Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

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

Виконав:
здобувач 4 року навчання
групи ПЗПІ-21-1

Євген ЛЯШЕНКО
(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Програмна інженерія
(повна назва освітньої програми)

Керівник ст.викл. кафедри ПІ Віталій ЛЯПОТА
(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри _____ Кирило СМЕЛЯКОВ
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти перший (бакалаврський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми Освітньо-професійна
Освітня програма Програмна Інженерія
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Ляшенку Євгену Андрійовичу
(прізвище, ім’я, по батькові)

1. Тема роботи Програмна система для організації збору коштів. Мобільний застосунок. Підсистема роботи з ініціативами та зборами
Затверджена наказом по університету від 19.05.2025 р. № 397 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 19.06. 2025
3. Вихідні дані до роботи Розробити мобільний застосунок, що забезпечує управління ініціативами, створення зборів, облік пожертв та інтеграцію з платіжними платформами, використовувати мову програмування Kotlin
4. Перелік питань, що потрібно опрацювати в роботі: Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, висновки, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	08.04.2025	*виконано*
2	Створення специфікації ПЗ	13.04.2025	*виконано*
3	Проектування ПЗ	16.04.2025	*виконано*
4	Розробка ПЗ	17.04.2025	*виконано*
5	Тестування ПЗ	22.05.2025	*виконано*
6	Оформлення пояснювальної записки	01.06.2025	*виконано*
7	Підготовка презентації та доповіді	02.06.2025	*виконано*
8	Попередній захист	18.06.2025	*виконано*
9	Нормоконтроль, рецензування	18.06.2025	*виконано*
10	Здача роботи у електронний архів	18.06.2025	*виконано*
11	Допуск до захисту у зав. кафедри	18.06.2025	*виконано*

Дата видачі завдання «08» «квітня» 2025 р.

Здобувач Євген ЛЯШЕНКО
(підпис)

Керівник роботи _____ ст.викл. Віталій ЛЯПОТА
(підпис) (посада, Власне ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра, 56 стор., 6 рис., 1 табл., 14 джерел, 3 додатків.

АРХІТЕКТУРА, АУТЕНТИФІКАЦІЯ, БЕЗПЕКА, ІНТЕГРАЦІЯ, МІКРОСЕРВІСИ, МОБІЛЬНІСТЬ, ПЛАТЕЖІ, СПОВІЩЕННЯ.

Метою роботи є створення мобільного застосунку для організації збору коштів із підсистемою управління ініціативами та кампаніями. На основі аналізу предметної області сформовано вимоги до ключових сервісів: модулу створення ініціатив із мінімальною формою і верифікацією організаторів, модуля запуску кампаній із параметрами цілі, термінів і реального часу оновлення прогресу, модуля безпечної обробки платежів із інтеграцією банківських сервісів і підтримкою двофакторної автентифікації, сервісу сповіщень для надсилання push- та електронних повідомлень і сервісу автентифікації та розподілу прав доступу за ролями. Інтерфейс застосунку забезпечує швидкий перегляд та фільтрацію ініціатив, детальний огляд кампаній і зручний процес донейту з вибором суми та способу оплати.

Для забезпечення надійності та відмовостійкості реалізовано автоматичне тестування, безперервну інтеграцію та розгортання, моніторинг показників системи та регулярне резервне копіювання даних.

Розроблена система поєднує інтуїтивність користувацького досвіду, гнучкість налаштувань і високий рівень безпеки, що дозволяє ефективно й прозоро проводити благодійні збори в мобільному середовищі.

ABSTRACT

ARCHITECTURE, AUTHENTICATION, SECURITY, MOBILITY, MICROSERVICES, PAYMENTS, NOTIFICATIONS, INTEGRATION.

The aim of this work is to develop a mobile application for fundraising organization with a subsystem for managing initiatives and campaigns. Based on an analysis of the subject area, requirements were defined for the key services: an initiative creation module with a minimal form and organizer verification, a campaign launch module with target parameters, deadlines, and real-time progress updates, a secure payment processing module with integration of banking services and support for two-factor authentication, a notification service for sending push and email messages, and an authentication and role-based access control service. The application interface enables fast browsing and filtering of initiatives, detailed campaign views, and a convenient donation process with customizable amounts and payment methods.

To ensure reliability and fault tolerance, automated testing, continuous integration and deployment, system metrics monitoring, and regular data backups have been implemented.

The developed system combines intuitive user experience, flexible configuration, and a high level of security, enabling efficient and transparent fundraising in a mobile environment.

ЗМІСТ

Вступ 7
1 Аналіз предметної галузі 8
1.1 Особливості управління ініціативами та фондами у мобільних фандрейзингових системах 8
1.2 Моделі руху коштів в мобільній підсистемі управління ініціативами та фондами 12
1.3 Постановка задачі 17
2 Формування вимог до програмної системи 19
3 Архітектура та проєктування програмного забезпечення 21
3.1 Проєктування архітектури ПЗ 22
3.2 Бібліотеки розробки застосунку 24
3.3 Проектування UI/UX дизайну 26
4 Опис прийнятих програмних рішень 32
4.1 Файлова структура проекту 32
4.2 Діаграма діяльності 33
4.3 Цікаві методи та алгоритми 34
5 Тестування програмного забезпечення 38
Висновки 40
Перелік джерел посилання 41
Додаток А Специфікація програмного забезпечення 43
Додаток Б Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ 51
Додаток В Слайди презентації 52

ВСТУП

Сучасні мобільні застосунки для організації збору коштів давно вийшли за межі пасивного перегляду кампаній і дедалі частіше перетворюються на інструмент оперативного запуску ініціатив з повним циклом управління. Саме мобільні інтерфейси сьогодні задають тон взаємодії між організаторами кампаній та донорами – від першого натискання кнопки «створити ініціативу» до перегляду звіту про витрати. В умовах зростання конкуренції між платформами та підвищеної чутливості користувачів до безпеки і прозорості.

Актуальність теми обумовлена необхідністю розробки мобільного клієнта, який дозволяє не тільки здійснювати пожертви, а й повноцінно керувати ініціативами, створювати нові збори, оновлювати дані, відстежувати динаміку надходжень і комунікувати з аудиторією в режимі реального часу. Така підсистема має враховувати не лише потреби користувачів, а й технічні, юридичні та UX-виклики: адаптивність, верифікацію, підтримку багатомовності та відповідність законодавчим нормам щодо захисту даних.

Метою курсової роботи є створення архітектури та реалізація мобільного клієнта підсистеми ініціатив і зборів, який забезпечує зручну навігацію, швидке створення та редагування компаній, стабільну обробку пожертв, персоналізовану стрічку зборів і надійний зворотний зв’язок. Особлива увага приділяється модульності структури, безпеці даних і реалізації інтерфейсів, які максимально відповідають очікуванням сучасного користувача.

Об’єкт дослідження – процес мобільної взаємодії з кампаніями збору коштів, предмет – моделі реалізації підсистеми управління ініціативами у мобільному застосунку. Новизна роботи полягає в поєднанні інструментів адаптивного UX, безпечних транзакційних флоу та логіки створення і супроводу зборів у вигляді єдиного компактного рішення, придатного для масштабування. Практичне значення полягає у формуванні технічної основи, яка може бути використана як стартапами, так і некомерційними організаціями для ефективної організації благодійних кампаній.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Особливості управління ініціативами та фондами у мобільних фандрейзингових системах

Коли розмова заходить про мобільний додаток, який дозволяє не просто кидати «копійчину в банку», а повноцінно запускати й адмініструвати власні ініціативи, перше, що видно неозброєним оком, – це кардинальний зсув від вітринного веб-краудфандинґу до кишенькової студії мікропідприємництва. За останні кілька років мобільний телефон перетворився з додаткового каналу на головний орган керування грошовими потоками: натискаючи іконку на головному екрані, користувач опиняється в середовищі, де створення кампанії займає менше часу, ніж пошук зарядного дроту. Якщо веб-портал досі схожий на форму податкової декларації, то мобільний застосунок грає у швидку драматургію: короткий опис, обкладинка, мета, ліміт часу, і через дві хвилини сторінка збоку вже блимає першим внеском від сусіда з чату.

У глобальному розрізі драйверами цього зсуву слугують і старі веб-гіганти, і нові мобільні спецназівці. Відомий майданчик Kickstarter люб’язно демонструє статистику, де абсолютна більшість успішних проєктів збирає до десяти тисяч доларів, проте кожного року виростає помітний пласт шести- та семизначних кейсів, і левова частка трафіку на ці сторінки вже приходить зі смартфонів [1]. Паралельно GoFundMe звітує, що шістдесят відсотків пожертв у дві тисячі двадцять четвертому році зроблено саме з мобільних пристроїв, а це означає: без оновленого мобільного UX шансів на велике зростання в публічних ініціатив замало. Той, хто досі змушує організатора відкривати ноутбук заради кнопки «Створити кампанію», фактично дарує конкурентам половину ринку.

В Україні найяскравіший приклад миттєвої ініціативи – «банка» від monobank. Буквально один фінт пальцем і перед вами нова сторінка збору, яка поширюється через скопійований лінк швидше, ніж встигає охолонути кава. Цей формат довів, що навіть суто мобільний інструмент може акумулювати мільйони гривень за лічені години: кейс із жартівливим «збором на ядерну боєголовку»

набрав двадцять чотири мільйони гривень за добу і залишив по собі не тільки новий рекорд, а й гібридний фольклор для всіх наступних волонтерів [2]. Наявність готового, майже ігрового сценарію «створити – зашерити – спостерігати, як цифра росте» виявилася сильнішим мотиватором, ніж традиційні рекламні кампанії.

Поруч із «банкою» позиціонується dobro.ua, яке, попри класичне веб-коріння, теж активно випускає мобільні оновлення і хизується цифрою у понад вісімсот двадцять дев’ять мільйонів гривень, акумульованих на майже дев’ять тисяч проєктів. У цьому кейсі видно, що зростання каталогу ініціатив поєднується з постійною спробою спростити мобільний борд для організатора: що коротша форма, то швидше на платформі народжується свіжа кампанія; що більше перевіреної інформації автоматично підтягнено з відкритих реєстрів, то менше підозр виникає у донора, який гортає стрічку проєктів дорогою на роботу [3]. Надійна верифікація й видимий бейдж «ініціатива перевірена» тепер важать для конверсії майже стільки ж, скільки вдало підібране фото на обкладинці.

Інтерфейс створення ініціативи в успішному мобільному застосунку більше нагадує редактор сторіс, ніж банківську форму. Скрипт покроково пропонує заголовок, короткий опис, високий обкладковий кадр, цільову суму і строк, а далі вмикається попередній перегляд, де майбутній організатор уперше бачить свій проєкт так, як побачать донори. Редактори UX-копі доповнюють усе підказками рівня «замініть загальне речення на конкретику: не просто «допомогти армії», а «купити тепловізор Pulsar Helion 2 XP50». Уся магія полягає в тому, щоб процес закінчився ще до того, як з’явиться сумнів: «а чи не надто багато полів я заповнюю». Попередні дослідження платформи GoFundMe показали, що надмірна деталізація на старті зменшує ймовірність запуску збору майже на третину, тому мобільний застосунок обмежує перший екран п’ятьма-шістьма ввідними компонентами, перекидаючи ґрунтовний опис на етап після публікації, коли адреналін уже зробив свою справу і ініціатива живе власним життям.

Після старту починається боротьба за довіру. Усі великі майданчики, від Kickstarter до місцевих волонтерських платформ, доходять до однієї точки: чим

швидше з’явиться підтвердження руху грошей, тим охочіше донори доповнюють баланс. GoFundMe свого часу мусило публічно коментувати хибні чутки про те, що значна частина пожертв йде на комісію, і навіть уточнювати, що дев’яносто сім відсотків суми потрапляє до адресних бенефіціарів, а решта – це транзакційний збір, який ніяк не «з’їдає» пожертви. Така прозорість стала не просто елементом PR, а необхідною відповіддю на потенційні атаки недовіри, адже одна хвиля негативних постів може знести збір швидше, ніж будь-яка технічна помилка [4].

Внутрішня логіка мобільної підсистеми «ініціативи-фонди» будується на принципі постійного візуального фідбеку: прогрес-бар зростає в реальному часі, сума цьогомиттєвого внеску підскакує під звукову анімацію, а у стрічку активності підтягуються імена або нікнейми останніх меценатів. Той самий ефект працює і для організатора: додаток шепоче, що варто оновити статус, бо аудиторія без свіжих фото холоне швидше, ніж нічний чай у наметі. За кулісами з цим візуальним балетом синхронізується подієвий стрим: WebSocket-канал, брокер подій і кеш безпечно підгодовують фронт даними так, щоб між натиском «підтвердити платіж» і «побачити, що сума зросла» минула лише одна-дві секунди.

Гроші, що заходять у проєкт, сьогодні дедалі частіше приходять у змішаній формі. До класичних карткових платежів додалися мультивалютні шлюзи, криптовалютні on-ramp-сервіси й навіть локальні ваучери, які можна купити в торговельній мережі. Організатор, який запускає ініціативу з телефоном у руці, очікує побачити в короткому налаштуванні можливість ставити мінімум три валюти, а то й два стейблкоїни. Якщо застосунок цього не дає, він ризикує втратити аудиторію, що користується лише криптогаманцем. Тут платформа вмикає додаткові перевірки: суми понад певний поріг йдуть через KYC, транзакції з підозрілих адрес ставляться в чергу ручної модерації, а у разі надзвичайно великого переказу лунає алерт службі фінмоніторингу, щоби не довелося після факту пояснювати слідчому, чому анонімна адреса принесла двадцять тисяч USDT за одну ніч.

На горизонті управління ініціативами з’являється ще один цікавий гравець – суміщений аналітичний модуль, який радить організатору наступний крок. Платформи на кшталт Patreon уже звітують про чотириразове зростання одноразових покупок після впровадження внутрішніх підказок та чату, що автоматично кидає ідею подарункового контенту зі знижкою. Звіт компанії за двадцять четвертий рік свідчить, що кількість творців, які під’єднали чат-функцію, зросла у п’ять разів, а це прямий маркер того, як швидке зворотне коло комунікації підвищує середній чек і утримання підписників [5]. Благодійні ініціативи в мобільному просторі беруть до уваги цей урок: якщо додаток одразу підказує «запости відео-апдейт, бо активність упала на двадцять відсотків», організатор відчуває, що система працює напару з ним, а не проти нього.

Технічно вся ця машина тримається на поділеній архітектурі: мікросервіс, що відповідає за створення збору, відділений від сервісу платежів, який, у свою чергу, живе окремо від реєстру організаторів, шифрованого згідно з принципом zero-trust. Кожен із них масштабовується горизонтально, щоби витримати ефект телевізійного сюжету, після якого трафік стрибає з тисячі користувачів до сотні тисяч. Фокус на idempotent-ендпоїнти дозволяє запобігти дублю транзакцій, адже один і той самий жест пальцем користувача може випадково відправити повторний запит, коли мережа підвисає в метро. Для фронтенду це невидима магія, проте саме вона гарантує, що донор спише гроші лише раз, а не двічі.

У зоні ризик-менеджменту мобільні платформи постійно балансують: треба швидко запалити ініціативу, але не допустити фальшивий збір. Алгоритми детектують клоновані описи, порівнюють фотографії обкладинок із базою зображень і вмикають ручну перевірку, коли ймовірність шахрайства виходить за поріг. Соцмережеві тренди додають ще один головний біль: хвиля репостів може принести десять тисяч нових відвідувачів протягом п’ятнадцяти хвилин, і якщо бекенд захлинеться, організатор потім тиждень пояснює, чому прогрес-бар зависав саме в розпал аукціону довіри. Тому автомасштабування, кеш-кордони й CDN-шари вже не технічна розкіш, а банальний must-have для репутаційної безпеки.

Дивлячись у найближче завтра, легко помітити, як логіка «ініціатива – це міні-проєкт» трансформується в ідею «ініціатива – це живий продукт із власним життєвим циклом». Поява Passkey-логіна скорочує час входу до секунди, а експерименти з нуль-знаннєвими доказами натякають, що невдовзі організатор підтверджуватиме свою юрисдикцію або статус неприбуткової організації без викладання документів на сервер, просто продемонструвавши математичний факт. До цього додається сегментація аудиторії за інтересами, навчена на мільйонах анонімізованих кліків: система радить організатору не відправляти повідомлення про збір на артилерію тим, хто донатить лише на тварин, і навпаки. Таким чином оптимізується й конверсія, і витрати на push-квоти, адже зайвий байт в Apple Notification Service тепер коштує дорожче, ніж хвилина копірайтера.

Зрештою, успішна підсистема ініціатив і фондів – це добре змазаний конвеєр, де організатор проходить шлях від ідеї до першого внеску швидше, ніж стигне неміцна кава, а донор бачить ефект своїх грошей раніше, ніж встигає з’явитися природній скепсис. Техніка, право і маркетинг обертаються довкола цього емоційного «моменту істини». Якщо додаток із першої ж спроби дає цю емоцію, користувач натисне кнопку «створити нову кампанію» ще не раз, а ринок отримає те, заради чого взагалі вигадував мобільні фінтех-інструменти: максимальну швидкість руху грошей до тих, кому вони потрібні.

1.2 Моделі руху коштів в мобільній підсистемі управління ініціативами та фондами

Усередині мобільного застосунку, який дозволяє запускати власні ініціативи й керувати зібраними коштами, фінансовий потік перетворюється на швидкісний атракціон, де кожна секунда між натисканням кнопки «Підтримати» та зростанням цифри на екрані може принести або втратити довіру. Користувач, привчений до безболісних Apple Pay-платежів за каву, очікує абсолютно такого самого досвіду, коли перераховує сто гривень на тепловізор, а якщо за лаштунками ховається трирівнева валідація, Strong Customer Authentication і додатковий чек списання, то весь цей складний коридор має пролітати непомітно,

щоб не зламати магію однієї дотичної дії. Ключова напруга полягає в тому, що швидкість, якої вимагав маркетинг і UX-дизайн, стикається з тоннами регуляторних бар’єрів, покликаних запобігти шахрайству, відмиванню грошей і банальним помилкам подвійного списання.

Починати доводиться з того, що донорські транзакції в смартфоні сьогодні не обмежуються однією картковою схемою. Користувачі легко перемикаються між банківськими картками, мультимедійними гаманцями, криптовалютними on-ramp-сервісами та, у випадку транснаціональних благодійних ланцюгів, навіть між різними юрисдикційними правилами про податкові пільги. Коли у стрічці з’являється нова ініціатива, платформа повинна миттєво запропонувати доречні методи оплати й непомітно перевірити, чи підпадає транзакція під дію PSD2, а якщо підпадає, чи можна скористатися винятком із Strong Customer Authentication. Європейські аналітики нагадують, що SCA – це правило, а не побажання, а отже мобільний додаток, який приймає платежі від користувача з Франції чи Німеччини, мусить або попросити біометричне підтвердження, або переконатися, що операція проходить під пороговим лімітом і має маркер low-risk exemption. Детальна інструкція для розробників прямо каже: найнадійніший спосіб уникнути відмов – вбудувати 3-D Secure другого покоління, який зміг показати значно менше відсіву транзакцій за рахунок дотекстових підказок і ширшої підтримки мобільних SDK [6].

Якщо поглянути на тренди повторюваного фінансування, одразу видно, що саме підписки формують кістяк стабільних надходжень. Дослідження Neon One з кінця двадцять третього року демонструє разючу різницю: річна утримуваність регулярних жертводавців наближається до восьмидесяти відсотків, тоді як разові донори тримаються вдвічі гірше [7]. Висновок простий: підсистема управління ініціативами повинна з першого дня давати організатору інструмент прозорого рек’юринґ-менеджменту – календар автосписань, гнучке редагування дати й суми, миттєвий відпис без дзвінка в сапорт. Усе це не косметика, а вимога банального вирахування LTV: кожен секундний бар’єр на шляху редагування підписки відбирає майбутнє пожертвування.

Щоб забезпечити таку гнучкість, застосунку потрібна двоярусна фінансова архітектура. На першому ярусі живуть токени платіжних інструментів: картки, банківські мандати, криптогаманці. На другому – логіка кампанії, яка асоціює конкретний токен з конкретною метою, часом нагадуючи спеціальну багаточастинну кишеню в гаманці, де кожен відсік пронумерований і запечатаний від сусіда. Подібне розділення не просто зручно для бухгалтерії, воно ще й відповідає принципу «segregated accounts» у європейських правилах PSD2, адже кошти формально залишаються власністю донора, поки не відправлені на кінцевий escrow-рахунок ініціативи. Будь-яка спроба змішати «баланс кампанії» з власними активами платформи миттєво класифікує компанію як платіжну інституцію і вмикає додаткову ліцензійну бюрократію. Тому сучасний код застосунку тримає над собою дамоклів меч: торкнешся чужих грошей – негайно отримуєш зобов’язання незалежного аудиту, капітальних резервів і повномасштабного PCI DSS L1.

Окремої уваги потребує ініціатива, яка приймає криптовалюту. Приплив стейблкоїнів виглядає привабливо, бо комісія менша за еквайринг, а прозорість блокчейну створює ілюзію ідеального сліду. Проте ілюзія руйнується, коли у справу вступає анти-відмивне законодавство. На початку двадцять четвертого року Рада ЄС і Європарламент погодили пакет жорсткіших правил AML, встановивши граничний ліміт на великі готівкові операції та давши країнам можливість знижувати пороги відповідно до національного ризику [8]. Хоча ці зміни безпосередньо говорять про кеш, вони демонструють загальний рух до обмеження анонімних транзакцій. Для застосунку це означає, що кожне крипто-поповнення понад певну суму автоматично відстежується через сервіс chain-analysis, а якщо адреса має «грею» репутацію, система або блокує платіж, або ставить його на ручний чек. Торкання незаконних коштів загрожує не лише штрафом, а й репутаційним крахом, відновлення після якого може коштувати дорожче, ніж весь маркетинговий бюджет за рік.

Паралельно з євро-правилами існує українська специфіка. Експертні записки парламентської дослідницької служби нагадують, що в українському

правовому полі краудфандинг поки що лишається фрагментованим: частину ринку регулює закон про благодійну діяльність, іншу регулює право електронної комерції; процедури фінмоніторингу регулярно оновлює Нацбанк через свої постанови [9]. Проте навіть у такому випадку національна служба фінансів вимагає звітувати про транзакції від п’ятнадцяти тисяч гривень. Отже платформа, яка приймає українські карткові внески, мусить записувати ідентифікатор донора, гуртувати платежі від одного учасника, рахувати суму за добу і, в разі перевищення, передавати пакет даних у держреєстр фінмоніторингу. Для організатора ці процеси мають бути прозорими: він бачить у дашборді, що десять платежів по дві тисячі склали ризикову суму, і може бути впевнений, що система згенерує правильний csv-репорт.

Продуктивна модель руху коштів неможлива без безперервної телеметрії. У двадцять четвертому році майже кожна велика платформа публікує власну патоку цифр про кількість транзакцій за секунду. GoFundMe у звіті Year in Help повідомляє, що середній темп донатів складає дві операції щосекунди, а перелік повторних донорів досяг однієї третини всієї бази [10].

Варто звернути увагу на ефект мережевої прозорості, який з’являється, коли донор бачить «потойбічну» кухню ініціативи. Чимало платформ додали окрему вкладку «витрати», де вміщують скан-чеків, але мобільна естетика спонукає перейти від сканів до коротких роликів і стрічки фотографій у форматі інстаграм-каруселі. За внутрішньою аналітикою кількох волонтерських проєктів, публікація відео з моментом купівлі піднімає темп нових надходжень у середньому на дванадцять відсотків протягом наступних двадцяти чотирьох годин; офіційні звіти великі, але ними любуються бухгалтери, а не донори. Парадоксально, але для виконання цього емоційного запиту потрібно ще більше паперової роботи: відскануй чек, завантаж pdf-копію, підтягни довідку кур’єра і, головне, переконайся, що жоден uber-вечеря не потрапив під тег «логістика». У підсистемі управління фондами ці дані мають проходити валідацію, перш ніж вийдуть у публіку, інакше випадкова помилка здатна здути кампанію швидше, ніж накрутка ботів.

У самій структурі взаємодії користувачів із грошима спостерігається постійний дрейф від двохролевої моделі «донор-ініціатор» до багатошарового ансамблю. Один і той самий користувач може сьогодні перерахувати п’ятдесят гривень на чужу зборівлю, завтра запустити власну на п’ять тисяч, а післязавтра призначити довірену особу, яка займатиметься закупівлею й логістикою. Мобільний застосунок зобов’язаний дати йому можливість гнучко делегувати права, причому таким чином, щоб права не накладалися й не створювали «кланових» дірок у безпеці. Коли йдеться про гроші, правило найменших привілеїв перестає бути теорією – воно перетворюється на запобіжник від катастрофи. Тому токени авторизації прив’язуються до конкретної піддії: отримати звіт, змінити контакт, підтвердити витрату. Навіть сам організатор, якщо він надає функції бухгалтеру, отримує нотіфікацію з проханням затвердити права, а рішення логуються в audit-trail, який неможливо редагувати в UI.

Ідея полягає в тому, що людина підтверджує свою юридичну правоздатність і податковий статус через математику, а не через jpg-копію паспорта. Криптографічні протоколи zk-SNARK дозволяють довести приналежність до реєстру некомерційних організацій, не видаючи публіці ні назву, ні адресу, тільки цифровий доказ. Якщо регулятор визнає подібну форму, мобільні платформи отримають ще один спосіб мінімізувати зайву обробку персональних даних і водночас підвищать привабливість для донорів.

І нарешті, питання спадковості й довгої пам’яті. Гроші в благодійності – це не лише швидкий «запустив-зібрав-забув»; багато ініціатив перетворюються на багаторічні марафони. Отже система звітності повинна тягнути дані про трансакції не рік, а мінімум шість чи сім, як того вимагає податкове й аудиторське законодавство більшості країн. Водночас ніхто не давав права зберігати зайві особисті поля після того, як вони більше не потрібні. Тому успішна підсистема автоматизує процедуру логічної анонімізації: залишає суми, дати, ідентифікатори escrow-рахунків, але видаляє прямі персональні маркери. Цей процес іде фоном, наче дефрагментація диска, поки донори продовжують бачити свіжі фотографії, а організатор – нові сповіщення про платежі.

У підсумку фінансова й регуляторна логіка мобільної системи «ініціативи – фонди» нагадує балансування на канаті. По один бік стоїть користувач, який звик, що після подвійного тапа застосунок вже стрибає конфетті й пише «ви допомогли». По другий бік – псалми законодавця, який раз на квартал додає ще один обов’язковий крок перевірки та ще одну статтю за порушення даних. Мисливці за найкоротшим шляхом та інженери комплаєнсу грають у довгу шахову партію, і перемагає той продукт, що дає донору відчуття майже ідеальної миттєвості, а ревізору залишає бездоганні логи, шифровані поля і статутні резерви. Саме ці, на перший погляд, непоєднувані вимоги і формують сучасний стандарт мобільної підсистеми управління ініціативами та фондами: миттєвий дотик, надійний захист, максимально довга пам’ять про кожну гривню та кожен благий намір.

1.3 Постановка задачі

У ході даної роботи нами буде досліджено предметну область мобільного фандрейзингу з акцентом на підсистему управління ініціативами та зборами як ключовий компонент сучасних платформ залучення благодійних внесків.

Метою є виявлення вимог до підсистеми, яка забезпечить зручне створення, редагування та супровід благодійних ініціатив, прозорий механізм публікації та адміністрування зборів, а також довіру з боку донорів через доступ до актуальної й верифікованої інформації про перебіг кампаній.

Реалізація проєкту складається з наступних етапів:

- аналіз предметної області мобільного управління ініціативами та пов’язаними зборами;
- формування функціональних та нефункціональних вимог до підсистеми;
- проєктування архітектури мобільного клієнта та побудова логіки навігації між основними екранами;
- створення інтерфейсів для перегляду, створення та редагування ініціатив і зборів;

− реалізація функціоналу підписки, фільтрації та перегляду історії взаємодії користувача;

− тестування, валідація зібраних даних та оцінка ефективності реалізованої підсистеми.

Ця робота спрямована на розробку мобільного клієнта підсистеми ініціатив та зборів, який інтегрує сучасні принципи UX/UI-дизайну, підтримку фільтрації та сортування, перегляд деталізованих профілів зборів, а також функціонал створення ініціатив та контролю зборів коштів, що дозволить вирішити проблеми фрагментованої взаємодії, відсутності довіри та низької залученості користувачів.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

На основі вивчення особливостей мобільного фандрейзингу сформовано основні вимоги до підсистеми, яка відповідає за створення, перегляд та адміністрування ініціатив і пов’язаних з ними зборів у межах мобільного застосунку. Ця підсистема має забезпечувати організаторам максимально простий та швидкий процес запуску кампаній, а користувачам – зручну навігацію, миттєву взаємодію та прозорість використання коштів.

Перш за все, функціональність мобільного додатку повинна охоплювати повний життєвий цикл кампанії: від створення ініціативи, публікації збору та його модерації – до збору пожертв, їх відображення, оновлень і звітності. Застосунок має передбачати підтримку різних ролей: авторів ініціатив, адміністраторів платформи та донорів. Для забезпечення достовірності зборів необхідно впровадити механізм верифікації організаторів – із перевіркою ідентифікаційної інформації перед публікацією кампанії.

Інтерфейс взаємодії повинен бути інтуїтивно зрозумілим, адаптованим під мобільні екрани, з мінімальною кількістю кроків для запуску збору чи здійснення донату. Для авторів важливо мати змогу створити ініціативу за кілька хвилин: коротка назва, опис, обкладинка, сума та термін – усе це має вводитися в межах одного сценарію. Для донорів – миттєвий доступ до найактуальніших зборів, фільтрація за темами, перегляд статусу та прозора історія транзакцій.

Особливу увагу слід приділити механізмам підтвердження внесків. Система повинна підтримувати інтеграцію з локальними та міжнародними платіжними провайдерами, забезпечувати стабільну обробку транзакцій і негайне відображення змін у прогрес-барах кампаній. Кожна пожертва супроводжується квитанцією, яка надсилається користувачеві через обраний канал комунікації – push, email або вбудоване повідомлення.

Для підвищення довіри між сторонами система повинна мати двосторонні комунікаційні функції. Організатори повинні мати змогу публікувати оновлення, ділитися фото чи відео, відповідати на коментарі донорів. У свою чергу, користувачі будуть отримувати повідомлення про перебіг кампаній, а також

залишати відгуки або запитання. Сповіщення мають бути персоналізованими, контрольованими й прозоро керованими зі сторони користувача.

Щодо нефункціональних характеристик, система повинна бути здатна масштабуватись у разі різкого зростання навантаження – наприклад, під час масових кампаній. Рекомендується архітектура з окремими модулями для обробки ініціатив, зборів, платежів і сповіщень, із можливістю горизонтального масштабування. Важливою є також інтеграція з системами моніторингу, резервного копіювання та відновлення після збоїв.

Додаток повинен підтримувати багатомовність, зміну валют, локалізацію дати й часу, що дозволить забезпечити користувацький досвід як для української, так і міжнародної аудиторії. Передбачено підтримку адаптивного інтерфейсу, який коректно працює на телефонах, планшетах і інших мобільних пристроях.

У підсумку, вимоги до підсистеми ініціатив і зборів формуються як сукупність функціональних, нефункціональних і UX-орієнтованих характеристик, які мають забезпечити простоту використання, довіру до платформи, технічну стійкість і відповідність законодавчим нормам. Лише гармонійне поєднання цих компонентів забезпечить ефективну роботу мобільного застосунку для збору коштів у сучасному цифровому середовищі.

3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Проєктування архітектури мобільної частини системи для організації збору коштів базується на модульному підході, що забезпечує гнучкість, масштабованість і зручність у супроводі та розширенні функціоналу. Для візуалізації взаємозв'язків між основними компонентами мобільного застосунку було створено UML-діаграму (див. рис. 3.1), яка відображає ключові модулі, їхні обов'язки та інтеграцію із серверною частиною системи.

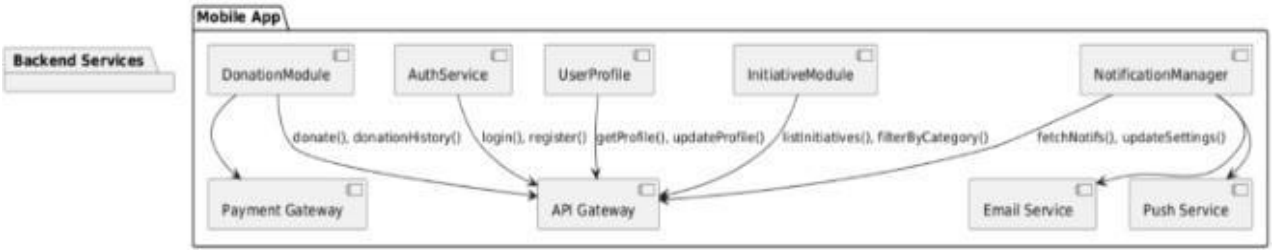

Рисунок 3.1 – UML-діаграма мобільної частини (рисунок виконано самостійно)

Нижче розглянуто кожен елемент більш детально.

AuthService:

- реєстрація та вхід користувачів;
- підтримка автентифікації через email та google oauth;
- взаємодіє з api gateway для обробки запитів на автентифікацію.

UserProfile:

- отримання та редагування особистих даних користувача (ім'я, email, аватар);
- доступ до історії донатів;
- використовує api gateway для взаємодії з сервером.

InitiativeModule:

- перегляд списку ініціатив;
- фільтрація ініціатив за категоріями та ключовими словами;

- взаємодія з api gateway для отримання актуальної інформації про збори.

DonationModule:

- створення донатів до обраних ініціатив;
- перегляд історії транзакцій;
- використання зовнішнього платіжного шлюзу (наприклад, stripe, liqpay) через payment gateway.

NotificationManager:

- отримання push- та email-сповіщень;
- керування налаштуваннями сповіщень користувача;
- зв’язок з push service та email service для надсилання повідомлень.
- API Gateway:
- центральна точка маршрутизації запитів з мобільного застосунку;
- забезпечує авторизацію, валідацію та перенаправлення до відповідних бекенд-сервісів;
- посередник між усіма мобільними модулями та серверною частиною;
- push service / email service / payment gateway;
- push service – надсилання push-нотифікацій користувачам;
- email service – доставка email-повідомлень, зокрема щодо нових зборів та статусів, Payment Gateway – обробка транзакцій через сторонні платіжні сервіси.

3.1 Проєктування архітектури ПЗ

У межах розробки мобільного застосунку, який є складовою програмної системи для організації збору коштів, важливим етапом стало визначення

архітектури підсистеми, що відповідає за роботу з ініціативами та пов’язаними з ними зборами. Архітектура даної підсистеми формується навколо чіткої взаємодії між користувачем і системою збору коштів через набір інтерфейсів (сторінок), кожен з яких реалізує окремий аспект функціональності.

Побудова інтерфейсу мобільного застосунку базується на концепції розділення відповідальностей, де кожна сторінка несе конкретне логічне навантаження, що дозволяє уникнути перевантаження інтерфейсу та забезпечує цільове використання кожного екрану. Це особливо критично в контексті мобільних рішень, де простота, швидкість доступу до інформації та інтуїтивна взаємодія відіграють ключову роль.

У межах підсистеми роботи з ініціативами та зборами визначено наступні основні сторінки мобільного застосунку:

- сторінка авторизації / реєстрації. Цей екран відповідає за початкову взаємодію користувача з системою. підтримується як вхід через email та пароль, так і google oauth. реєстрація доступна виключно для користувачів з роллю «user», оскільки адміністрування відбувається через окрему частину системи. верифікація здійснюється шляхом відправки коду на електронну пошту;
- головна сторінка ініціатив. Основна точка входу до функціоналу зборів. на цьому екрані відображається список доступних ініціатив з основною інформацією: назва, категорія, короткий опис, статус збору коштів та прогрес до цілі. передбачена фільтрація за категоріями (соціальні, медичні, екологічні тощо) та пошук за назвою чи описом;
- сторінка детального перегляду ініціативи. Тут користувач отримує повну інформацію про конкретну ініціативу: її мету, історію, організаторів, термін дії та інші супровідні збори. також на сторінці реалізовано функціонал підписки на оновлення щодо ініціативи, що забезпечує отримання push-сповіщень про нові або змінені збори;

- сторінка збору коштів. Призначена для безпосередньої взаємодії з конкретним збором. користувач може переглянути суму, яка вже зібрана, залишок до мети, термін завершення збору та деталі призначення коштів. інтегрована підтримка платіжних систем дозволяє здійснити донат безпосередньо з цієї сторінки;
- форма донату. Частина сторінки збору або окремий екран, де користувач обирає суму пожертви, платіжну систему (stripe, paypal, liqpay або інші) та підтверджує внесок. цей процес супроводжується відповідною перевіркою та записом транзакції в систему;
- сторінка «мої внески». Дає змогу користувачу переглянути історію своїх пожертв, зокрема дату, суму та збір, до якого було здійснено внесок. також тут можна відстежити статус зборів, до яких користувач долучився, що сприяє підвищенню довіри та залученості;
- сторінка «мої підписки». У цьому розділі зібрано всі активні підписки користувача на ініціативи. за потреби можна змінювати налаштування сповіщень або скасувати підписку.

Усі згадані сторінки об’єднані навігаційною структурою, що забезпечує послідовну та зручну взаємодію. Архітектурно система реалізована за принципом модульності, що дозволяє масштабувати окремі компоненти без потреби повної переробки інтерфейсу.

Таким чином, проєктування архітектури підсистеми роботи з ініціативами та зборами базується на цілісному підході до UX/UI взаємодії та логіки функціонування, орієнтованій на прозорість, зручність і ефективність підтримки зборів через мобільний застосунок.

3.2 Бібліотеки розробки застосунку

Для реалізації мобільного застосунку було обрано мову програмування Kotlin з огляду на низку її переваг, які забезпечують оптимальне поєднання продуктивності розробки, якості коду та загальної ефективності кінцевого програмного продукту. Kotlin є сучасною, повністю сумісною з Java мовою, розробленою з урахуванням потреб сучасних програмних систем, що дозволяє інтегруватися зі значною кількістю вже існуючих бібліотек та фреймворків, створених на платформі Java.

Вибір саме Kotlin як основної мови програмування обумовлений її високим ступенем виразності, що дозволяє значно скоротити обсяг коду, підвищуючи при цьому його читабельність та зрозумілість. Такий підхід зменшує кількість потенційних помилок, покращує підтримуваність проєкту і спрощує командну взаємодію.

Kotlin підтримує сучасні парадигми програмування, такі як функціональний підхід та реактивні концепції, які ідеально поєднуються з актуальними практиками розробки мобільних застосунків, зокрема з використанням Jetpack Compose. Це дозволяє створювати гнучкі та інтерактивні користувацькі інтерфейси з мінімальними витратами на додаткову імплементацію. Крім того, у Kotlin наявні вбудовані засоби управління null-безпекою, що значною мірою зменшує ризик виникнення критичних помилок під час виконання програми, поліпшуючи загальну стабільність застосунку [11].

Ще одним важливим аспектом є активна підтримка Kotlin з боку компанії Google, яка офіційно визнала цю мову як рекомендовану для Android-розробки, що гарантує тривалу підтримку, швидке оновлення та відповідність сучасним стандартам розробки на Android-платформі. Вибір Kotlin як основної мови для реалізації мобільного застосунку дозволив досягти високої продуктивності роботи команди, забезпечити чіткість і структурованість кодової бази та створити передумови для легкого масштабування і подальшого розвитку програмної системи.

Архітектура мобільного застосунку, реалізованого з використанням Kotlin, побудована з урахуванням сучасних принципів розробки програмного забезпечення для платформи Android. В її основі лежить підхід, що передбачає чітке розмежування відповідальностей між різними логічними компонентами, а також використання сучасних бібліотек та інструментів, що надаються екосистемою Android Jetpack. Kotlin, як мова програмування, підтримує парадигми функціонального та об'єктно-орієнтованого програмування, що дозволяє гнучко структурувати архітектуру додатку відповідно до завдань проєкту.

Основним елементом презентаційного шару виступає Jetpack Compose, який надає можливість описувати інтерфейс користувача декларативним способом, тим самим зменшуючи кількість стандартних шаблонних конструкцій та дозволяючи розробникам зосередитись на логіці взаємодії з користувачем. Використання Compose у поєднанні з Kotlin дозволяє створювати модульні, читабельні компоненти, що легко піддаються тестуванню та підтримці. Додатковим елементом інтерфейсної частини є навігація, що реалізується за допомогою бібліотеки Navigation Component, яка дозволяє централізовано та зручно керувати переходами між екранами.

Для управління станом та життєвим циклом екранів застосовується архітектурний патерн MVVM (Model-View-ViewModel). Використання ViewModel з Android Jetpack Lifecycle дозволяє ефективно розмежувати логіку взаємодії з даними від інтерфейсної частини, забезпечуючи збереження стану навіть при зміні конфігурації пристрою (наприклад, при повороті екрана). Крім того, для асинхронної обробки даних і запитів до сервера використовується бібліотека Kotlin Coroutines, яка надає зручні механізми для виконання фонових задач та керування багатопоточністю. Це дозволяє уникнути блокування основного потоку UI, що підвищує загальну продуктивність та швидкість реакції інтерфейсу [12].

Для забезпечення реактивності інтерфейсу використовується StateFlow та LiveData, які дозволяють у режимі реального часу оновлювати стан UI залежно від змін у даних. Ці компоненти органічно поєднуються з Kotlin Coroutines і підтримують чітку та прозору передачу змін між шарами додатку. Для організації

доступу до зовнішніх ресурсів, таких як API, додаток використовує Retrofit у комбінації з бібліотекою Moshi або Kotlin Serialization, що дозволяє ефективно обробляти та парсити дані у форматі JSON.

Таким чином, архітектура оточення, реалізована на базі Kotlin, складається з взаємопов'язаних компонентів, що забезпечують чисту, надійну та легко масштабовану структуру додатку. Цей підхід дозволяє зменшити кількість помилок, покращити продуктивність розробки, спростити подальшу підтримку проєкту та забезпечити високий рівень якості користувацького досвіду.

3.3 Проектування UI/UX дизайну

У рамках мобільного застосунку, підсистеми роботи з ініціативами та зборами, проєктування інтерфейсу користувача (UI) та користувацького досвіду (UX) стало ключовим етапом, спрямованим на досягнення інтуїтивності, емоційної привабливості та зручності взаємодії. Головною метою цього етапу було створити зрозуміле і просте середовище, яке дозволяє користувачам швидко знаходити потрібну інформацію, брати участь у благодійних ініціативах та здійснювати донати з мінімальною кількістю кроків.

Основою роботи нашої частини є зокрема робота безпосередньо зі сборами та ініціативами. Зокрема реалізовані сторінки – головна, з фільтрацією ініціатив, деталі ініціативи з фільтрацією по сборам, сторінки створення ініціативи та сбору, сторінка редагування ініціатив та сборів.

Нижче наведено попередній перелік і призначення реалізованих сторінок. Головна сторінка ініціатив:

- служить точкою входу до всіх доступних ініціатив, яка відображає загальний список актуальних ініціатив із коротким описом, категорією, статусом та індикатором прогресу збору коштів;
- можливий пошук за назвою або ключовими словами;

- фільтрація за категоріями, наприклад: соціальні, медичні, екологічні тощо;
- сортування за популярністю або датою створення.

Сторінка детального перегляду ініціативи:

- надання користувачу повної інформації про обрану ініціативу: опис, автор, тривалість, категорія, кількість підписників;
- кнопка підписки на ініціативу;
- фільтрація усіх зборів, пов’язаних із цією ініціативою (наприклад: активні, завершені, з найбільшою сумою);
- переходи до деталей кожного збору.

Сторінка створення ініціативи:

- доступна для верифікованих користувачів. дозволяє створити нову ініціативу, вказавши її назву, опис, категорію, зображення та тривалість;
- валідація обов’язкових полів;
- підказки щодо формату заповнення;
- відправка на модерацію або активацію одразу.

Сторінка створення збору:

- прив’язана до раніше створеної ініціативи, яка дозволяє автору ініціативи створити новий збір коштів;
- введення цілі збору, суми, тривалості, опису;
- вказівка призначення (на що підуть кошти).

Сторінка редагування ініціативи;

- доступна лише автору ініціативи;
- дозволяє змінювати основну інформацію про ініціативу (назва, опис, категорія, обкладинка).

Сторінка редагування збору:

- дозволяє редагувати активні збори;
- можна змінити дедлайн, ціль (якщо ще не досягнуто).

Особистий кабінет (Basic Profile):

- дозволяє користувачеві переглядати свою історію активності;
- список донатів з датою, сумою, назвою ініціативи;
- перехід до деталей ініціативи або збору;
- (можливе розширення) – редагування профілю, аватарки, налаштування.

Ці сторінки формують основний цикл взаємодії користувача з підсистемою: перегляд, створення, редагування, донат або підтримка. Нижче наведено приклад інтерфейсу перегляду ініціативи та деталей (див.рис. 3.2, 3.3 відповідно).

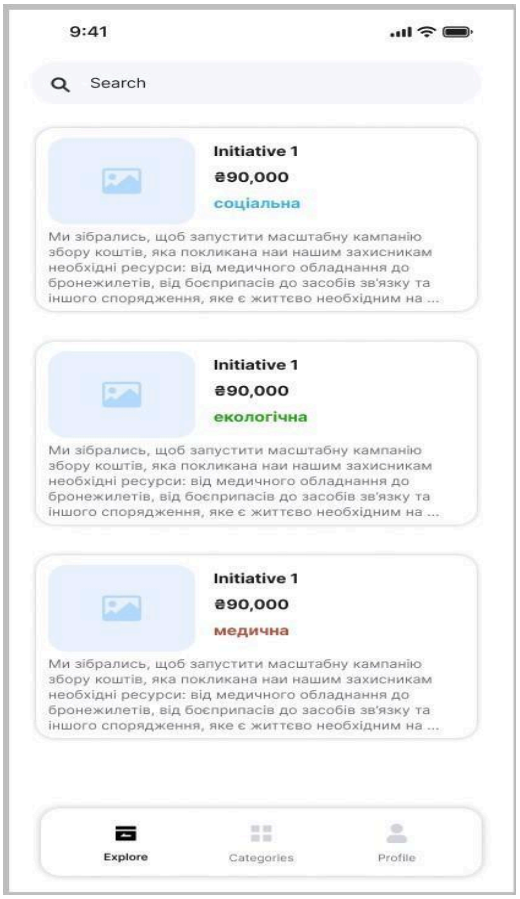

Рисунок 3.2 – Перегляд ініціатив (рисунок виконано самостійно)

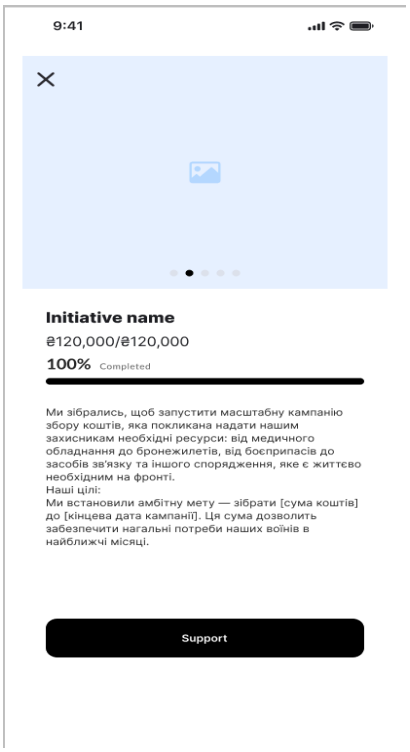

Рисунок 3.3 – Деталі ініціатив (рисунок виконано самостійно)

Також на рисунку 3.4 наведено перегляд зборів ініціатив.

Рисунок 3.4 – Відображення зборів (рисунок виконано самостійно)

Інтерфейс мобільного застосунку, демонструє екран перегляду ініціатив для збору коштів. Він містить пошуковий рядок у верхній частині, що дозволяє користувачу швидко знаходити потрібні кампанії.

Поруч із полем пошуку розташована кнопка сортування, натискання на яку відкриває список категорій ініціатив: соціальна, екологічна та медична. Відображені ініціативи представлені у вигляді карток, кожна з яких містить заголовок, суму збору та короткий опис мети кампанії, що дає користувачу змогу ознайомитися з її суттю ще до відкриття детальної інформації. Нижня навігаційна панель містить три вкладки: Explore, Categories та Profile, що забезпечує зручний доступ до основних розділів застосунку та підвищує загальну юзабіліті інтерфейсу. Дизайн виконано в сучасному мінімалістичному стилі з використанням білого фону та контрастного тексту, що створює чистий та зрозумілий вигляд. Усі елементи інтерфейсу логічно згруповані, що сприяє швидкому орієнтуванню користувача та спрощує процес взаємодії із застосунком.

У межах комплексного проєкту ми відповідали за мобільний клієнт підсистеми ініціатив та зборів. Зосередившись на архітектурі, UI/UX і інтеграції з бекенд-API, ми перетворили складну бізнес-логіку платформи на зручний «кишеньковий» інструмент: було втілено стрічку ініціатив із фільтрами й пошуком, створення та підтримку зборів, а також безпечний платіжний флоу, що дозволяв користувачам здійснювати донати за кілька дотиків.

Завдяки цій роботі мобільний застосунок став єдиним вікном взаємодії: від авторизації й першого скролу стрічки до підтвердження транзакції та перегляду історії внесків. Ми забезпечили прозорість, швидкість і довіру донаторів, що безпосередньо підсилювало ефективність усієї платформи збору коштів.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

4.1 Файлова структура

Далі розглянуто файлову структуру проекту мобільного застосунку (див. рис. 4.1).

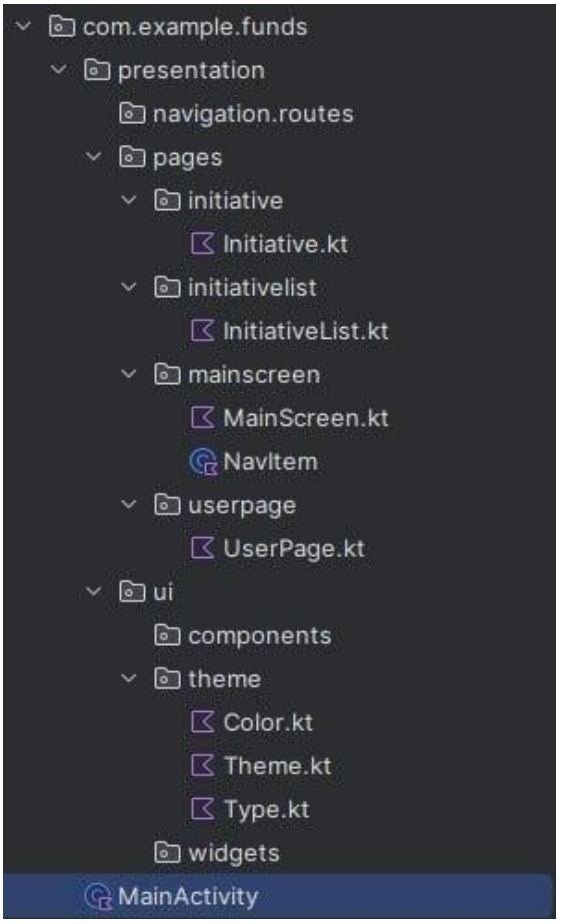

Рисунок 4.1 – Файлова структура (рисунок виконано самостійно)

Файлова структура мобільного застосунку побудована відповідно до принципів модульності, розділення відповідальностей та масштабованості, що характерно для архітектури, орієнтованої на презентаційний шар (presentation-centric structure). Вся логіка інтерфейсу користувача зосереджена в пакеті presentation, який є базовим контейнером для екранних компонентів, навігації, тем і візуальних елементів. Це дозволяє легко орієнтуватися в кодовій базі, спрощує командну роботу та сприяє розширенню функціоналу без порушення існуючої структури.

Усередині presentation реалізовано папку pages, яка згрупована за логікою екранного відображення: окремі директорії для ініціатив, списку ініціатив, головного екрана та профілю користувача. Кожна з них містить ізольовані компоненти, що відповідають лише за свою область, наприклад, Initiative.kt – за перегляд окремої ініціативи, UserPage.kt – за екран користувача тощо.

Папка navigation.routes містить опис усіх маршрутів переходів між екранами, що дозволяє централізовано керувати навігацією. Окремо винесено ui, де зібрано стилі, кольори, типографіку (theme) та повторно використовувані компоненти (components і widgets). Це дозволяє забезпечити єдність візуального стилю та підтримку принципів повторного використання елементів інтерфейсу.

Основна точка входу застосунку – файл MainActivity.kt, у якому ініціалізуються тема, навігація та початковий екран. Такий підхід до архітектури забезпечує чисту, підтримувану структуру проєкту, яка відповідає вимогам сучасної Android-розробки з використанням Jetpack Compose [13].

4.2 Діаграма діяльності

Діаграма діяльності демонструє основний сценарій взаємодії користувача з підсистемою роботи з ініціативами та зборами у мобільному застосунку. Починається процес з відкриття екрану списку ініціатив, після чого здійснюється завантаження актуального переліку доступних кампаній. Користувач може скористатися пошуком або фільтрацією для зручного знаходження потрібної ініціативи, вводячи запит у відповідне поле чи обираючи категорію сортування зі списку. Після перегляду карток ініціатив користувач може перейти до деталізації конкретної ініціативи, де відображається розширена інформація про збір.

Розглянемо діаграму діяльності для підсистема роботи з ініціативами та зборами (див. рис. 4.2).

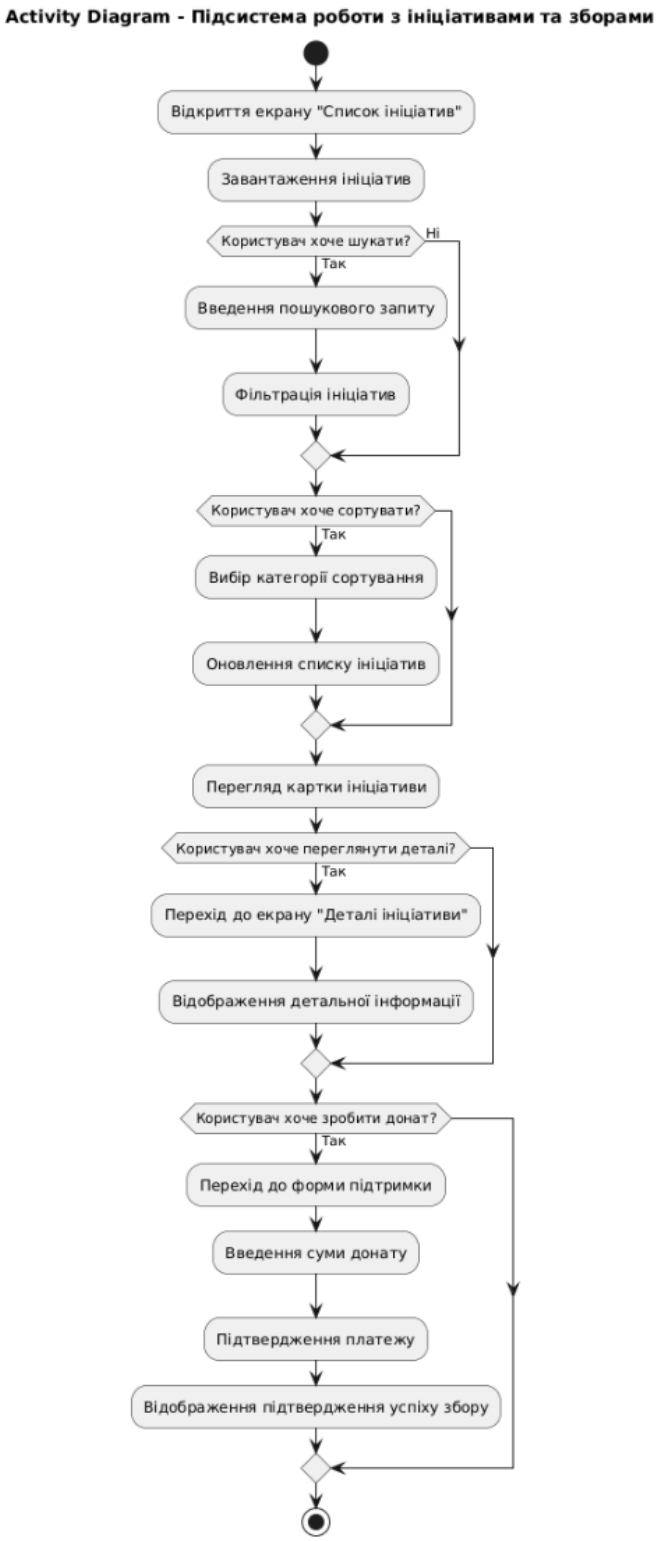

Рисунок 4.2 – Діаграма діяльності (рисунок виконано самостійно)

Якщо користувач бажає підтримати ініціативу, він має можливість ввести суму донату, підтвердити платіж та побачити повідомлення про успішне завершення операції. Уся послідовність дій чітко структурована і забезпечує

інтуїтивно зрозумілу навігацію, з урахуванням варіативності вибору користувача та можливості швидкого доступу до ключових функцій застосунку.

4.3 Цікаві методи та алгоритми

Нижче розглянуто звернення до серверної частини застосунку та адаптація обробки платежу за допомогою платіжної системи Stripe.

```
package com.example.funds.donate
import com.squareup.moshi.JsonClass
import java.math.BigDecimal
@JsonClass(generateAdapter = true)
data class DonateRequest(val amount: BigDecimal)
@JsonClass(generateAdapter = true)
data class PaymentIntentResponse(val clientSecret: String)
import retrofit2.http.Body
import retrofit2.http.POST
import retrofit2.http.Path
interface FundraisingStripeApi {
    @POST("{id}/donate")
    suspend fun createPaymentIntent(
        @Path("id") fundraisingId: Int,
        @Body donateRequest: DonateRequest
    ): PaymentIntentResponse}
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.withContext
class DonateStripeRepository(
    private val api: FundraisingStripeApi) {
    suspend fun startDonation(id: Int, amount: BigDecimal): String =
        withContext(Dispatchers.IO) {
                                              api.createPaymentIntent(id,
DonateRequest(amount)).clientSecret
        }}
import androidx.{…}
class DonateStripeViewModel(
    private val repo: DonateStripeRepository
) : ViewModel() {
    val ui = MutableStateFlow<UiState>(UiState.Idle)
    fun launchDonation(
```

```
            fundId: Int,
            amount: BigDecimal,
            paymentSheet: PaymentSheet
        ) = viewModelScope.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 ?: "Unknown error")}}
        fun onPaymentResult(result: PaymentSheetResult) {
            ui.value = when (result) {
                is PaymentSheetResult.Completed -> UiState.Success
                is PaymentSheetResult.Canceled  -> UiState.Idle
                                                is   PaymentSheetResult.Failed
UiState.Error(result.error.message ?: "Stripe error")
            }}}
    sealed interface UiState {
        object Idle : UiState
        object Loading : UiState
             data class AwaitPayment(val secret: String, val sheet:
PaymentSheet) : UiState
        object Success : UiState
        data class Error(val msg: String) : UiState}
```

Усю клієнтську обробку платежу «Donate» з опорою на Stripe SDK код організовує в чітко розділені шари, які разом перетворюють натискання кнопки «Donate» на гарантований платіж і відповідний запис у бекенді. На мережевому рівні клас DonateRequest у форматі Moshi точно відповідає серверній моделі DonateComponentModel, серіалізуючи поле amount як BigDecimal, щоб не втрачати копійок і уникати округлення, притаманного типу double.

Інтерфейс FundraisingStripeApi описує маршрут POST {id}/donate; після виклику бекенд створює Stripe PaymentIntent і повертає лише clientSecret, не розкриваючи чутливих ключів – така відповідь десеріалізується у PaymentIntentResponse. Репозиторій DonateStripeRepository, працюючи у

Dispatchers.IO, асинхронно викликає цей метод і віддає clientSecret у ViewModel, фактично роблячи сполучну ланку між транспортом і презентаційною логікою.

DonateStripeViewModel веде власний StateFlow, послідовно фіксуючи стани Idle → Loading → AwaitPayment (зі збереженим secret і екземпляром PaymentSheet) → Success або Error; завдяки цьому Jetpack Compose реактивно перемальовує інтерфейс без заколисувань колбеків. Найпомітніше в очах користувача відбувається у Composable DonateButton: при стані AwaitPayment виконується побічний ефект PaymentSheet.presentWithPaymentIntent, що піднімає нативну форму Stripe і проводить користувача через Apple Pay, Google Pay чи введення картки. Після завершення Stripe надсилає результат у ViewModel через колбек onPaymentResult; якщо статус Completed, екран дякує за підтримку, інакше система повертається до початкового стану або показує помилку.

Уся критична логіка–створення PaymentIntent, утримання суми, фактичний капчер – залишається на бекенді; клієнт ніколи не зберігає секретних ключів і не підміняє суму, оскільки остання жорстко передана у DonateRequest і закріплена PaymentIntent-ом. Такий поділ – Retrofit + Repository + ViewModel + Compose – гарантує прозорість даних, тестованість (репозиторій легко мокувати), дотримання MVVM та відкриває шлях до подальшого розширення: той самий модуль можна перенацілити, наприклад, на PayPal або LiqPay, замінивши лише API-інтерфейс і репозиторій, не торкаючись ViewModel чи UI.

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У цьому розділі розглянуто декілька сценаріїв тест-кейсів для візуального розуміння процесу розробки та прогресу виконання кожного етапу відповідно технічному завданню. Далі у таблиці 5.1 наведено тестові сценарії.

Таблиця 5.1 – Тест-кейс №1 (таблиця виконана самостійно)

Інформація про тест-кейс			
Ідентифікатор тесту:		Тест-кейс №1	
Власник тесту:		Ляшенко Євген Андрійович	
Дата створення:		23.05.2025	
Мета тесту:		APK у режимі *debug* встановлений на Android 13; бекенд 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»	Лінійний 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 підрядок; інші зникають	Пройдено
3	Натиснути «Sort» → вибрати категорію medical	Відображаються лише ініціативи з категорією «medical»; бейдж категорії підсвічено	Пройдено
4	Свайп-refresh при ввімкненому фільтрі	Дані перезавантажено; активний фільтр не скидається	Пройдено
5	Тап по картці ініціативи	Відкриваються деталі: фото, опис, прогрес-бар, кнопка «Donate»	Пройдено
6	Згорнути екран деталей ← та повернутися до списку	Позиція прокрутки збережена; картка виділена тонким обводом	Пройдено
7	Активувати офлайн-режим (Wi-Fi OFF) та відкрити список	Локальний кеш показує останні 10 ініціатив, snackbar «Offline mode»	Пройдено
8	Спроба свайп-refresh в офлайн-режимі	Toast «No internet»; список не змінюється	Пройдено

Кінець таблиці 5.1.

№	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»	Список доступний для гостей, але кнопка «Create» зникла	Пройдено
Донат через Stripe та оновлення прогресу			
№	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	Повернувшись на екран, спостерігати прогрес	Прогрес-бар і сума збільшились без перезапуску екрана (подія WebSocket)	Пройдено
5	В історії донатів (Profile → My Donations) з’явився запис із сьогоднішньою датою	Запис містить id ініціативи, суму, статус «Succeeded»	Пройдено
6	Повторити крок 1, але натиснути «X» у PaymentSheet	Платіж скасовано; стан кампанії не змінено; у історії запис відсутній	Пройдено
7	Спроба донату 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; ViewModel переходить у UiState.Error	Пройдено
9	Отримати webhook payment_intent.succeeded із затримкою >30 с	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

ВИСНОВОК

У межах передатестаційної практики було спроєктовано та детально описано концепцію мобільного застосунку з підсистемою управління ініціативами та зборами, який має забезпечити повний цикл організації благодійних кампаній – від створення ініціативи та верифікації організатора до запуску збору коштів і відстеження прогресу в реальному часі. Аналіз предметної галузі засвідчив, що мобільні рішення із мінімальною формою введення та швидкою реакцією значно підвищують конверсію донатів завдяки зниженню «тертя» у процесі пожертви. Запроєктований застосунок враховує ці принципи через інтуїтивний інтерфейс, багатомодульну архітектуру та передбачену інтеграцію з платіжними шлюзами (Stripe, LiqPay, QR-платежі).

Архітектурне проєктування виконано на основі UML-діаграм і передбачає мікросервісну структуру, яка гарантує модульність, можливість горизонтального масштабування в середовищі Kubernetes і швидке розгортання.

UI/UX-дизайн застосунку розроблено відповідно до сучасних мінімалістичних стандартів: лаконічні форми, чіткі заклики до дії, адаптивність до різних екранів і миттєвий фідбек. Особлива увага приділяється доступності та зручності на всіх етапах використання – від створення ініціатив до редагування профілю й налаштування сповіщень.

Запропоновані програмні рішення передбачають реалізацію підходу Zero Trust: багаторівневе шифрування даних, псевдонімізацію персональної інформації (PII) та суворе розмежування доступу відповідно до ролей користувачів. Це дозволяє поєднати високий рівень безпеки з якісним користувацьким досвідом.

Отже, спроєктована система відповідає сучасним функціональним, продуктивним і регуляторним стандартам (GDPR, PSD2, AML/KYC) та формує надійну основу для подальшої реалізації – впровадження аналітичних і AI-інструментів, розширення міжнародної підтримки й інтеграції нових комунікаційних каналів у сфері мобільного фандрейзингу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. GoFundMe Statistics [Електронний ресурс]. – Режим доступу: https://www.sci-tech-today.com/stats/gofundme-statistics/ (дата звернення: 11.04.2025).

2. The owner of Monobank is fundraising money for nuclear weapons: 24 million hryvnias were raised in one day [Електронний ресурс]. – Режим доступу: https://mind.ua/en/news/20285819-the-owner-of-monobank-is-fundraising-money-for-nuclear-weapons-24-million-hryvnias-were-raised-in (дата звернення: 13.04.2025).

3. About us – Добро.ua [Електронний ресурс]. – Режим доступу: https://dobro.ua/en/about_us/ (дата звернення: 14.04.2025).

4. Fact Check: GoFundMe donations will support Hurricane Helene victims [Електронний ресурс]. – Режим доступу: https://www.reuters.com/fact-check/trump-gofundme-donations-will-support-hurricane-helene-victims-2024-10-04/ (дата звернення: 25.04.2025).

5. Patreon Stats: Revenue & Users [Електронний ресурс]. – Режим доступу: https://influencermarketinghub.com/patreon-stats-revenue-users/ (дата звернення: 27.04.2025).

6. The scoop on Strong Customer Authentication (SCA) [Електронний ресурс]. – Режим доступу: https://www.pingidentity.com/en/resources/blog/post/the-scoop-on-strong-customer-authentication-sca.html?utm_source=chatgpt.com (дата звернення: 03.05.2025).

7. Recurring donors: benefits and trends [Електронний ресурс]. – Режим доступу: https://neonone.com/resources/blog/recurring-donors/?utm_source=chatgpt.com (дата звернення: 06.05.2025).

8. Anti-money laundering: Council and Parliament strike deal on stricter rules [Електронний ресурс]. – Режим доступу: https://www.consilium.europa.eu/en/press/press-releases/2024/01/18/anti-money-launde

ring-council-and-parliament-strike-deal-on-stricter-rules/?utm_source=chatgpt.com (дата звернення: 08.05.2025).

9. Documents – сторінка 90, Парламентська дослідницька служба [Електронний ресурс]. – Режим доступу: https://research.rada.gov.ua/en/documents/page/90?utm_source=chatgpt.com (дата звернення: 09.05.2025).

10. GoFundMe 2024 Year in Help [Електронний ресурс]. – Режим доступу: https://www.gofundme.com/c/gofundme-2024-year-in-help?utm_source=chatgpt.com (дата звернення: 11.05.2025).

11. Aigner, S., Elizarov, R., Isakova, S., Jemerov, D. *Kotlin in Action. Second Edition*. — New York : Manning Publications, 2024. — 560 p. — ISBN 978-1-61729-960-5.

12. TechTarget. Model-View-ViewModel (MVVM) [Електронний ресурс] // TechTarget. — Режим доступу: https://www.techtarget.com/whatis/definition/Model-View-ViewModel, вільний. (дата звернення: 15.05.2025).

13. Google LLC. Jetpack Compose [Електронний ресурс] // Android Developers. — Режим доступу: https://developer.android.com/compose, вільний (дата звернення: 16.05.2025).

14. GitHub посилання на виконану роботу — Режим доступу: https://github.com/NureLiashenkoYevhen/2025_B_PI_PZPI-21-1_Liashenko_Y_A