Министерство образования Калининградской области

государственное бюджетное учреждение Калининградской области
профессиональная образовательная организация
«Колледж информационных технологий и строительства»
(ГБУ КО ПОО «КИТиС»)

ТЕХНИЧЕСКОЕ ЗАДАНИЕ
на разработку приложения "Крипто-Трекер".

Выполнил: Студент 3 курса
группы ИСп22-1
Логвинов Артём Стасович

_____
(подпись)

Калининград, 2025

Содержание

1 Общие положения

1.1 Полное наименование

Мобильное приложение для отслеживания курса криптовалют «CryptoManiya».

1.2 Определения, обозначения и сокращения

- Мобильное Приложение – разрабатываемое программное обеспечение, предназначенное для работы на смартфонах конкретной мобильной платформы (в нашем случае, Android).
- Пользователь – студент (16-20 лет), который будет использовать разрабатываемое программное обеспечение.
- API – CoinGecko API для получения данных о курсах.
- MVVM – Model-View-ViewModel — архитектурный паттерн.
- DI – Dependency Injection (Dagger-Android или Hilt).
- Room – библиотека для работы с SQLite базой данных.
- LiveData, ViewModel – компоненты Android Architecture Components.
- Retrofit – HTTP-клиент для запросов.
- Glide – библиотека для загрузки изображений (иконок криптовалют).
- MPAndroidChart – библиотека для отображения графиков.
- Coroutine — Kotlin Coroutines для асинхронности.

2 Назначение и цели создания

2.1 Назначение

Предоставить пользователям удобный инструмент для отслеживания курса криптовалют, анализа динамики и хранения избранных активов.

2.2 Цели

- Показ актуальных котировок криптовалют в реальном времени.
- Визуализация исторических цен через графики.
- Лёгкая навигация, чистая архитектура (MVVM + DI).
- Поддержка любимых (избранных) валют.

3 Требования к приложению

3.1 Структура разделов

1. Главный экран (список валют):
   - Отображение названия, текущей цены, 24-часового изменения.
2. Экран детали валюты:
   - График исторической цены (7/30/90 дней).
   - Дополнительные метрики (объём, изменение, описание и т. д.).
   - Иконка валюты (Glide).
3. Избранное
   - Отмеченные/отслеживаемые валюты, отдельный список.
4. Настройки
   - Тема (светлая/тёмная).
   - Язык интерфейса.

3.2 Нефункциональные требования

3.2.1 Требования к интерфейсу

- Соответствие Material Design, адаптивность под разные экраны.
- Поддержка светлой и тёмной тем (темная тема по запросу).
- Локализация (RU/EN).

3.3 Функциональные требования

1. Интеграция с CoinGecko API:
   - Использовать Retrofit + Coroutines.
   - Обрабатывать потенциальные ошибки: сетевые, парсинг, лимиты API.
2. Получение и обновление курсов
   - Авто-обновление.

3. Графики исторических данных
   - MPAndroidChart: отображение динамики за разные периоды с zoom/pan.
4. Избранное
   - Добавление/удаление валют из избранного, сохранение в БД.
5. Загрузка изображений
   - Glide для иконок валют.
6. Навигация
   - Android Navigation Component.
7. Асинхронность
   - Kotlin Coroutines + ViewModelScope.
8. UI–логика
   - LiveData: ViewModels наблюдают за данными.
9. Обработка ошибок и состояния
   - ProgressBar при загрузке.
   - Сообщения об ошибках.

Опциональные улучшения

- Уведомления при достижении ценовых уровней.
- Конвертер валют (фиат ↔ крипто).
- Сравнение валют на одном графике.

4 Структура архитектуры приложения

Приложение построено на архитектурном паттерне MVVM с использованием Dagger-Hilt для внедрения зависимостей.

4.1 Основные слои

- Model – получение и кеширование данных (репозиторий, API, Room)
- ViewModel – логика взаимодействия с View, подписка на LiveData
- View – фрагменты/активности, отображающие UI и слушающие события

5 Бизнес-сценарии (End-to-End)

Сценарий 1: Просмотр списка криптовалют

Актор: Пользователь

Предусловие: есть соединение с интернетом

Шаги:

1. Пользователь открывает приложение
2. Приложение отображает список криптовалют с ценами и изменениями
3. Пользователь может нажать на валюту, чтобы перейти в детали

Результат: Пользователь видит список актуальных данных по валютам

Сценарий 2: Работа с избранным

Шаги:

1. Пользователь нажимает на "сердечко" у монеты
2. Монета добавляется в Room с флагом isFavorite = true
3. На экране "Избранное" отображаются только такие валюты

Сценарий 3: Просмотр динамики

Шаги:

1. Пользователь выбирает валюту
2. Открывается экран с графиком
3. Пользователь выбирает диапазон: 7 / 30 / 90 дней
4. График обновляется, данные приходят через API

6 Структура экранов (UI-навигация)

Экран 1: Главный

- Поиск
- Список: название, символ, цена, изменение %, иконка кнопка "В избранное"

Экран 2: Детали

- Название
- График
- Текущая цена

Экран 3: Избранное

- Отображает только те монеты, у которых нажата кнопка “добавить в избранное”.

Экран 4: Настройки

- Тема
- Язык (при наличии)

8 Требования к безопасности и качеству кода.

- Все сетевые запросы должны использовать HTTPS
- Обработка ошибок (API, сеть, нет данных)
- Ограничение повторных вызовов API (дебаунс/троттлинг)
- Чистая архитектура;
- Покрытие тестами ViewModel и Repository;
- Использование Jetpack Components.