Специфікація ПЗ

1 ВСТУП

* 1. Огляд продукту

Програмна система для моніторингу та управління криптовалютними активами на iOS-пристроях є мобільним для перегляду криптовалют, створеним з використанням SwiftUI, Combine, CoreData та архітектури MVVM. Система призначена для демонстрації реального курсу криптовалют, перегляду деталей монет, збереження портфоліо користувача в локальній базі даних та динамічного оновлення UI.

### 1.2 Мета

Метою проєкту є розробка сучасного iOS-застосунку для відображення криптовалютної інформації та керування персональним портфелем активів. Застосунок має забезпечувати автономну роботу, бути модульним, масштабованим і простим у супроводі. Для досягнення цієї мети використовується архітектура MVVM (Model–View–ViewModel), яка сприяє чіткому розділенню відповідальностей між шаром даних, логіки та інтерфейсу, полегшує підтримку та повторне використання компонентів.

1.3 Межі

Програмне забезпечення призначене виключно для мобільних пристроїв на базі операційної системи iOS з версією не нижче 15. Його функціональність орієнтована на автономну роботу користувача без необхідності авторизації або створення облікового запису. Система не передбачає реалізації мультикористувацького доступу, а також не включає підтримки серверної частини чи інтеграції з хмарними сервісами. Усі дані зберігаються локально на пристрої користувача, що обмежує можливості синхронізації між різними пристроями та доступу з інших платформ.

### **1.5 Означення та абревіатури**

**iOS** — мобільна операційна система компанії Apple, яка використовується на iPhone та iPad.

**MVVM (Model–View–ViewModel)** — архітектурний шаблон, що розділяє структуру застосунку на три логічні компоненти: модель даних (Model), представлення (View) і модель представлення (ViewModel).

**API (Application Programming Interface)** — інтерфейс прикладного програмування; набір засобів для взаємодії з віддаленими або внутрішніми сервісами.

**UI (User Interface)** — користувацький інтерфейс; спосіб взаємодії користувача із застосунком через візуальні елементи.

**UX (User Experience)** — досвід користувача; сприйняття та реакція користувача під час взаємодії із застосунком.

**Swift** — мова програмування, розроблена компанією Apple для створення застосунків під iOS, macOS та інші платформи.

**SwiftUI** — декларативний фреймворк для створення інтерфейсів користувача на Swift.

**Combine** — фреймворк від Apple для реактивного програмування, що дозволяє працювати з асинхронними подіями.

**CoreData** — фреймворк Apple для управління локальною базою даних, який забезпечує збереження, відновлення та керування об’єктами моделі.

2 ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

Кваліфікаційна робота має високий потенціал як автономний мобільний застосунок для моніторингу криптовалютного ринку та ведення персонального портфеля з локальним збереженням даних. Завдяки відсутності потреби в постійному інтернет-з'єднанні, він забезпечує підвищений рівень безпеки, конфіденційності та незалежності від сторонніх сервісів. Такий підхід особливо актуальний для користувачів, які прагнуть повного контролю над своїми фінансовими даними без ризику передачі їх на віддалені сервери.

У майбутньому робота може бути розширений за рахунок нових аналітичних можливостей, підтримки мультипортфельного обліку, поглибленої візуалізації динаміки активів та додаткових локальних інструментів прогнозування. Архітектура системи дозволяє масштабування та адаптацію під потреби різних категорій користувачів. Це створює основу для комерційного розвитку — як незалежного фінансового асистента, орієнтованого на приватність, автономність та стабільну роботу навіть в умовах обмеженого доступу до мережі.

### **2.2 Функції продукту**

Програмний продукт забезпечує користувачеві зручний доступ до основної інформації про ринок криптовалют, включаючи перегляд поточних курсів, ринкової капіталізації, динаміки змін цін та інших ключових показників. Застосунок дозволяє формувати персональний інвестиційний портфель, додавати активи, фіксувати операції купівлі та продажу, переглядати статистику, розраховувати прибутковість та відстежувати загальну вартість активів у зручному графічному форматі. Інформація оновлюється в реальному часі при наявності з’єднання з мережею, а у разі його відсутності застосунок працює в автономному режимі, використовуючи збережені локально дані.

Крім основної функціональності, застосунок реалізує модуль налаштувань, де користувач може обирати тему оформлення інтерфейсу, мову. Передбачено ефективну інтеграцію з API зовнішніх сервісів для завантаження ринкових даних, а також використання локальної бази даних (CoreData) для збереження інформації про портфель та історію дій.

### **2.3 Характеристики користувачів**

Цільовою аудиторією застосунку є приватні особи, які інвестують у криптовалюту або регулярно відстежують динаміку цифрових активів. Основними користувачами є користувачі iOS-пристроїв з базовими або середніми технічними навичками, які шукають інструмент для контролю за станом свого криптопортфеля без необхідності підключення до бірж чи сторонніх хмарних сервісів. Для таких користувачів важливими є простота інтерфейсу, безпека збереження даних, автономність, а також швидкий доступ до актуальної ринкової інформації.

Також до потенційних користувачів належать фінансово свідомі особи, які цінують приватність і віддають перевагу локальним рішенням, а не онлайн-сервісам з централізованим зберіганням даних. Застосунок може бути корисним як для довгострокових інвесторів, що ведуть облік активів у власному портфелі, так і для активних спостерігачів ринку, які регулярно переглядають тренди та аналізують зміну вартості криптовалют.

### **2.4 Загальні обмеження**

Застосунок розроблено виключно для платформи iOS, що обмежує його використання лише власниками пристроїв Apple, таких як iPhone та iPad. Через автономну архітектуру без авторизації користувачів і синхронізації з хмарними сервісами, усі дані зберігаються локально на пристрої, що виключає можливість відновлення портфеля у разі втрати або пошкодження пристрою. Також застосунок залежить від зовнішніх API для отримання ринкових даних, тому при відсутності з'єднання з інтернетом оновлення інформації недоступне — використовуються останні збережені локальні дані.

### **2.5 Припущення й залежності**

Під час розробки кваліфікаційної роботи прийнято кілька ключових припущень, які визначають як технічні особливості, так і середовище експлуатації програмного продукту. Передбачається, що кінцевий користувач має базове розуміння принципів роботи з мобільними застосунками та криптовалютами, вміє додавати активи, редагувати дані портфеля та працювати з візуалізацією фінансової інформації.

Функціонування застосунку залежить від ряду зовнішніх і внутрішніх умов:

* стабільної роботи зовнішнього API для отримання ринкових даних (у разі відсутності підключення система використовує кешовану інформацію);
* підтримки актуальної версії iOS (мінімальна версія — iOS 15 або новіша);
* наявності достатнього обсягу вільної пам’яті на пристрої для локального збереження даних;
* функціональності системних компонентів SwiftUI, Combine, CoreData;
* відсутності жорстких обмежень на використання інтернету з боку системних політик пристрою.

Припускається, що користувач зберігає відповідальність за резервне копіювання даних власними засобами, оскільки застосунок не передбачає хмарної синхронізації або централізованого зберігання. Усі критичні обчислення, а також формування інтерфейсу, виконуються локально, що виключає залежність від серверної частини, але накладає обмеження на складність обробки даних при масштабуванні функціоналу в майбутньому.

3 КОНКРЕТНІ ВИМОГИ

### **3.1 Вимоги до зовнішніх інтерфейсів**

**3.1.1 Інтерфейс користувач**

Інтерфейс реалізований на базі SwiftUI з використанням вкладеної навігації, реактивного оновлення контенту та підтримкою адаптивного дизайну. Застосунок має кілька основних екранів: ринок криптовалют, портфель користувача, налаштування та сторінку деталей активу. Інтерфейс підтримує темну й світлу теми, зручний для користувачів з базовим досвідом, та доступний на різних розмірах екранів iPhone/iPad.

**3.1.2 Апаратний інтерфейс**

Використовується лише стандартна сенсорна взаємодія з пристроєм (тапи, свайпи, клавіатура). Застосунок не потребує підключення до додаткового обладнання чи сенсорів (наприклад, камери, Bluetooth або біометрії). Достатньо наявності стандартного сучасного мобільного пристрою Apple.

**3.1.3 Програмний інтерфейс**

Застосунок звертається до зовнішнього REST API для отримання ринкової інформації про криптовалюти. Для локального збереження даних використовується CoreData. Реалізовано гнучкий обмін даними між Model та View через Combine-потоки. Інтеграція з іншими застосунками або системами не передбачається.

**3.1.4 Обмеження памʼяті**

Програма не створює значного навантаження на внутрішнє сховище пристрою. Зберігається лише необхідна інформація: дані портфеля, кеш ринку та налаштування. Обсяг використаної пам’яті залежить від обсягу історії операцій, але зазвичай не перевищує кількох десятків мегабайт.

**3.1.5 Операції**

Користувач має змогу переглядати динаміку крипторинку, додавати активи до портфеля, редагувати або видаляти їх, переглядати статистику, керувати системними налаштуваннями. Кожна операція ініціює автоматичне оновлення інтерфейсу без перезапуску застосунку.

**3.1.6 Функції продукту**

Застосунок забезпечує базовий функціонал для криптоінвестора: моніторинг курсу валют, створення й облік власного портфеля, побудова графіків, підрахунок змін вартості активів, ведення історії операцій, налаштування мови та теми, вибір основної валюти. Підтримується офлайн-режим.

**3.1.7 Припущення й залежності**

Передбачається, що користувач має актуальний пристрій Apple з iOS 15 або новішою версією. Для повноцінної роботи застосунку необхідне періодичне підключення до інтернету для оновлення ринкових даних. Програма залежить від стабільної роботи API, проте зберігає кеш для автономного доступу.

3.2 Властивості програмного продукту

Програмний продукт відзначається високою надійністю завдяки чіткій архітектурі MVVM, яка розділяє логіку, дані та інтерфейс, що сприяє ізоляції помилок і їх швидкому виявленню. Застосунок забезпечує безперебійну роботу в умовах нестабільного інтернет-з’єднання, зберігаючи останні кешовані дані для офлайн-доступу. Інтерфейс адаптивний і інтуїтивно зрозумілий, підтримує світлу та темну теми, а також локалізацію, що підвищує доступність для широкого кола користувачів. Безпека реалізована через локальне збереження даних без передачі особистої інформації на зовнішні сервери, що знижує ризик витоку даних.

3.3 Атрибути програмного продукту

### 3.3.1 Надійність

Програмний продукт забезпечує стабільну та безперебійну роботу завдяки чіткій архітектурі MVVM, яка розмежовує логіку, дані та інтерфейс, що мінімізує кількість помилок і спрощує їх виявлення. Застосунок коректно обробляє мережеві збої, підтримуючи кешування даних для офлайн-доступу, що гарантує безперервність користування навіть при відсутності інтернет-з’єднання.

### 3.3.2 Доступність

Інтерфейс продукту адаптований під різні розміри екранів пристроїв Apple, таких як iPhone та iPad, підтримує світлу та темну теми, а також локалізацію. Це забезпечує зручність користування для широкого кола користувачів із різними потребами, включно з тими, хто має обмеження за зоровими або моторними навичками.

### 3.3.3 Безпека

Усі персональні та фінансові дані користувача зберігаються виключно локально на пристрої, що виключає ризик несанкціонованого доступу чи витоку інформації. Продукт відповідає сучасним стандартам і рекомендаціям Apple щодо захисту даних і забезпечує конфіденційність інформації користувача.

### 3.3.4 Супроводжуваність

Застосунок розроблений із використанням сучасних технологій Apple — SwiftUI, Combine та CoreData, що забезпечує модульність, чистоту коду та простоту внесення змін. Це полегшує розширення функціоналу, підтримку та тестування продукту на всіх етапах його життєвого циклу.

### 3.3.5 Переносимість

Програмний продукт сумісний із актуальними версіями iOS і оптимізований для роботи на різних пристроях Apple, включно з iPhone та iPad. Адаптивність інтерфейсу і використання стандартних фреймворків забезпечують коректне функціонування без необхідності додаткової конфігурації.

### 3.3.6 Продуктивність

Застосунок оптимізовано для швидкої обробки великих обсягів даних та плавного оновлення інтерфейсу. Використання реактивного програмування і локального кешування знижує навантаження на систему, забезпечуючи швидку відповідь і стабільну роботу навіть за значної кількості транзакцій та активів.