	<ol style="list-style-type: none"> 1. Обрати пункт меню «Доходи». 2. Натиснути кнопку «Новий дохід» 3. Ввести інформацію про суму доходу. 4. Обрати дату доходу (опціонально). 5. Натиснути кнопку «Додати».
Постумова	Успішне виконання: оновлюється список доходів з новим записом про дохід. При помилці відображається повідомлення про помилку.

Таблиця 2.3 – Варіант використання «Внесення витрати»

Контекст використання	UC-02 Внесення витрати
Діюча особа	Користувач
Передумова	Запущено додаток
Тригер	Виконана нова витрата
Сценарій	<ol style="list-style-type: none"> 1. Обрати пункт меню «Витрати». 2. Натиснути кнопку «Нова транзакція» 3. Обрати категорію витрати зі списку категорій. 4. Ввести суму витрати. 5. Обрати дату витрати (Опціонально). 6. Натиснути кнопку «Додати».
Постумова	Успішне виконання: оновлюється список витрат з новим записом про витрату. При помилці відображається повідомлення про помилку.

Таблиця 2.4 – Варіант використання «Перегляд витрат за категорією»

Контекст використання	UC-03 Перегляд витрат за категорією
Діюча особа	Користувач
Передумова	Запущено додаток
Тригер	Необхідність переглянути витрати за певною категорією
Сценарій	Натиснути на категорію, за якою необхідно переглянути витрати.
Постумова	Успішне виконання: відображається інформація про витрати за категорією. При помилці відображається повідомлення про помилку.

Таблиця 2.5 – Варіант використання «Видалення запису»

Контекст використання	UC-04 Видалення запису про дохід.
Діюча особа	Користувач
Передумова	Внесений хоча б один запис про дохід
Тригер	Необхідність видалити запис з інформацією про дохід
Сценарій	1. Обрати пункт меню «Доходи». 2. Натиснути кнопку «x» порід із записом, який необхідно видалити. 3. Натиснути кнопку «Видалити».
Постумова	Успішне виконання: список доходів оновлюється без щойно видаленого запису про дохід. При помилці відображається повідомлення про помилку.

Таблиця 2.6 – Варіант використання «Формування нового звіту про стан фінансів за період»

Контекст використання	UC-05 Формування нового звіту про стан фінансів за період
Діюча особа	Користувач
Передумова	Запущено додаток
Тригер	Необхідність переглянути стан фінансів за певний період
Сценарій	1. Обрати пункт меню «Звіти». 2. Натиснути кнопку «Сформувати звіт». 3. Ввести назву звіту. 4. Обрати початкову та кінцеву дату (опціонально). 5. Натиснути кнопку «Створити звіт».
Постумова	Успішне виконання: оновлюється список звітів з новим записом про звіт. При помилці відображається повідомлення про помилку.

2.3 Специфікація функціональних та нефункціональних вимог

Функціональні вимоги формулюють те, що саме у системі повинно бути реалізовано, а нефункціональні вимоги – з дотриманням яких умов це має виконуватися.

Специфікація функціональних вимог наведена в табл. 2.7.

Таблиця 2.7 – Специфікація функціональних вимог

Ідентифікатор вимоги	Назва вимоги (варіанту використання)	Атрибути вимог	
		Пріоритет	Трудність
FR-UC-01	Реалізувати можливість внесення даних про прибутки	1	Середня
FR-UC-02	Реалізувати можливість внесення даних про витрати	1	Середня
FR-UC-03	Реалізувати можливість видалення даних про прибутки	2	Середня
FR-UC-04	Реалізувати можливість видалення даних про витрати	2	Середня
FR-UC-05	Реалізувати можливість створення звіту	2	Важка
FR-UC-06	Реалізувати можливість видалення звіту	3	Середня
FR-UC-07	Реалізувати можливість збереження звіту на пристрій	5	Важка
FR-UC-08	Реалізувати можливість перегляду історії прибутків	2	Середня
FR-UC-09	Реалізувати можливість перегляду історії витрат	2	Середня
FR-UC-10	Реалізувати можливість перегляду історії звітів	3	Середня
FR-UC-11	Реалізувати можливість перегляду історії витрат за категорією	4	Важка

Контактом та виконавцем у специфікації функціональних вимог (таблиця 2.7) є головний директор ФОП «Шнуренко Дмитро Геннадійович».

Специфікація нефункціональних вимог наведена в табл. 2.8.

Таблиця 2.8 – Специфікація нефункціональних вимог

Ідентифікатор Вимоги	Назва вимоги	Атрибути вимог	
		Пріоритет	Трудність
1	2	3	4
1. Застосовність			
SUPP-01	Час завантаження форми – не більше 5 сек.	Обов'язкова	Низька
SUPP-02	Час відгуку програми – не більше 1 сек.	Обов'язкова	Середня
2. Надійність			
SUPP-03	Стабільність у роботі	Обов'язкова	Важка
SUPP-04	Час безвідмовної роботи не менше 60 годин	Обов'язкова	Середня
SUPP-05	Стійкість до збоїв	Обов'язкова	Важка
SUPP-06	Запобігання втраті даних	Обов'язкова	Важка
3. Робочі характеристики			
SUPP-07	Швидкість обробки нових даних 2 – 5 сек.	Обов'язкова	Середня
SUPP-08	Швидкість обробки запиту на формування звіту – 0 - 3 сек.	Обов'язкова	Середня
4. Проектні обмеження			
SUPP-09	Мова програмування Java	Обов'язкова	Низька
5. Інтерфейси			
5.1. Інтерфейси користувача			
SUPP-10	Мінімалістичний дизайн віконних форм	Обов'язкова	Середня
SUPP-11	Приємні, неяскаві кольори	Обов'язкова	Середня
SUPP-12	Повідомлення про помилку виводиться у вікні, що впливає	Обов'язкова	Низька
5.2. Апаратні інтерфейси			
SUPP-13	Наявність мобільного пристрою із ОС Android	Обов'язкова	Середня
SUPP-14	Наявність мобільного пристрою з об'ємом оперативної пам'яті від 2 Гб	Обов'язкова	Середня

Закінчення таблиці 2.8

1	2	3	4
6. Вживані стандарти			
SUPP-15	Стандарт якості програмного продукту ISO/IEC 9126	Обов'язкова	Середня
SUPP-16	Стандарт якості програмного продукту ISO/IEC 25010	Обов'язкова	Середня

Контактом у специфікації нефункціональних вимог (таблиця 2.8) є керівник відділу розробки програмного забезпечення.

2.4 Проектування інтерфейсу користувача

При розробці дизайну інтерфейсу додатку було дотримано наступних критеріїв:

1. Інтуїтивність
2. Логічність
3. Відповідність ТЗ
4. Адаптивність

Для створення мокапу інтерфейсу користувача було застосовано додаток Figma. Це веб сервіс, який не потребує встановлення на ПК, має широкий спектр інструментів, також має готові шаблони вікон під основні пристрої, такі як iPhone, Google Pixel, Android, iPad, MacBook, iMac, AppleWatch. Також має шаблони аркушів паперу та публікацій у соціальних мережах.

На Рис.2.2 відображене головне меню додатку.

У головному меню відображено всі категорії витрат та процес виконання плану. Коли витрати перевищують ліміт, шкала заповнення стає червоною, а у звіті ця категорія буде відмічена, як найбільш витратна.

У головному меню користувач може перейти до історії прибутків (рис. 2.3), витрат (рис. 2.5) та переглянути звіти (рис. 2.8). Крім того, має змогу переглянути історію витрат кожної категорії (рис. 2.9), натиснувши на назву потрібної категорії. Також відображена інформація про загальний прибуток та загальні витрати за місяць.

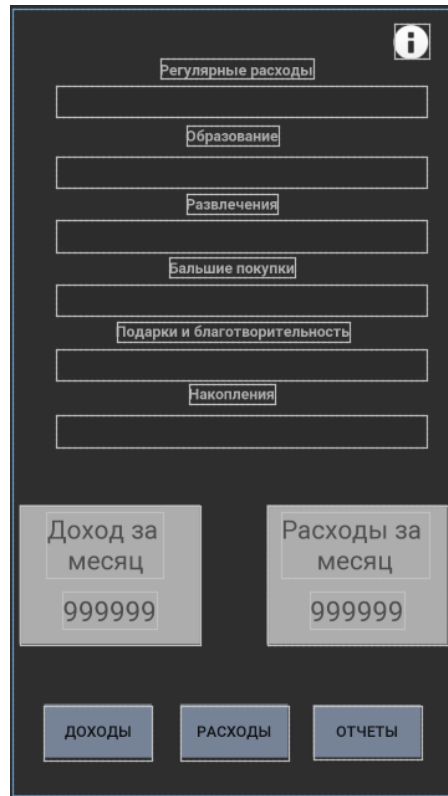


Рисунок 2.2 – Головне меню



Рисунок 2.3 – Історія прибутків

У вікні перегляду історії прибутків користувач може ввести новий прибуток (рис. 2.4), переглянути інформацію про дату транзакції та суму прибутку, видалити будь-який запис.

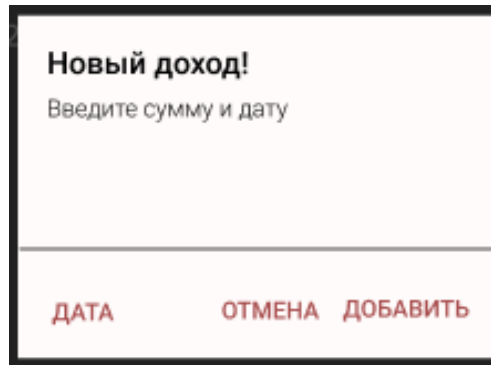


Рисунок 2.4 – Діалог внесення нового прибутку

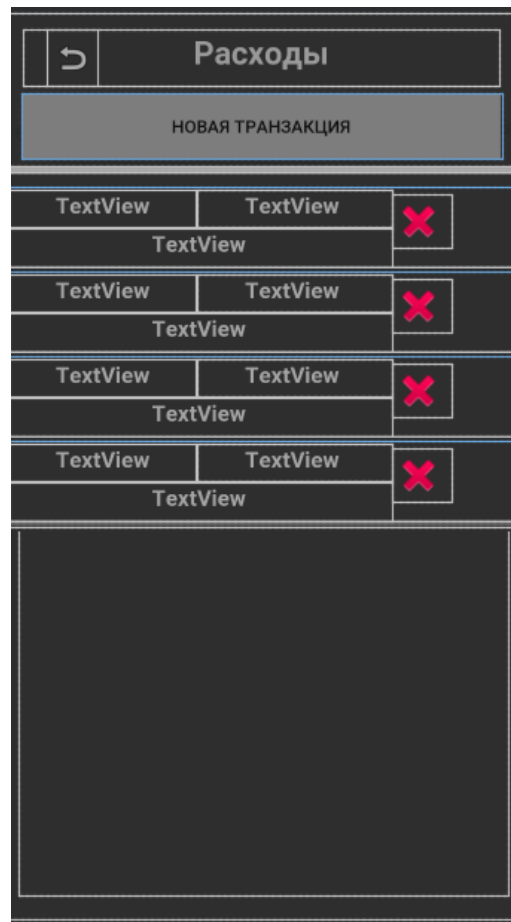


Рисунок 2.5 – Вікно історії витрат

У цьому вікні відображено інформацію про дату транзакції, підкатегорію витрати та суму. Також будь-який запис можна видалити або

додати новій. Якщо користувач натисне кнопку «Новая транзакция», то відкриється вікно вибору категорії витрат (рис.2.6).

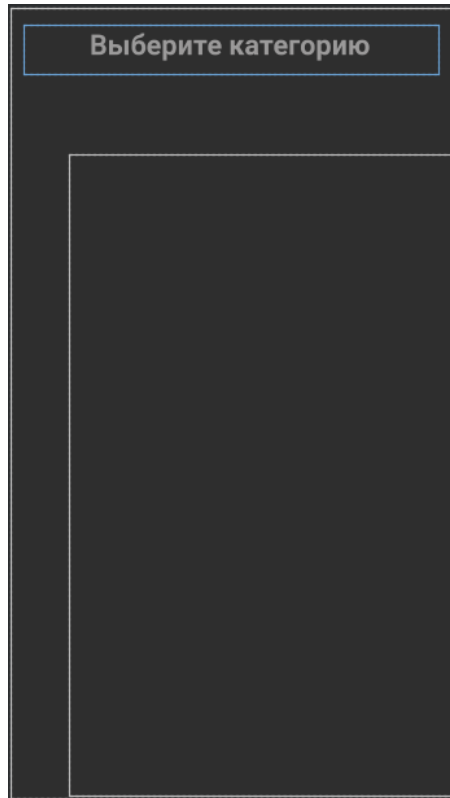


Рисунок 2.6 – Вікно вибору категорії

Після вибору категорії відкриється діалогове вікно, в якому користувач має ввести суму витрати, дату та зберегти інформацію (рис.2.7).

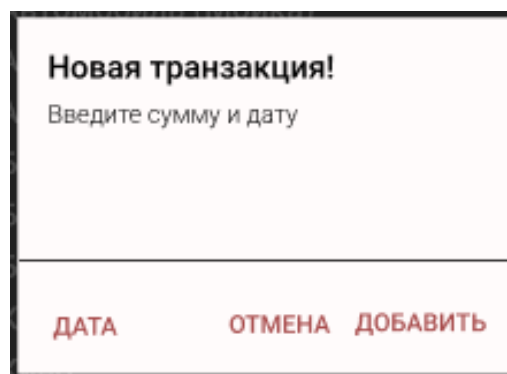


Рисунок 2.7 – Діалогове вікно внесення витрати

Як вже було зазначено, користувач може переглядати список звітів (рис.2.8) та, за необхідності, формувати нові звіти (рис. 2.9).

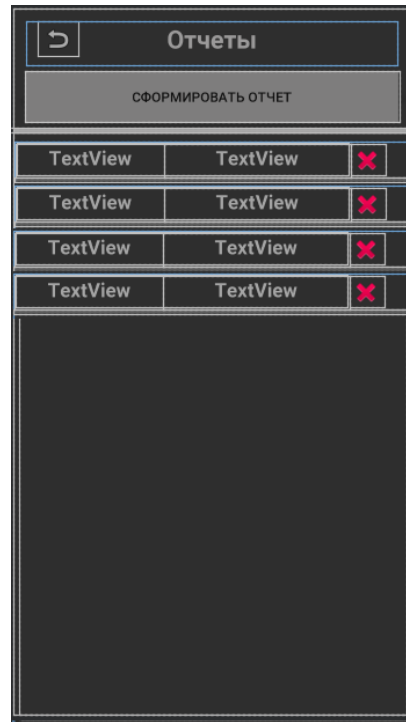


Рисунок 2.8 – Вікно перегляду звітів



Рисунок 2.9 – Вікно формування звіту

3 ПРОЄКТНІ ТА ТЕХНІЧНІ РІШЕННЯ

3.1 Математична та логічна постановка

Розв'язання задачі починається, коли система отримує на вхід інформацію про доходи. Користувач вносить новий дохід, заповнюючи поля суми та дати доходу. Після цього система формує план розподілу фінансів за категоріями, застосовуючи математичні формули, які описано у таблиці 3.1.

Таблиця 3.1 – Реквізити відомостей обліку бюджету.

Найменування реквізиту	Вид реквізиту	Примітки
Дохід (Д)	Початковий	Вводиться користувачем
Регулярні витрати (РВ)	Розрахунковий	$РВ=Д*0,6$
Розваги (Р)	Розрахунковий	$Р=Д*0,1$
Освіта (О)	Розрахунковий	$О=Д*0,1$
Великі витрати (ВВ)	Розрахунковий	$ВВ=Д*0,05$
Благодійність та подарунки (БП)	Розрахунковий	$БП=Д*0,05$
Накопичення (Н)	Розрахунковий	$Н=Д*0,1$

Наступним кроком користувач повинен ввести інформацію про витрати, обираючи категорію витрат зі списку категорій, які містяться в системі. Після вибору категорії витрат, користувач має ввести суму витрати та обрати дату, якщо вона не є поточною.

Під час перегляду історії витрат та доходів користувач може видалити обраний запис, натиснувши хрестик навпроти.

Під час формування звіту користувач має ввести назву звіту, обрати початкову та кінцеву дати, кінцева дата не може бути раніше початкової. Система сортує дані про витрати та прибуток за обраним періодом, відображає прогрес виконання плану витрат за категоріями, порівнює заплановані витрати з фактичними та відображає рекомендації щодо оптимізації витрат.

Під час перегляду історії звітів користувач може переглянути будь-який раніше сформований звіт та, за необхідності, скачати його на пристрій у форматі .txt . Також користувач може видалити будь-який звіт, натиснувши на хрестик навпроти потрібного запису.

3.2 Проєктування структури бази даних

3.2.1 Концептуальне інфологічне проєктування

На основі аналізу вхідних та вихідних документів побудовано модель відображення множини реквізитів вихідних і вхідних документів на множину елементів даних, що підлягають збереженню у базі даних, виконано приведення зібраної інформації до вигляду, зручного для проєктування. Для спрощення цієї роботи було складено словник даних (табл. 3.2).

Таблиця 3.2 – Словник даних

Найменування елемента	Ідентифікатор	Тип і довжина	Призначення елемента
Назва категорії	categoryName	Текст, 50 знаків	Фактичний
Назва підкатегорії	subcategoryName	Текст, 50 знаків	Фактичний
Дата витрати	expenseDate	Текст, до 10 знаків	Фактичний
Сума витрати	expenseAmount	Число, дробове	Фактичний
Дата прибутку	incomeDate	Текст, до 10 знаків	Фактичний
Сума прибутку	incomeAmount	Число, дробове	Фактичний
Початкова дата	firstDate	Текст, до 10 знаків	Фактичний
Кінцева дата	secondDate	Текст, до 10 знаків	Фактичний
Дата звіту	reportDate	Текст, до 10 знаків	Фактичний
Назва звіту	reportName	Текст, до 50 знаків	Фактичний
Текст звіту	reportText	Текст, до 1000 знаків	Фактичний
Початкова дата звіту	firstReportDate	Текст, до 10 знаків	Фактичний
Кінцева дата звіту	secondReportDate	Текст, до 10 знаків	Фактичний

Обмеження унікальності наведено у таблиці 3.3.

Таблиця 3.3 – Обмеження атрибутів сутностей

Ім'я атрибута	Межі \ допустимі значення	Структура (формат)	Умова	Значення за замовчуванням
1	2	3	4	5
Category_id	Ціле, більше нуля, NOT NULL, унікальний	9(0)	-	1 або на 1 більше від попереднього
Expense_id	Ціле, більше нуля, NOT NULL, унікальний	9(0)	-	1 або на 1 більше від попереднього

Закінчення таблиці 3.3

1	2	3	4	5
Income_id	Ціле, більше нуля, NOT NULL, унікальний	9(0)	-	1 або на 1 більше від попереднього
Report_id	Ціле, більше нуля, NOT NULL, унікальний	9(0)	-	1 або на 1 більше від попереднього
categoryName	NOT NULL	X(50)	-	-
subcategoryName	NOT NULL	X(50)	-	-
expenseDate	Може містити точку, NOT NULL	X(10)	-	Сьогоднішня дата
expenseAmount	NOT NULL	9(0)	-	-
incomeDate	Може містити точку, NOT NULL	X(10)	-	Сьогоднішня дата
firstDate	Може містити точку, NOT NULL	X(10)	-	Місяць до сьогоднішньої дати
secondDate	Може містити точку, NOT NULL	X(10)	-	Сьогоднішня дата
reportDate	Може містити точку, NOT NULL	X(10)	-	Сьогоднішня дата
reportName	NOT NULL	X(50)	-	-
reportText	NOT NULL	X(1000)	-	-
firstReportDate	Може містити точку, NOT NULL	X(10)	-	Місяць до сьогоднішньої дати
secondReportDate	Може містити точку, NOT NULL	X(10)	-	Сьогоднішня дата

Обмеження кортежів потребують таблиці «expense», «report» та «income».

Expense (expense_id, expenseDate, expenseAmount, subcategory_id) - expenseAmount > 0.

Income (income_id, incomeDate, incomeAmount) - incomeAmount > 0.

dates (dates_id, firstDate, secondDate) - firstDate < secondDate.

report (report_id, date, firstReportDate, secondReportDate, reportName, reportText, expense_id, income_id) - firstReportDate < secondReportDate.

Правила посилальної цілісності наведено у таблиці 3.4.

Таблиця 3.4 – Правила посилальної цілісності

Батьківське відношення	Дочірнє відношення	Ключ, який є у дочірньому відношенні	Обмеження кортежу
Subcategories	Expense	subcategory_id	Expense не може містити значень атрибуту subcategory_id, яке не використовується у Subcategories
Categories	Subcategories	category_id	Subcategories не може містити значень атрибуту category_id, яке не використовується у Categories
Expense	Report	Expense_id	Report не може містити значень атрибуту Expense_id, яке не використовується у Expense
Income	Report	Income_id	Report не може містити значень атрибуту Income_id, яке не використовується у Income

3.2.2 Проектування логічної моделі даних

Для створення бази даних була обрана бібліотека SQLite, оскільки це вбудована бібліотека, яка реалізує автономний, безсерверний, нульової конфігурації, транзакційний механізм СУБД SQL. Це база даних, яка налаштована на нуль, що означає, як і інші бази даних, які вам не потрібно налаштовувати в вашій системі.

SQLite не є автономним процесом, як інші бази даних, ви можете пов'язати його статично або динамічно відповідно до вашим вимогою з вашим додатком. SQLite безпосередньо звертається до своїх файлів зберігання.

Ця технологія має наступні особливості у порівнянні із іншими СУБД:

- SQLite не вимагає окремого процесу сервера або системи для роботи (без сервера).
- SQLite поставляється з нульовою конфігурацією, що означає відсутність необхідності в налаштуванні або адмініструванні.
- Повна база даних SQLite зберігається в одному крос-платформенному

- диску.
- SQLite дуже маленький і легкий, менш 400KiB повністю сконфігурованих-менш 250KiB з додатковими функціями, опущеними.
 - SQLite є автономним, що означає відсутність зовнішніх залежностей.
 - SQLite-транзакції повністю сумісні з ACID, забезпечуючи безпечний доступ до декільком процесам або потокам.
 - SQLite підтримує більшість функцій мови запитів, знайдених в стандарті SQL92 (SQL2).
 - SQLite написаний на ANSI-C і забезпечує легкий і простий у використанні API.
 - SQLite доступний в UNIX (Linux, Mac OS-X, Android, iOS) і Windows (Win32, WinCE, WinRT).

Логічну модель даних в нотатції IDEF1X відображено на рис. 3.1.

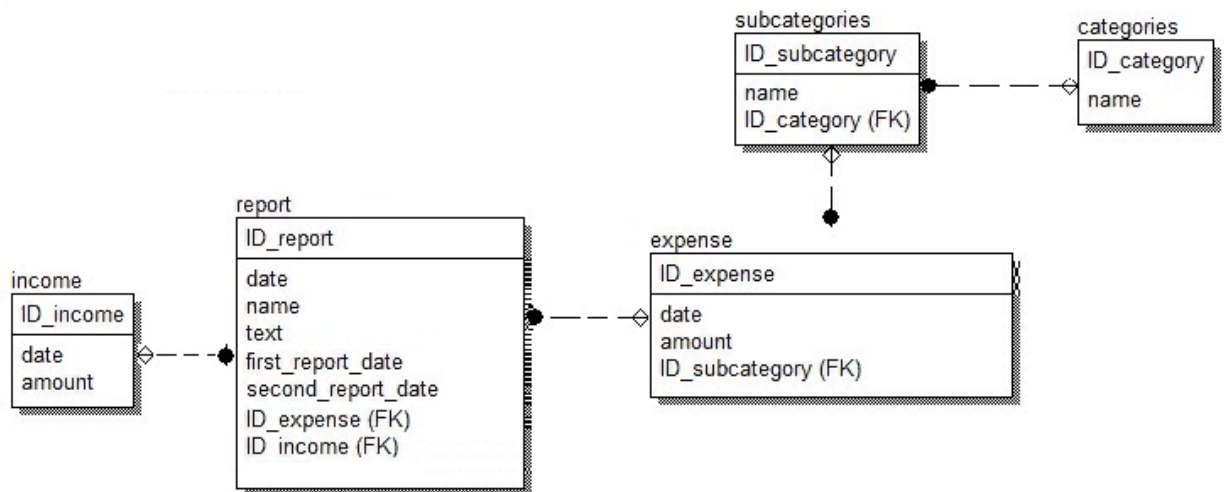


Рисунок 3.1 – Логічна модель даних

SQLite відповідає вимогам ACID (атомарність, узгодженість, ізольованість і надійність) і відомий своєю посиленою і транзакційною цілісністю. Первинні ключі, що обмежують і каскадні зовнішні ключі, унікальні обмеження, обмеження NOT NULL, перевірочні обмеження та інші функції забезпечення цілісності даних дають впевненість, що тільки коректні дані будуть збережені.

Первинна захищеність та безпека бази даних забезпечена за допомогою валідації введених даних. Додаткова система безпеки, по типу авторизації, не є необхідною для даного модуля.

3.2.3 Проектування фізичної моделі даних

На рис. 3.2 відображено фізичну модель даних. Код, який було застосовано для створення таблиць, наведено у додатку А.

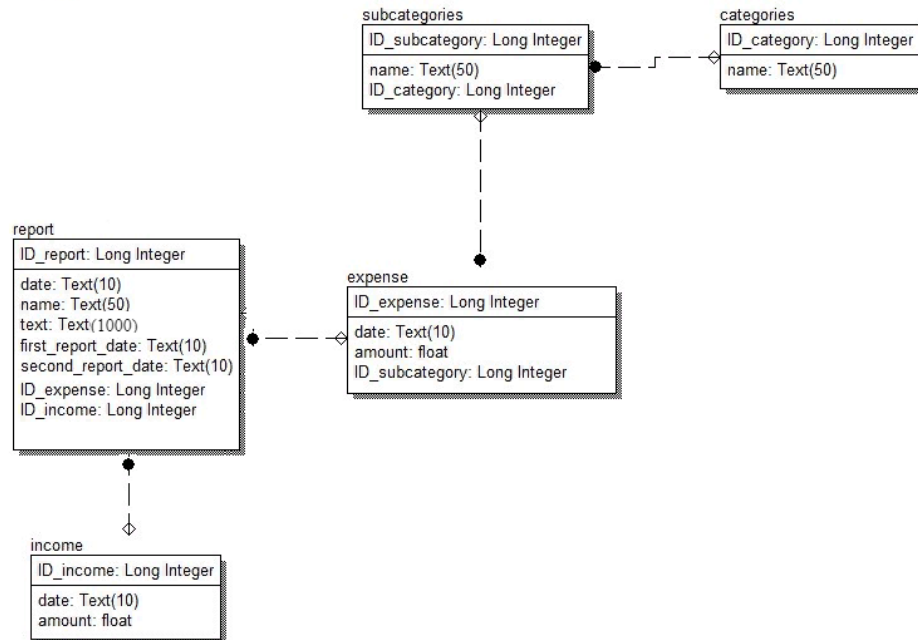


Рисунок 3.2 – Фізична модель даних

3.3 Проектування програмного забезпечення

Для розробки програмного забезпечення було обрано мову програмування Java, оскільки вона є об'єктно-орієнтованою, незалежною від платформи, має багато безкоштовних інструментів для програмування.

В якості середовища розробки коду було застосовано Android Studio. Оскільки це найпопулярніша середовище розробки для системи Android та має вбудований інструмент для роботи з XML файлами розмітки. Також вона має безкоштовний варіант з підтримкою мови Java та працює з системами контролю версій, зокрема з Git. Також Android Studio може вводити код SQL, HTML, CSS у строкові літерали Java. Android Studio вміє аналізувати код під час завантаження та введення, вказувати на можливі проблеми. Також одною із особливостей цієї середовища розробки є те, що вона містить вбудований інструмент авто-зборки проєктів Gradle, що дозволяє підключати сторонні бібліотеки напряму із серверів та аналізувати проєктні пакети на етапі проєктування.

У якості системи контролю версій було застосовано GitHub – веб-додаток та систему для управління репозиторіями програмного коду для Git. GitHub пропонує вирішення для зберігання коду масштабних проєктів. Репозиторій містить в собі систему контролю версій для розміщення

різноманітних ланцюгів та гілок, надаючи розробникам перевіряти код та відкатувати версії програм у разі непередбачуваних проблем.

Також для створення додатку з насиченим графічним інтерфейсом було застосовано вбудований у середу розробки архітектурний компонент Navigation. Він дозволяє спростити реалізацію навігації між екранами призначення (destinations) в вашому додатку. За замовчуванням, Navigation підтримує фрагменти (Fragments) і активності (Activities) в якості екранів призначення.

Для відображення статичної структури моделі програмної системи було розроблено UML-діаграму класів. На діаграмі відображено інформацію про класи, інтерфейси, поля та методи класів, спадкування класів. Діаграма міститься у додатку Б.

Коректна робота додатку забезпечена взаємодією груп класів із відповідними layout-файлами інтерфейсу користувача. Так безпосередньо із xml файлами взаємодіють класи activity (Додаток В), котрі обробляють івенти дій користувача; класи adapters (Додаток Г) виконують роль генерації та обробки даних із бази даних для відображення; класи entities (Додаток Д) описують структури даних, котрі використовуються на етапах проектування та розробки додатку.

Для детального опису станів системи під час взаємодії з користувачем було розроблено UML-діаграми станів елементів графічного інтерфейсу (рис 3.3).

На діаграмі детально та послідовно описані дії користувача під час усієї роботи з програмою та відгук системи на ці дії.

Зображення працюючих вікон програми наведено у додатку Е.

3.4 Тестування програмної системи

Для перевірки системи на відповідність вимогам, виявлення дефектів системи та умов, за яких ці дефекти проявляються, а також для можливості усунення виявлених дефектів було проведено системне тестування системи, містить в собі функціональне тестування та тестування безпеки.

Було розроблено тест-вимоги, які наведено у таблиці 3.4.

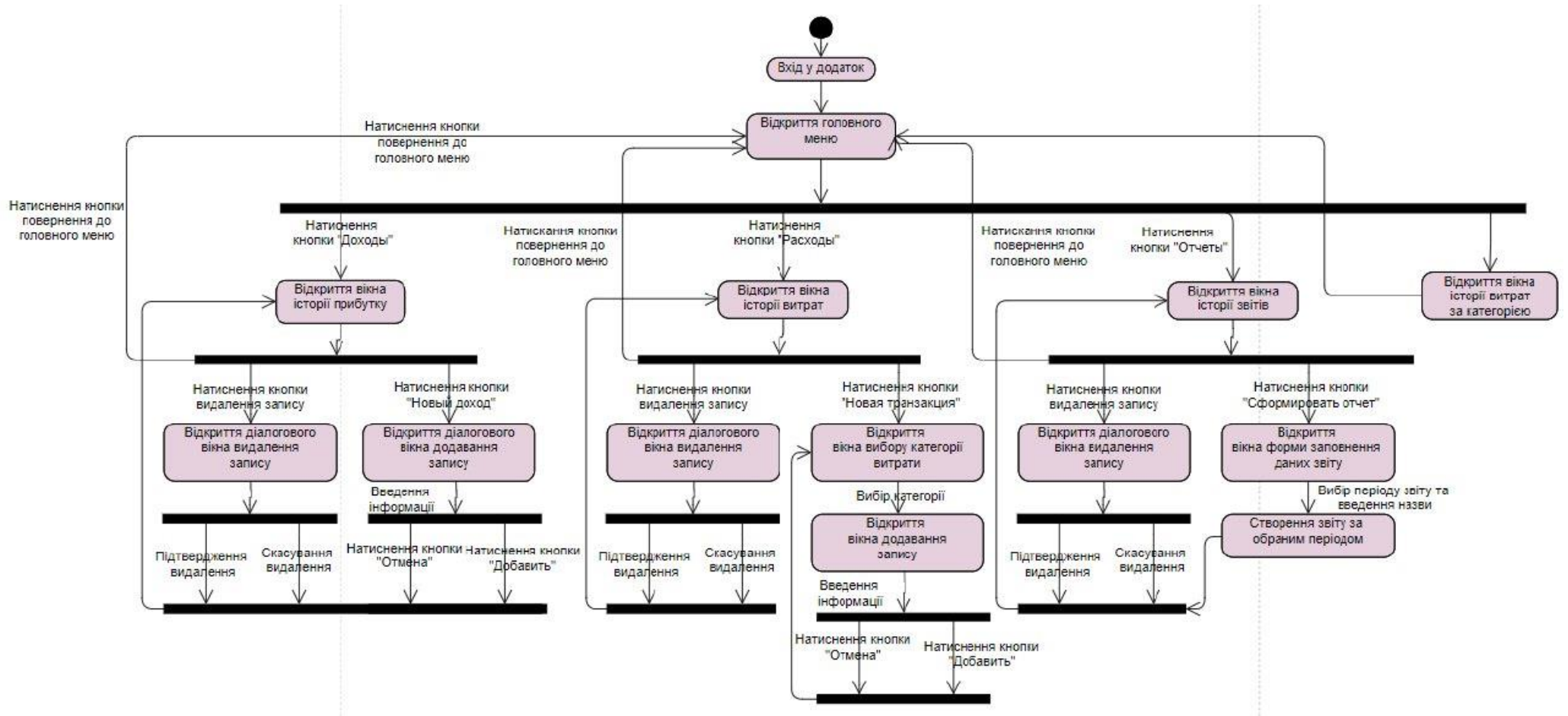


Рисунок 3.3 – Діаграма станів інтерфейсу користувач

Таблиця 3.4 – Тест-вимоги

Ідентифікація вимоги	Назва вимоги	Опис вимоги	Очікуваний результат	Фактичний результат
TR-01	Внесення нового запису доходу до бази даних	Перевірка текстового поля суми на валідність.	Система виводить попередження про помилку в даних.	Система виводить попередження про помилку в даних.
TR-02	Внесення нового запису витрати до бази даних	Перевірка текстового поля суми на валідність.	Система виводить попередження про помилку в даних.	Система виводить попередження про помилку в даних.
TR-03	Внесення нового запису звіту до бази даних	Перевірка текстових полів на валідність та коректність вибору дати.	Система виводить попередження про помилку в даних.	Система виводить попередження про помилку в даних.
TR-04	Збереження текстового файлу звіту на пристрій	Коректне збереження файлу в каталог із програмою	Система виводить попередження про помилку збереження.	Система виводить попередження про помилку збереження.
TR-05	Видалення доходу з бази даних	Коректне видалення запису з БД.	Дохід не відображається у списках.	Дохід не відображається у списках.
TR-06	Видалення витрати з бази даних	Коректне видалення запису з БД.	Витрата не відображається у списках.	Витрата не відображається у списках.
TR-07	Видалення звіту з бази даних	Коректне видалення запису з БД.	Звіт не відображається у списках.	Звіт не відображається у списках.

Для більш детального опису тестування функціоналу системи було розроблено тест-плани до кожної тест-вимоги. Перелік тест-планів наведено нижче.

Тест-план №1

Призначення: перевірка того, що при додаванні нового запису доходу додаток виконує перевірку на валідність текстового поля суми.

Тест-вимога, що перевіряється: TR-1

Передумови для тесту: система має бути переведена в стан роботи, відкрите вікно доходів, натиснута кнопка «Додати дохід».

Критерій проходження тесту: всі реальні значення збігаються з очікуваними.

Тест-план №1 наведений у таблиці 3.5.

Таблиця 3.5 – Тест-план №1

Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
Ввести нечислове значення у текстове поле суми	Має з'явитися впливаюче повідомлення про некоректність введених даних	З'явилося впливаюче повідомлення про некоректність введених даних	Так
Ввести значення «1000» у текстове поле суми	-	-	-
Натиснути кнопку «Дата»	Має з'явитися діалогове вікно вибору дати	З'явилося діалогове вікно вибору дати	Так
Натиснути кнопку «Додати»	Має закритися діалогове вікно додавання запису, запис відображений у списку	Діалогове вікно закрилося, запис коректно відображений у списку доходів	Так

Відмітка про проходження тесту (пройдений/не пройдений): пройдений.

Тест-план №2

Призначення: перевірка того, під час додавання витрати система перевіряє дані, які вводить користувач на валідність та зберігає запис у базу даних.

Тест-вимога, що перевіряється: TR-02

Передумови для тесту: система має бути приведена в стан роботи, відкрите вікно витрат, натиснута кнопка «Нова транзакція».

Критерій проходження тесту: всі реальні значення збігаються з очікуваними.

Тест-план №2 наведений у таблиці 3.6.

Таблиця 3.6 – Тест-план №2

Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
У новому вікні обрати категорію витрати зі списку	-	-	-
Ввести від'ємне значення суми у текстовому полі суми	Має з'явитися впливаюче повідомлення про некоректні введені дані	З'явилося впливаюче повідомлення про некоректні введені дані	Так
Ввести значення «550» у полі суми	-	-	-
Натиснути кнопку «Додати»	Має відкритися вікно витрат, доданий запис повинен відобразитися у списку витрат	Вікно витрат відкрилося, з'явився новий запис у списку	Так

Відмітка про проходження тесту (пройдений/не пройдений): пройдений.

Тест-план №3

Призначення: перевірка того, що під час додавання нового звіту система перевіряє обрані дати та зберігає запис у базу даних.

Тест-вимога, що перевіряється: TR-03

Передумови для тесту: система має бути приведена в стан роботи, відкрите вікно звітів, натиснута кнопка «Сформувати звіт».

Критерій проходження тесту: всі реальні значення збігаються з очікуваними.

Тест-план №3 наведений у таблиці 3.7.

Таблиця 3.7 – Тест-план №3

Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
Ввести «Новий звіт» у полі назви звіту	-	-	-
Натиснути кнопку вибору початкової дати	-	-	-
Обрати дату «20.01.2020» у діалоговому вікні	-	-	-
Натиснути кнопку вибору кінцевої дати			
Обрати дату «01.01.2020» у діалоговому вікні	Дати повинні замінитися одна на одну	Початкова та кінцева дати замінилися одна на одну	Так
Натиснути кнопку «Додати звіт»	Має завантажитися вікно звітів, новий запис повинен відображатися у списку звітів	Відповідне вікно завантажилось, запис коректно відобразився у списку звітів	Так

Відмітка про проходження тесту (пройдений/не пройдений): пройдений.

Тест-план №4

Призначення: перевірка того, текстовий файл звіту зберігається на пристрої.

Тест-вимога, що перевіряється: TR-04

Передумови для тесту: система має бути приведена в стан роботи, відкрите вікно звітів.

Критерій проходження тесту: всі реальні значення збігаються з очікуваними.

Тест-план №4 наведений у таблиці 3.8.

Таблиця 3.8 – Тест-план №4

Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
Натиснути на запис «Новий звіт»	Має відкритися діалогове вікно із текстом звіту	Діалогове вікно відкрилося	Так
Натиснути кнопку «зберегти»	Має з'явитися вспливаюче повідомлення про успішне збереження, текстовий файл з'явився у каталозі додатку	З'явилося вспливаюче вікно із повідомленням «Saved». На пристрої зберігся текстовий файл звіту	Так

Відмітка про проходження тесту (пройдений/не пройдений): пройдений.

Тест-план №5

Призначення: перевірка того, що обраний запис доходу видалється із бази даних.

Тест-вимога, що перевіряється: TR-5

Передумови для тесту: система має бути приведена в стан роботи, відкрите вікно доходів.

Критерій проходження тесту: всі реальні значення збігаються з очікуваними.

Тест-план №5 наведений у таблиці 3.9.

Таблиця 3.9 – Тест-план №5

Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
Натиснути на запис доходу зі списку	Має з'явитися діалогове вікно підтвердження видалення запису	з'явилося діалогове вікно підтвердження видалення запису	Так
Натиснути кнопку «Видалити»	Має закритися діалогове вікно підтвердження, запис видалений зі списку доходів	Діалогове вікно закрилося, запис зник зі списку доходів	Так

Відмітка про проходження тесту (пройдений/не пройдений): пройдений.

Тест-план №6

Призначення: перевірка того, що обраний запис витрати видалається із бази даних.

Тест-вимога, що перевіряється: TR-6

Передумови для тесту: система має бути приведена в стан роботи, відкрите вікно витрат.

Критерій проходження тесту: всі реальні значення збігаються з очікуваними.

Тест-план №6 наведений у таблиці 3.10.

Таблиця 3.10 – Тест-план №6

Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
Натиснути на запис витрати зі списку	Має з'явитися діалогове вікно підтвердження видалення запису	З'явилося діалогове вікно підтвердження видалення запису	Так
Натиснути кнопку «Видалити»	Має закритися діалогове вікно підтвердження, запис видалений зі списку витрат	Діалогове вікно закрилося, запис зник зі списку витрат	Так

Відмітка про проходження тесту (пройдений/не пройдений): пройдений.

Тест-план №7

Призначення: перевірка того, що обраний запис звіту видалається із бази даних.

Тест-вимога, що перевіряється: TR-7

Передумови для тесту: система має бути приведена в стан роботи, відкрите вікно звітів.

Критерій проходження тесту: всі реальні значення збігаються з очікуваними.

Тест-план №7 наведений у таблиці 3.11.

Таблиця 3.11 – Тест-план №7

Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
Натиснути на запис звіту зі списку	Має з'явитися діалогове вікно підтвердження видалення запису	З'явилося діалогове вікно підтвердження видалення запису	Так
Натиснути кнопку «Видалити»	Має закритися діалогове вікно підтвердження, запис видалений зі списку звітів	Діалогове вікно закрилося, запис не зник зі списку звітів, з'явилося повідомлення про помилку видалення	Ні

Відмітка про проходження тесту (пройдений/не пройдений): не пройдений.
Тестових прикладів пройдено 6/ усього 7.

Помилка при видаленні запису звіту була виправлена, повторне проходження тест-плану №7 дало позитивний результат.

3.5 Розгортання програмного продукту

Файл додатку має розширення APK, тому для коректного функціонування додатку необхідно виконати наступні вимоги до апаратних та програмних засобів:

- Операційна система Android версії sdk 15 та вище;
- Об'єм оперативної пам'яті не менше 2G;
- Об'єм вільного дискового простору не менше 1G;

Користувач має натиснути на APK-файл та встановиться додаток. Можливо, відкриється діалогове вікно вибору дозволу доступу до файлової системи пристрою. У цьому разі необхідно увімкнути перемикач навпроти запиту. У разі помилки при встановленні необхідно перевірити дозвіл на встановлення сторонніх додатків у налаштуваннях смартфона.

Інший спосіб встановити додаток – завантажити з Google Play Market. У даному виді установки усі налаштування відбуваються автоматично.

При успішному встановленні додатку, у списку додатків на смартфоні повинна з'явитися іконка додатку з ім'ям «MyBudget».

Для запуску додатку необхідно натиснути на його іконку у списку програм смартфона.

ВИСНОВКИ

У результаті написання дипломного проекту було розроблено програмний модуль обліку особистих фінансів у вигляді мобільного додатку для смартфонів операційної системи Android. Даний програмний продукт дозволяє стежити за станом фінансів, розподіляючи доходи та витрати на 6 різних категорій за системою контролю фінансів «Метод глечиків». За допомогою цієї програми можна відстежувати поточний та минулий стан доходів та витрат та аналізувати процес ведення особистого бюджету за допомогою системи формування звітів про стан фінансів за певний період.

Було реалізовано наступні вимоги: наведено коротку характеристику об'єкту управління «ФОП Шнуренко Дмитро Геннадійович» за допомогою програмного засобу проектування Aris Express; наведено опис предметної області «Облік особистих фінансів» у вигляді діаграм *idef0* у Ramus Educational та Aris Express; проведено огляд і аналіз існуючих аналогів, що реалізують функції предметної області, таких як Moneop та Monefy; розроблено глосарій термінів; розроблено діаграму варіантів використання за допомогою Use-Case засобів; наведено специфікацію варіантів використання та специфікацію функціональних та нефункціональних вимог; спроектовано інтерфейс користувача (мокап) у середовищі розробки дизайну Figma; наведено математичну та логічну постанову завдання; спроектовано базу даних (інфологічне проектування, фізична на логічна модель) за допомогою програмного засобу проектування бази даних ErWin; проведено тестування програмного продукту та наведено розгортання програми з вимогами до системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Методичні рекомендації до виконання курсового проекту з навчальної дисципліни "Організація баз даних та знань" / уклад. Л. А. Павленко, О. В. Тарасов, М. Ю. Лосєв та ін. Харків. : Вид. ХНЕУ, 2013. 84 с.
- 2 Ушакова І. О. Проектування інформаційних систем : Практикум / І. О. Ушакова. Харків : Вид. ХНЕУ ім. С. Кузнеця, 2015. 250 с.
- 3 Методичні рекомендації до виконання кваліфікаційних робіт для студентів спеціальностей 121 "Інженерія програмного забезпечення", 122 "Комп'ютерні науки", 126 "Інформаційні системи та технології" першого (бакалаврського) рівня / укл. Ю. Е. Парфьонов, І. О. Ушакова; ХНЕУ ім. С. Кузнеця. Харків : ХНЕУ ім. С. Кузнеця, 2020. 61 с.
- 4 Сервіс для створення Use-case діаграм : Веб-сайт. URL: <https://app.creately.com/diagram/h6x1W4Qrm3r/edit> (дата звернення: 01.05.2020).
- 5 IDEF0 как инструмент моделирования процессов – довідник з роботи з діаграмами IDEF0 : Веб-сайт. URL: <http://www.logists.by/library/view/instrymenty-biznes-modelirovaniya> (дата звернення: 01.05.2020).
- 6 Навчально–методичний посібник Моделювання бізнес-процесів з використанням методології ARIS / укл. В.И. Морозова, К.Э. Врублевський; Російський університет транспорту (МІІТ) Москва: 2017. 49 с.
- 7 Метод кувшинов. Эффективная система ведения вашего бюджета : Веб-сайт. URL: <https://lifehacker.ru/byudzhet-metod-kuvshinov/> (дата звернення: 05.04.2020).
- 8 Android Studio та його компоненти : Веб-сайт. URL: <http://developer.alexanderklimov.ru/android/> (дата звернення: 05.04.2020).
- 9 ListView – використання та елементи управління списками : Веб-сайт. URL: <http://developer.alexanderklimov.ru/android/views/listview.php> (дата звернення: 15.05.2020).
- 10 Методичні рекомендації до виконання курсового проекту з навчальної дисципліни "Організація баз даних та знань" / уклад. Л. А. Павленко, О. В. Тарасов, М. Ю. Лосєв та ін. Харків. : Вид. ХНЕУ, 2013. 84 с. 2. Ушакова І. О.
- 11 Проектування інформаційних систем : Практикум / І. О. Ушакова. – Харків : Вид. ХНЕУ ім. С. Кузнеця, 2015. 250 с.
- 12 Глушаков С. В. Базы даних / С. В. Глушаков, Д. В. Ломотько. Харків : Фоліо, 2017. 504 с.
- 13 Дейт К. Дж. Введение у системи баз даних / К. Дж. Дейт. 6-е вид.

Київ : Діалектика, 2015. 784 с.

14 ДСТУ 8302:2015 "Бібліографічне посилання. Загальні положення та правила складання" Київ : Держстандарт України, 2007. 52 с.

15 Котляров В. П. Основи тестування програмного забезпечення : навчальний посібник / В. П. Котляров, Т. В. Коликова. Москва : Інтернет-Університет Інформаційних Технологій ; БІНОМ. Лабораторія знань, 2009. 285 с.

16 Проектування баз даних. Інформаційний надлишок. Надмірність у базах даних. – IT-блог : Веб-сайт. URL: <https://zametkinapolyah.ru/zametki-o-mysql/proektirovanie-baz-dannyx-informacionnaya-izbytochnost-izbytochnost-dannyx-v-baze-dannyx-problemy-voznikayushhie-iz-za-informacionnoj-izbytochnosti.html> (дата звернення: 02.05.2020).

17 Сервіс для створення Use-case діаграм : Веб-сайт. URL: <https://online.visual-paradigm.com/app/diagrams/#diagram:proj=0&type=StateMachineDiagram&gallery=/repository/a481a001-4075-47dc-b968-c8c191f88dad.xml&name=Orthogonal%20State> (дата звернення: 06.05.2020).

18 RoundCornerProgressBar : Веб-сайт. URL: <https://github.com/akexorcist/RoundCornerProgressBar> (дата звернення: 10.04.2020).

19 Опис терміну «Категорія» : Веб-сайт. URL: <https://uk.wikipedia.org/wiki/Категорія> (дата звернення: 18.05.2020).

20 Сервіс для підрахунку кількості символів тексту : Веб-сайт. URL: <https://text.ru/seo> (дата звернення: 18.05.2020).

21 Сервіс для побудови uml діаграм : Веб-сайт. URL: <https://www.genmymodel.com/class-diagram-online> (дата звернення: 20.05.2020).

22 Керівництво як будувати uml діаграми в середовищах IntelliJIDEA та Android Studio : Веб-сайт. URL: <https://stackoverflow.com/questions/8942751/use-intellij-to-generate-class-diagram> (дата звернення: 20.05.2020).

23 Aris Express : Веб-сайт. URL: <https://www.ariscommunity.com/aris-express> (дата звернення: 02.05.2020).

24 Ramus Educational : Веб-сайт. URL: <https://ramus-educational.software.informer.com/> (дата звернення: 02.05.2020).

25 Керівництво як працювати у Ramus Educational : Веб-сайт. URL: <https://infourok.ru/metodicheskie-ukazaniya-po-vipolneniyu-laboratornoy-raboti-sozdanie-funkcionalnoy-modeli-v-ramus-educational-1791311.html> (дата звернення: 02.05.2020).

26 Моделювання у середовищі Aris Express : Веб-сайт. URL:

<https://www.bazt.ru/publications/моделирование-aris-express> (дата звернення: 02.05.2020).

27 Посібник роботи з SQLite DB : Веб-сайт. URL: <https://www.sqlitetutorial.net/> (дата звернення: 02.05.2020).

28 Керівництво по роботі з layout файлами у Android studio : Веб-сайт. URL: <https://startandroid.ru/ru/uroki/vse-uroki-spiskom/14-urok-5-layout-kak-ispolzovat-smena-orientatsii-ekrana.html> (дата звернення: 08.04.2020).

29 Опис xml середовища у Android studio : Веб-сайт. URL: <https://developer.android.com/guide/topics/ui/declaring-layout?hl=ru> (дата звернення: 08.04.2020).

30 Робота з класами адаптерів у Java та Android studio : Веб-сайт. URL: <http://developer.alexanderklimov.ru/android/theory/adapters.php> (дата звернення: 15.05.2020).

31 Керівництво пов'язання Adapter класів зі списками ListView : Веб-сайт. URL: <https://metanit.com/java/android/5.1.php> (дата звернення: 15.05.2020).