Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(Университет ИТМО)

Факультет **Прикладной информатики**

Направление подготовки **45.03.04 Интеллектуальные системы в гуманитарной сфере**
Образовательная программа **Языковые модели и искусственный интеллект**

К У Р С О В О Й П Р О Е К Т

Тема: «iOS приложение CryptoTracker»

Обучающийся: Попов Артемий Албертович, K3161

Санкт-Петербург 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Рынок криптовалют активно растёт, привлекая всё больше инвесторов. Для успешной работы на этом рынке необходим удобный и эффективный инструмент, позволяющий отслеживать цены, изменения рыночной капитализации и другие ключевые показатели. Мобильное приложение CryptoTracker для iOS призвано решить эту задачу, предоставляя пользователям удобный способ мониторинга криптовалютного рынка. Готовое приложение станет полезным инструментом как для опытных трейдеров, так и для начинающих инвесторов, которые хотят эффективно управлять своими активами.

Цель проекта — разработать приложение CryptoTracker для iOS, которое обеспечит удобный и информативный мониторинг криптовалютного рынка.

Основные задачи проекта:

- построение базовой архитектуры приложения [1].
- реализация функционала получения данных через REST API [2].
- создание экрана списка криптовалют и экрана подробной информации о монете.
- интеграция сетевого слоя с пользовательским интерфейсом.

1 Описание проекта

Проект CryptoTracker — это мобильное приложение для iOS [3], предназначенное для отслеживания актуальной информации о криптовалютах. Оно предоставляет пользователям список криптовалют с их текущими ценами, изменениями в процентах за последние 24 часа, а также позволяет детально изучить данные по каждой монете. Интерфейс приложения выполнен в современном стиле с учетом принципов UX/UI, что делает его интуитивно понятным. Для получения актуальных данных о рынке криптовалют в реальном времени используется REST API [2].

Основная цель проекта — упрощение доступа к информации о криптовалютах, что делает приложение полезным инструментом как для начинающих инвесторов, так и для опытных трейдеров.

2 Ход выполнения работ над проектом

Разработка проекта началась с этапа планирования, где задачи были распределены между участниками команды. В качестве основной архитектурной модели был выбран подход MVVM (Model-View-ViewModel). Этот шаблон позволяет четко разделить бизнес-логику и логику отображения данных, что упрощает поддержку и дальнейшее расширение функциональности приложения.

Для разработки пользовательского интерфейса использовались фреймворк UIKit [3] и библиотека SnapKit [4], которая помогла настроить зависимости между элементами интерфейса. Взаимодействие с API и обработка данных были реализованы с помощью URLSession, а для управления асинхронными операциями применялись замыкания. Работа в команде была организована через систему задач с четкими сроками выполнения. Регулярные встречи и проведение код-ревью позволяли своевременно выявлять ошибки и находить оптимальные решения.

3 Поставленная задача

Передо мной были поставлены такие задачи, как создание кастомных UI элементов, то есть отображение ошибки в случае чего, и добавление кнопки с возможностью сортировки изменения цены за сутки или за час.

4 Отображение ошибки

Я создал класс CustomButton, наследующийся от UIButton, класс TextFieldView, наследующийся от UIView. Также добавлен CustomTextField и CustomLabel в качестве подвидов этого класса.

Настроен их расположение и внешний вид.

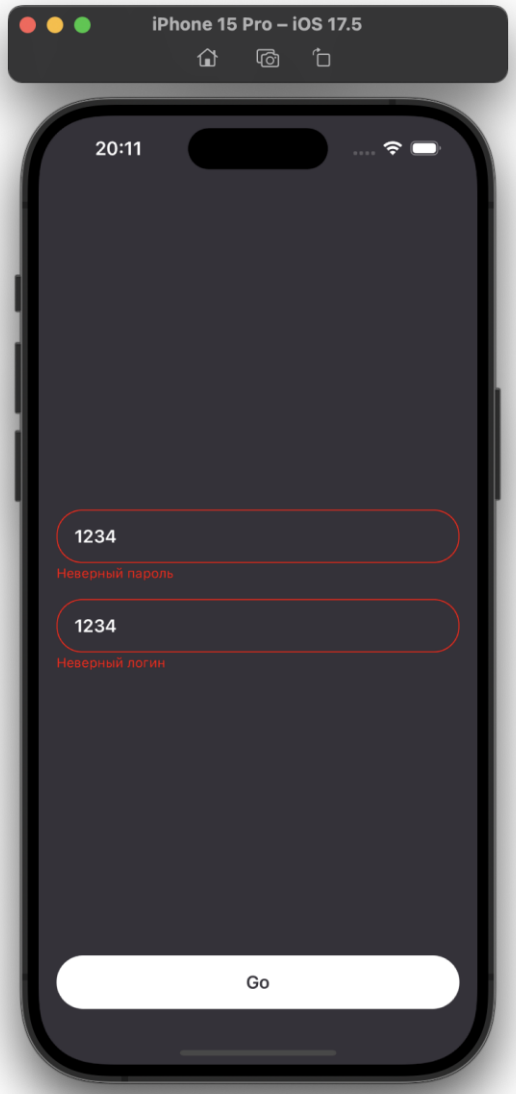

Рисунок 1 – Отображение ошибки

5 Кнопка сортировки

В navigation bar добавить кнопку с возможностью сортировки изменения цены за сутки или за час (список или по возрастанию или по убыванию).

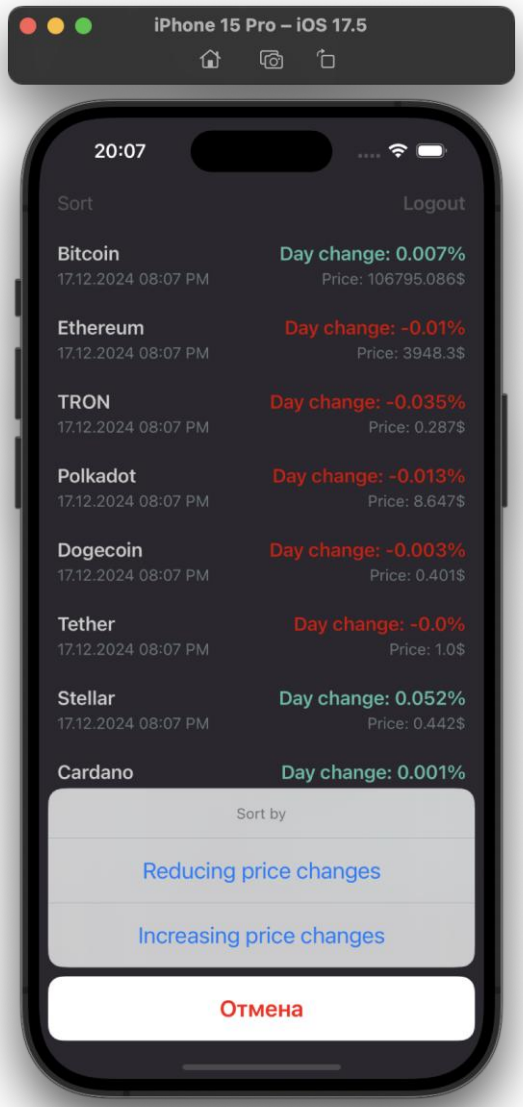

Рисунок 2 – Кнопка сортировки

6 Анализ работы и выводы

Работа над проектом дала мне возможность познакомиться с библиотеками Swift, такими как Combine. В процессе разработки возникли сложности, связанные с изучением нового языка и программирования и применения его на практике.

Несмотря на возникающие трудности, я смог систематически выполнять поставленные задачи, соблюдая установленные сроки. Проект стал для меня ценным опытом, который позволил не только изучить новые технологии, но и улучшить навыки командной работы.

7 Взаимодействие с командой и руководителем проекта

Работа в команде была выстроена очень слаженно. У каждого члена команды была своя четко поставленная задача, за которую он был ответственный. Обсуждение в команде во многом помогла в решении проблем.

Руководитель в течении всего проекта очень помогал. Он предлагал различные варианты решения и поддерживал на протяжении всего времени.

8 Оценка руководителя команды

Руководитель команды продемонстрировал себя как грамотный и уверенный лидер, который эффективно организовал работу всех участников проекта. Он чётко распределял задачи, оперативно предоставлял обратную связь и контролировал, чтобы проект двигался в соответствии с намеченными целями.

Его подход помог создать продуктивную атмосферу в команде, где каждый осознавал свои обязанности и зону ответственности. Руководитель умело сочетал поддержку индивидуальной инициативы с соблюдением общей стратегии, что способствовало оперативному решению возникающих вопросов.

Его советы по реализации функций приложения и анализу обратной

связи от пользователей значительно повысили качество конечного продукта.

Таким образом, его руководство было эффективным, что способствовало успешному завершению проекта.

ЗАКЛЮЧЕНИЕ

В заключение можно отметить, что поставленная цель была успешно реализована. Все ключевые задачи выполнены в установленные сроки. Я лично внес существенный вклад в проект, разработав экран авторизации. Работа над этим проектом стала для меня ценным опытом в разработке iOS-приложений, а также в командной работе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. MVVM архитектруа приложения - https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel
2. API приложения: Messary IO - https://messari.io/
3. Официальная документация Apple - https://developer.apple.com/documentation/
4. Официальная документация UIKit - https://developer.apple.com/documentation/uikit/
5. Официальная документация Snapkit - https://snapkit.github.io/SnapKit/docs/
6. Фреймворк Combine и реактивное программирование - https://developer.apple.com/documentation/combine

ПРИЛОЖЕНИЕ

Этапы задач:

- Анализ предметной области,
- Проектирование,
- Разработка,
- Ручное тестирование.

Задачи:

- Название: Разработать экран авторизации
  Ответственный: Баташов Богдан Александрович
  Описание: Первый вью контроллер для авторизации (AuthViewController), просто 2 текс Филда (UITextField) и кнопка (UIButton).

- Название: Разработать экран списка монет
  Ответственный: Востров Илья Анатольевич
  Описание: Список всех монет сделанный через tableview.
  На ячейке укажите название монеты, ее стоимость, и ее изменение за сутки или час. (Данные брать из API)
  Пока список монет грузится, отображается спинер (UIActivityIndicator).
  После тапа на ячейку, осуществляется переход на 3-тий вью контроллер.

- Название: Разработка экрана подробной информации о монете
  Ответственный: Титор Матвей Андреевич
  Описание: 3 - ий вью контроллер. Просто детальная информация о монете.
  (Можно в ряд добавить лейблы с любой дополнительной информацией о монете)
  При нажатии на ячейку доставать название монеты и передавать на этот экран. Дальше использовать название для запроса

- Название: Внедрить координатор для навигации между контроллерами
  Ответственный: Мелихов Андрей Юрьевич
  Описание: Нужно добавить Координатор
  - может пушить и попать вьюконтроллер

- Название: Разработать класс StorageService для сохранения состояния авторизации пользователя

Ответственный: Мелихов Андрей Юрьевич

Описание: Класс который инкапсулирует в себе логику (методы) для сохранения состояния isAuth: Bool переменной в UserDefaults

- Название: Разработать класс NetworkLayer, инкапсулирующий логику взаимодействия с сетью

  Ответственный: Титор Матвей Андреевич

  Описание: - Построить сетевой слой. (Network Layer in Swift)

  - Разобраться как использовать Codable = Encodable & Decodable

  - GET/POST/PATCH - REST API

  - URLSeesion, URL, URLRequest

  - JSONDecoder

- Название: Разработать сервис икапсулирующий логику работы с CoreData (Опционально)

  Ответственный: Востров Илья Анатольевич

  Описание: Создание класса CoreDataManager. Если юзер не был авторизован

- Название: Добавить кнопки фильтрации в NavigationBar

  Ответственный: Попов Артемий Альбертович

  Описание: В навигейшен бар слева добавьте кнопку с возможностью сортировки изменения цены за сутки или за час

- Название: Настроить проект

  Ответственный: Востров Илья Анатольевич

  Описание: Интегрировать .gitignore файл, а также установить зависимости в проект (SnapKit), через cocoapods

- Название: Создать кастомные UI элементы

  Ответственный: Попов Артемий Альбертович

  Описание: - Наследник UITextField, который умеет менять бордер, если произошла ошибка и отображать внизу error message (по сути надо сделать наследника CustomTextField: UITextField, настроить его делегаты, а потом его вместе с CustomLabel: UILabel поместить в класс TextFieldView: UIView и при верстке использовать только его)