

ДОДАТОК А

Специфікація програмного забезпечення

А.1 ВСТУП

А.1.1 Огляд продукту

Це рішення — універсальна фандрейзингова платформа, що складається з веб- та мобільної частин, призначена для швидкого й безпечного збору коштів на благодійні та громадські ініціативи. Система включає:

Користувацький інтерфейс (веб/мобільний) із «one-tap» донатами, компонентним UI, адаптивним дизайном та соціальними функціями (share-кнопки, лідерборди донорів).

Серверну частину, яка забезпечує реєстрацію й автентифікацію користувачів, управління ініціативами й кампаніями, облік донатів, сповіщення та звітність.

Адміністративний модуль для модерації кампаній, аналітики та контролю безпеки.

А.1.2 Мета

Забезпечити донорів зручним, прозорим та безпечним інструментом для пожертв.

Підвищити конверсію завдяки мінімальній кількості кроків у процесі донату та миттєвим оновленням прогрес-барів.

Дати організаторам і адміністраторам потужні засоби модерації та аналітики, включно з “Топ-донорами”, графіками й деталізованими звітами.

Підтримати вимоги безпеки (JWT/OAuth2, захист від OWASP Top 10), регуляторні стандарти (GDPR, PSD2/SCA, PCI DSS) і локальні норми України.

А.1.3 Межі

Входить до системи:

Реєстрація, верифікація та авторизація користувачів (email/пароль, Google OAuth, 2FA).

CRUD-операції над ініціативами та кампаніями; створення донатів й обробка платежів через Stripe, LiqPay, WayForPay.

Система асинхронних сповіщень (email, push) із чергами й retry-механізмом.

Адмін-панель для модерації, аналітики та експорту звітів (CSV/JSON/PDF).

Не входить до системи:

Обробка офлайн-платежів (готівка тощо).

Розробка сторонніх SDK — лише інтеграція з існуючими платіжними провайдерами.

Автоматизоване сканування шахрайських ініціатив із машинним навчанням (планується в майбутніх версіях).

А.1.4 Посилання

Стандарти та специфікації:

- RFC 7519 (JSON Web Token)
- OAuth 2.0 Specification
- OWASP Top 10
- PCI DSS v4.0, PSD2 Technical Standards
- GDPR Regulation (EU) 2016/679

Інструменти та технології:

- React 17+ / Next.js;

- Kotlin + Jetpack Compose;
- .NET 6 / ASP.NET Core Web API або Node.js/Spring Boot;
- PostgreSQL 14+, EF Core або еквівалент ORM;
- Docker, Kubernetes, Helm.

А.2 ЗАГАЛЬНИЙ ОПИС

А.2.1 Перспективи продукту

Фандрейзингова платформа поєднує веб- та мобільні клієнти з потужною серверною частиною, що дозволяє:

Швидкий запуск кампаній – організатори створюють і публікують збір коштів за лічені хвилини, задаючи мету, дедлайн та категорію.

Масштабованість – автоматичне горизонтальне масштабування мікросервісів у Kubernetes для обробки пікових навантажень (тисячі RPS).

Універсальність інтеграцій – підтримка основних платіжних провайдерів (Stripe, LiqPay, WayForPay), Google OAuth, SMTP, FCM.

Аналітика й звіти в реальному часі – графіки прогресу, “Топ-донори”, динамічна статистика, щоденний та місячний звіт у PDF/CSV/JSON.

Гнучкість розгортання – Docker-контейнери, Helm-чарти, мультизональне розгортання PostgreSQL із резервним копіюванням.

А.2.2 Функції продукту

Управління користувачами

Реєстрація, верифікація email, авторизація (email/пароль, Google OAuth, 2FA).

Особистий кабінет із відображенням історії донатів, налаштуваннями сповіщень і профілю.

CRUD ініціатив і кампаній

Створення, редагування, видалення та перегляд ініціатив із описом, картинками, категоріями.

Запуск кампаній (Fundraising) із вибором валюти, цілі, дедлайну.

Процес донату

Інтерфейс “one-tap” (веб/мобільний) із миттєвим оновленням прогрес-бару через WebSocket/EventStream.

Вибір суми, способу оплати, обробка успіху/помилки, автоматичне автозаповнення реквізитів.

Система сповіщень

Підписка на оновлення кампаній, асинхронні черги для email (SMTP) та push (FCM/APNs).

Налаштування частоти й каналів, retry-механізм, логування всіх відправлених повідомлень.

Адміністративний модуль

Моніторинг і модерація ініціатив: фільтри за категоріями, сумами, кількістю донатів.

Блокування шахрайських кампаній, перегляд детальної статистики, управління ролями.

Аналітика та звіти

REST-ендпоінти для отримання агрегованих даних: загальна сума, DailyIncomes, TopDonors.

A.2.3. Характеристики користувачів

Донори: новачки й постійні благодійники, очікують простий та швидкий UX, прозору звітність і різноманітні канали сповіщень.

Організатори: соціальні активісти та НГО, потребують гнучких інструментів створення кампаній, аналітики та експорту даних.

Адміністратори: оператори платформи, відповідальні за безпеку, модерацию та підтримку користувачів, потребують детальних логів і фільтрів.

Гості: можуть переглядати ініціативи й кампанії без реєстрації, ознайомлюватися з прогресом зборів.

A.2.4. Загальні обмеження

Технічні: підтримка сучасних браузерів (Chrome, Firefox, Safari, Edge) і мобільних OS (Android 8+, iOS 13+).

Продуктові: не передбачено обробку готівкових платежів, функції машинного навчання для виявлення шахрайства чи рекомендацій кампаній.

Регуляторні: дотримання обмежень GDPR, PCI DSS, PSD2/SCA; обмежена локалізація наразі лише для UA/EN.

A.2.5 Припущення і залежності

Платформа інтегрується лише з сертифікованими платіжними провайдерами, жодне платіжне рішення не розробляється власноруч.

Наявність стабільного інтернет-зв'язку для клієнтів і серверів; автономний режим (offline) обмежений кешуванням в IndexedDB/Room.

Використання зовнішніх сервісів (Google OAuth, FCM, Stripe), SLA та політики яких впливають на доступність і затримки повідомлень.

A.3 КОНКРЕТНІ ВИМОГИ

A.3.1 Вимоги до зовнішніх інтерфейсів

A.3.1.1 Інтерфейс користувача (UI)

Веб-версія: адаптивний дизайн (desktop/mobile), мінімалістичний макет, однаково зручний для Chrome, Firefox, Safari, Edge.

Мобільні застосунки: Android 8+ та iOS 13+, реалізація нативних елементів (Jetpack Compose, SwiftUI) та “one-tap” донатів.

Компоненти: картки ініціатив, прогрес-бар, модальні діалоги, форми (реєстрація, донат, налаштування).

Доступність: дотримання WCAG 2.1 (контраст тексту, навігація клавіатурою, ARIA-атрибути).

A.3.1.2 Апаратний інтерфейс (Hardware)

Сервери: x86_64-інстанси з ≥ 4 vCPU, ≥ 8 GiB RAM; SSD-диски для баз даних з IOPS ≥ 3000 .

Мережа: 1 Gbps канал, балансування навантаження (NGINX/ALB) із SSL-термінацією.

Пристрої клієнтів: екран ≥ 320 px, підтримка сенсорного управління.

A.3.1.3. Програмний інтерфейс (API)

REST API:

- Стандарт JSON over HTTPS, версії v1/v2 в URI.
- Автогенерація документації Swagger/OpenAPI.

Кінцеві точки:

- /auth/*, /users/*, /initiatives/*, /fundraisings/*, /donate, /subscriptions/*, /admin/*.
- Формати обміну: JSON (RFC 8259), ISO 8601 для дат, стандартні HTTP-коди (200, 201, 400, 401, 404, 500).
- gRPC (опціонально): для внутрішніх мікросервісів із Protobuf v3.

A.3.1.4 Комунікаційний протокол

HTTPS/TLS: мінімум TLS 1.2, сертифікати Let's Encrypt або еквівалент.

WebSocket / Server-Sent Events: для реального часу оновлень прогрес-барів і лічильників «Топ-донорів».

SMTP / FCM / APNs: для email та push-сповіщень; підтримка черг RabbitMQ/Redis Streams.

A.3.2. Властивості програмного продукту

A.3.2.1. Надійність (Reliability)

CLA \geq 99,9 % у мультизональному розгортанні.

Автоматичне резервне копіювання: щоденні знімки БД, інкрементальні кожні 6 годин.

Failover: гарячі резерви для ключових мікросервісів.

Health checks: HTTP /health, інтервали 30 с.

A.3.2.2 Доступність (Availability)

Масштабованість: горизонтальне масштабування мікросервісів у Kubernetes — Fargate або еквівалент.

Кешування: Redis для сесій і часто запитуваних даних; CDN для статичних ресурсів.

Гнучка балансировка: автоматичне додавання/видалення інстансів.

A.3.2.3 Безпека (Security)

Аутентифікація: JWT RS256, OAuth 2.0, Google OAuth, 2FA.

Авторизація: RBAC із ролями User, Organizer, Admin.

Захист від OWASP Top 10: валідація і санітизація вводів, CSRF-токени, HSTS.

Шифрування: AES-256 для чутливих полів у БД, HTTPS-трафік.

Аудит: логування всіх критичних дій із TLS-захищеними лог-сервером.

A.3.2.4 Супроводжуваність (Maintainability)

Чиста архітектура: розділення на шари Domain, Application, Infrastructure, Presentation.

Модульні тести: $\geq 80\%$ покриття (JUnit, Jest, Espresso).

Статичний аналіз: SonarQube, ESLint, StyleCop.

CI/CD: GitHub Actions → build → тестування → Docker image → Helm deployment.

A.3.2.5 Переносимість (Portability)

Контейнеризація: Docker, підтримка локального запуску через Docker Compose.

Конфігурація: 12-factor app (env vars).

Інфраструктура як код: Terraform / Helm Charts.

A.3.2.6 Продуктивність (Performance)

API response ≤ 200 мс при 1000 RPS.

UI P99 ≤ 200 ms для основних екранів.

DB-запити: оптимізовані індекси, EXPLAIN-аналіз.

Load testing: $\geq 10\,000$ одночасних користувачів (Locust/JMeter).

A.3.3 Атрибути програмного продукту

Надійність.

Доступність.

Безпека.

Супроводжуваність.

Переносимість.

Продуктивність.

A.3.4 Вимоги бази даних

Тип СУБД: PostgreSQL 14+ (реляційна), опціонально Redis для кешу.

Модель даних: оптимізована схема з нормалізацією до 3NF; сутності User, Initiative, Fundraising, Donation, NotificationSubscription, Statistics.

Реплікація: master–slaves із автоматичним failover.

Бекапи: повні щоденні, інкрементальні кожні 6 годин; зберігання архівів ≥ 30 днів.

Індекси: на полях, що часто фільтруються (user_id, initiative_id, status, created_at).

Міграції: Flyway / EF Core Migrations із контролем версій.

А.3.5 Інші вимоги

Логування й моніторинг: інтеграція з Prometheus, Grafana, ELK Stack.

Алгоритми резервного копіювання: регулярні dry-run бекапи, звіти про стан.

Документація: User Guide, API Reference (Swagger), архітектурні схеми.

Локалізація: UA/EN з можливістю додати інші мови.

Умови експлуатації: підтримка 24/7 з рішеннями інцидентів згідно з ІТІЛ.

ДОДАТОК Б

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ

Звіт подібності

метадані

Назва організації

Kharkiv National University of Radio Electronics

Заголовок

2025_Б_ПІ_ПЗПІ-21-1_Ляшенко_Є_А_скорочений

Автор

Науковий керівник / Експерт

Ляшенко Євген АндрійовичЄвген Кардаш

підрозділ

каф. ПІ

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Заверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2



6924

Кількість слів

55153

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		0
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		5

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копію тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Копію тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://openarchive.nure.ua/bitstreams/c8537124-abce-4e5e-b953-1112ab802c46/download	20 0.29 %
2	https://openarchive.nure.ua/bitstreams/d8ed8384-ab65-4b31-b933-953c9de1c5c8/download	11 0.16 %
3	https://blog.csdn.net/iamjustnobody/article/details/134673030	9 0.13 %
4	https://plembysystem.com/sealed-classes-in-kotlin-programming-language/	8 0.12 %
5	https://openarchive.nure.ua/bitstreams/c8537124-abce-4e5e-b953-1112ab802c46/download	8 0.12 %

ДОДАТОК В

Слайди презентації



Програмна система для організації збору коштів. Мобільний застосунок. Підсистема роботи з ініціативами та зборами

Ляшенко Є. А. ПЗПІ-21-3
Керівник: ст.викл. кафедри ПІ ВІТАЛІЙ ЛЯПОТА

8 червня 2025



ЗАГАЛЬНА ХАРАКТЕРИСТИКА

Актуальність	У сучасних умовах мобільні застосунки для організації збору коштів перетворюються на ключовий канал взаємодії між організаторами благодійних ініціатив і донорами. Зростаюча конкуренція між платформами та підвищені вимоги користувачів до швидкості, безпеки й прозорості транзакцій обумовлюють необхідність розробки мобільного клієнта з інтуїтивним UX/UI, що підтримує весь цикл: створення й верифікацію ініціатив, запуск кампаній, обробку платежів у реальному часі та сповіщення.
Об'єкт дослідження	Процес розробки програмної системи для організації збору коштів – зокрема мобільної підсистеми роботи з ініціативами та кампаніями зборів.
Предмет дослідження	Моделі та методи реалізації функціоналу створення, редагування й супроводу благодійних ініціатив та зборів у мобільному застосунку.
Мета роботи	Розробити архітектуру та реалізувати мобільний клієнт підсистеми роботи з ініціативами та зборами коштів
Наукова новизна	Поєднання адаптивного UX-підходу з багаторівневими механізмами безпеки транзакцій і real-time комунікації, яке дозволяє знизити час при жертві та підвищити довіру донорів завдяки миттєвому фідбеку й прозорості роботи платформи.
Практичні результати	Реалізовано мобільний застосунок на Kotlin + Jetpack Compose
Реалізований функціонал	<ul style="list-style-type: none">- Розроблено особистий кабінет користувача- Реалізовано систему Push-Notifications- Додано нативну функцію «Поглипитися ініціативою»:- Забезпечено оновлення даних у реальному часі через WebSocket/events для миттєвого відображення змін прогресу збору на екрані деталей ініціативи



АНАЛІЗ ПРОБЛЕМИ

Категорія	Перелік
Досліджені конкуренти	<ul style="list-style-type: none">- GoFundMe- Kickstarter- Indiegogo- Patreon- Monobank "Банка"- Dobro.ua
Прогалини в наявних аналогах ПЗ	<ul style="list-style-type: none">- Непрозорість алгоритмів- Відсутність часових шаблонів поведінки- Єдина «усереднена» модель користувача- Обмежена інтеграція з детальними логами- Відсутність інтерпретованих пояснень- Монолітна архітектура без окремого модуля моделювання- Недостатня адаптивність UI під індивідуальні патерни користувача

ПОСТАНОВКА ЗАДАЧІ

🎯 Мета:

Створити мобільний застосунок для збору коштів з підсистемою ініціатив та зборів.

⚠️ Проблема:

Існуючі рішення розпорозені, складні та непрозорі для користувачів.

🔍 Завдання:

- Модуль створення ініціатив із верифікацією
- Підсистема запуску й адміністрування зборів
- Інтеграція з платіжними шлюзами
- Сервіс real-time сповіщень

Результат:

Швидкий, безпечний і прозорий UX для організаторів і донорів.

ВИКОРИСТАНІ СЕРВІСИ

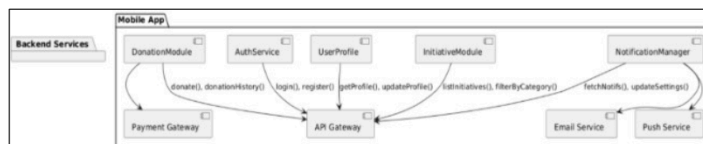
Інструмент / Сервіс	Використання
Stripe SDK	Створення та обробка PaymentIntent, інтеграція Apple/Google Pay
LiqPay	Альтернативний платіжний шлюз
API Gateway	Маршрутизація та авторизація запитів між клієнтом і мікросервісами
Push service	Надсилання push-сповіщень (Firebase Cloud Messaging)
Email service	Доставлення email-сповіщень щодо зборів та статусів
Git + GitHub	Контроль версій та code review

ВИКОРИСТАНІ ТЕХНОЛОГІЇ, ФРЕЙМВОРКИ ТА МЕТОДОЛОГІЇ

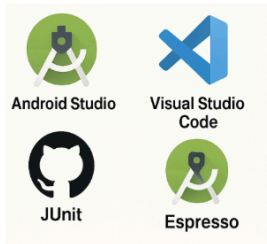
Компонент	Технологія	Опис
Мова програмування	Kotlin	Сучасна статично типізована мова від JetBrains, рекомендована Google для Android-розробки
UI-фреймворк	Jetpack Compose	Декларативний UI-фреймворк для створення реактивних інтерфейсів на Kotlin
HTTP-клієнт	Retrofit	Безпечний HTTP-клієнт від Square для роботи з REST-API
Push-сповіщення	Firebase Cloud Messaging (FCM)	Безкоштовне крос-платформене рішення для push-повідомлень
Архітектура	MVVM + Kotlin Coroutines	Патерн для асинхронної обробки й реактивного оновлення UI
Методологія розробки	Agile (Sprint Review & Retrospective)	Ітеративний підхід зі спринт-рев'ю та ретроспективами для постійного вдосконалення процесу

АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура мобільного клієнта побудована модульно. AuthService відповідає за логін/реєстрацію через API Gateway; UserProfile — за завантаження й оновлення даних користувача; InitiativeModule — за отримання списку ініціатив і фільтрацію; DonationModule — за створення донатів і перегляд історії через Payment Gateway; NotificationManager — за отримання й налаштування push та email-сповіщень. Усі запити клієнта проходять через єдиний API Gateway, який маршрутує їх до відповідних бекенд-сервісів.



ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЩО БУЛО ВИКОРИСТАНО У ДОСЛІДЖЕННІ



Kotlin, firebase dynamic links, firebase BOM, coil compose, okhttp

ДИЗАЙН СИСТЕМИ

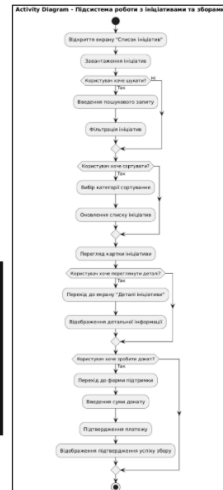
Проектування архітектури мобільної частини системи базується на модульному підході для забезпечення гнучкості, масштабованості та зручності супроводу. Всі запити й відповіді між мобільним клієнтом і бекендом проходять через центральний API Gateway, який забезпечує авторизацію, валідацію та маршрутизацію.

```

com.example.funds
├── presentation
│   ├── navigation.routes
│   └── pages
│       ├── initiative
│       │   ├── Initiative.kt
│       │   ├── InitiativeList.kt
│       │   ├── MainActivity.kt
│       │   └── NavHost.kt
│       ├── userpage
│       │   ├── UserPage.kt
│       └── ui
│           ├── components
│           │   ├── theme
│           │   ├── color.kt
│           │   ├── type.kt
│           └── widgets
└── MainActivity
    
```

```

com.example.funds/
├── navigation.routes/ # маршрутизація екранів
├── pages/ # UI екрани
│   ├── initiative/ # деталі ініціативи (Initiative.kt)
│   ├── initiativeList/ # список ініціатив (InitiativeList.kt)
│   ├── mainScreen/ # головний екран з NavHost (MainScreen.kt, NavHost.kt)
│   ├── userpage/ # профіль користувача (UserPage.kt)
│   └── ui/ # стилі в компонентах
├── components/ # повторно використовувані UI компоненти
│   ├── theme/ # кольори та типографіка (color.kt, theme.kt, type.kt)
│   └── widgets/ # додаткові віджети
└── MainActivity.kt # точка входу в додаток
    
```



ОГЛЯД АЛГОРИТМІВ

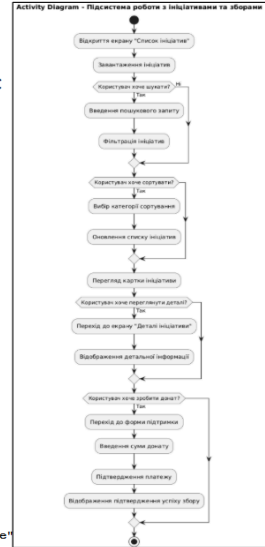
При натисканні кнопки «Пожертвувати» застосунок надсилає суму на сервер, створює платіж у системі банківських операцій, отримує підтвердження аутентифікації і відображає результат користувачу.

```
class DonateStripeViewModel(
    private val repo: DonateStripeRepository
) : ViewModel() {
    val ui = MutableStateFlow<UiState>(UiState.Idle)

    fun launchDonation(
        fundId: Int,
        amount: BigDecimal,
        paymentSheet: PaymentSheet
    ) = viewModelsScope.launch {
        ui.value = UiState.Loading

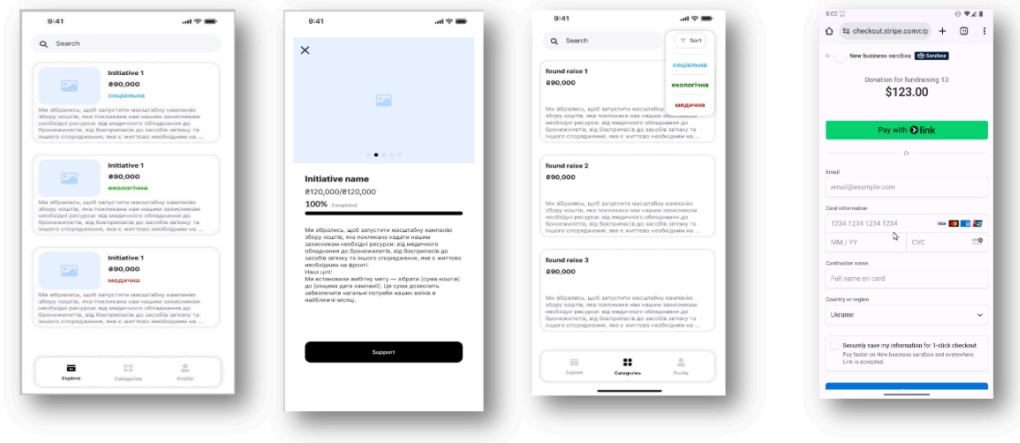
        runCatching {
            val secret = repo.startDonation(fundId, amount)
            ui.value = UiState.AwaitPayment(secret, paymentSheet)
        }.onFailure {
            ui.value = UiState.Error(it.message ?: "Помилка обробки платежу")
        }
    }

    fun onPaymentResult(result: PaymentSheetResult) {
        ui.value = when (result) {
            is PaymentSheetResult.Completed -> UiState.Success
            is PaymentSheetResult.Canceled -> UiState.Idle
            is PaymentSheetResult.Failed -> UiState.Error(result.errorMessage ?: "Помилка Stripe")
        }
    }
}
```



10

ПРИКЛАД РЕАЛІЗАЦІЇ



11

Тестування



Інформація про тест-кейс			
Ідентифікатор тесту:	Тест-кейс №1		
Власник тесту:	Лашенко Євген Андрійович		
Дата створення:	23.05.2025		
Мета тесту:	APK у режимі debug встановлений на Android 13; безпека та тестовий ключ Stripe активні; користувач авторизований (JWT у Secure Store).		
Перегляд і фільтрація ініціатив			
№	Опис випадку	Очікуваний результат	Висновок
1	Відкрити головний екран «Initiatives»	Листовий список ініціатив завантажено, у top-bar видно пошук і кнопку «Sort»	Пройдено
2	Ввести в пошук частину назви існуючої ініціативи	Список звужується до записів, що містять підрдок; інші зникають	Пройдено
3	Натиснути «Sort» → вибрати категорію medical	Відображаються лише ініціативи з категорією «medical»; бейдж категорії підсвічено	Пройдено
4	Свайп-гебей при вимкненому фільтрі	Дані перезавантажено; активний фільтр не зникає	Пройдено
5	Тап по картці ініціативи	Відкриваються деталі: фото, опис, прогрес-бар, кнопка «Donate»	Пройдено
6	Згорнути екран деталей «+» та повернутися до списку	Позиція продуктів збережена; картка виділена тонкими обводою	Пройдено
7	Активувати офлайн-режим (Wi-Fi OFF) та відкрити список	Індикаційний ікон показує останні 10 ініціатив, snackbar «Offline mode»	Пройдено
8	Спроба свайп-гебей в офлайн-режимі	Тост «No internet»; список не змінюється	Пройдено

№	Опис випадку	Очікуваний результат	Висновок
9	Глибока лінка app://initiative?id=42	Додаток відкриває деталі ініціативи #42, навіть якщо застосунок був закритий	Пройдено
10	Вийти з облікового запису → знову відкрити «Initiatives» → Донат через Stripe	Список доступний для гостей, але кнопка «Create» зникає ре та оновлення прогресу	Пройдено
№	Опис випадку	Очікуваний результат	Висновок
1	На деталях ініціативи натиснути «Donate»	З'являється слайд-ап форма вибору суми	Пройдено
2	Ввести валідну суму (≥1) та «Pay»	Відкривається Stripe PaymentSheet (тестова карта 4242...)	Пройдено
3	Завершити платіж успішно	PaymentSheet показує Success, діалог закривається	Пройдено
4	Повернутись на екран, спостерігати прогрес	Прогрес-бар і сума збільшились без перезавантаження (подія WebSocket)	Пройдено
5	В історії донатів (Profile → My Donations) з'явиться запис із сьогоднішньою датою	Запис містить ід ініціативи, суму, статус «Succeeded»	Пройдено
6	Повторити крок 1, але натиснути «X» у PaymentSheet	Платіж скасовано; стам кампанії не змінено; у історії запис відсутній	Пройдено
7	Спроба долати зі значенням 0 або «-5» грн	Кнопка «Pay» неактивна; під полем помилка «Min 1 UAH»	Пройдено
8	У процесі оплати відключити інтернет перед підтвердженням	PaymentSheet показує помилку мережі; ViewState переходить у ViewState.Error	Пройдено
9	Отримати webhook payment_intent.succeeded із загальною >30 €	Після приходу події прогрес-бар оновлюється навіть якщо екран був у фоні	Пройдено
10	Донатити суму, що перевищує goalAmount	Після вебхука кампанія стає «Completed»; кнопка «Donate» прихована	Пройдено

12

ПІДСУМКИ

Розроблений мобільний застосунок показав стабільну роботу в реальних умовах і підтвердив свою практичну користь. Завдяки модульній архітектурі та інтуїтивному UI/UX його легко інтегрувати в існуючі благодійні платформи для швидкого запуску кампаній і прозорості донорських внесків.

У майбутньому можливе розширення функціоналу — випуск iOS-версії, підтримка кількох мов, впровадження AI-аналітики рекомендацій та інтеграція додаткових платіжних провайдерів



13