

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет
по курсовой работе
по дисциплине «Разработка мобильных приложений»
название приложения: Финтек

Автор: Хайрат Марат

Факультет: ФПИиКТ

Группа: Р33212

Преподаватель: Ключев А.О.



Санкт-Петербург 2023

Оглавление

<i>Введение</i>	<i>3</i>
<i>Техническое задание</i>	<i>4</i>
<i>Архитектура системы</i>	<i>6</i>
<i>Тестирование</i>	<i>9</i>
<i>Заключение</i>	<i>11</i>
<i>Список литературы</i>	<i>12</i>
<i>Приложения</i>	<i>12</i>

Введение

Актуальность работы

Инвестиционные приложения приобрели значительную популярность в последние годы, поскольку предоставляют частным лицам доступные и удобные средства для участия в мире финансов и управления капиталом. Появление технологий изменило способ инвестирования людей, упростив как новичкам, так и опытным инвесторам ориентироваться на сложных финансовых рынках. И в то же время, при наличии большого количества инвестиционных приложений, нет приложений, позволяющих получить знания о криптовалюте. Каждая криптография сопровождается каким-то техническим документом, который слишком длинный, чтобы иметь мало знаний.

Цели и задачи работы

Цель работы — разработка собственного приложения для начинающих, которые хотят узнать о криптовалютах, которые их создали, какие технологии они использовали, и другую информацию. Кроме того, они могут видеть графики криптовалюты и могут конвертировать криптовалюту в такие валюты, как рубли, доллары и т.д. Приложение также будет интересно и полезно людям в возрасте 16+

Техническое задание

Функциональные требования

Аутентификация пользователя:

Регистрация пользователей: Пользователи должны иметь возможность регистрировать учетные записи, используя свои адреса электронной почты, и устанавливать защищенные пароли.

Логин пользователя: Зарегистрированные пользователи могут войти в систему, используя свой адрес электронной почты и пароль используя аутентификацию Firebase.

Сброс пароля: Пользователи могут запросить ссылку для сброса пароля на свой зарегистрированный адрес электронной почты в случае, если они забудут свой пароль. Встроенная функциональность Firebase обрабатывает процесс сброса пароля.

Основные характеристики

- Удобный интерфейс со списком различных криптовалют.
- Содержит краткие описания или значки, чтобы сделать монеты визуально привлекательными для молодых пользователей.
- Реализация функций, позволяющих пользователям выбирать конкретную монету для получения подробной информации.
- Разработка подробного представления для каждой выбранной монеты с отображением такой информации, как:
 - 1) Создатель(ы) crypto
 - 2) Технологии, использованные при разработке crypto
 - 3) Ключевые особенности и функционал.
 - 4) Любые заметные вехи или достижения.
- Крипто конвертер - предназначен для беспрепятственного обмена валют. Мгновенная конвертация между различными цифровыми валютами с точностью до реального времени. Просто введите сумму, выберите желаемую криптовалюту, и наш конвертер предоставит вам эквивалентную стоимость в выбранной вами валюте на основе последних обменных курсов. Благодаря поддержке различных криптовалют вы можете эффективно управлять своими цифровыми активами, гарантируя, что всегда будете в курсе своих инвестиций.
- График криптовалюты - Эта функция обеспечивает подробное визуальное представление динамики цен на криптовалюту с течением времени.

Нефункциональные требования

Безопасность:

Аутентификация и авторизация: Аутентификация Firebase используется для безопасного управления регистрацией и входом в систему пользователя, обеспечивая проверку личности пользователя перед предоставлением доступа.

Удобство использования:

Интуитивно понятный дизайн пользовательского интерфейса: Приложение имеет интуитивно понятный пользовательский интерфейс

Адаптивный дизайн: Приложение адаптируется к различным размерам экранов и устройствам, обеспечивая единообразный пользовательский опыт

Тестирование:

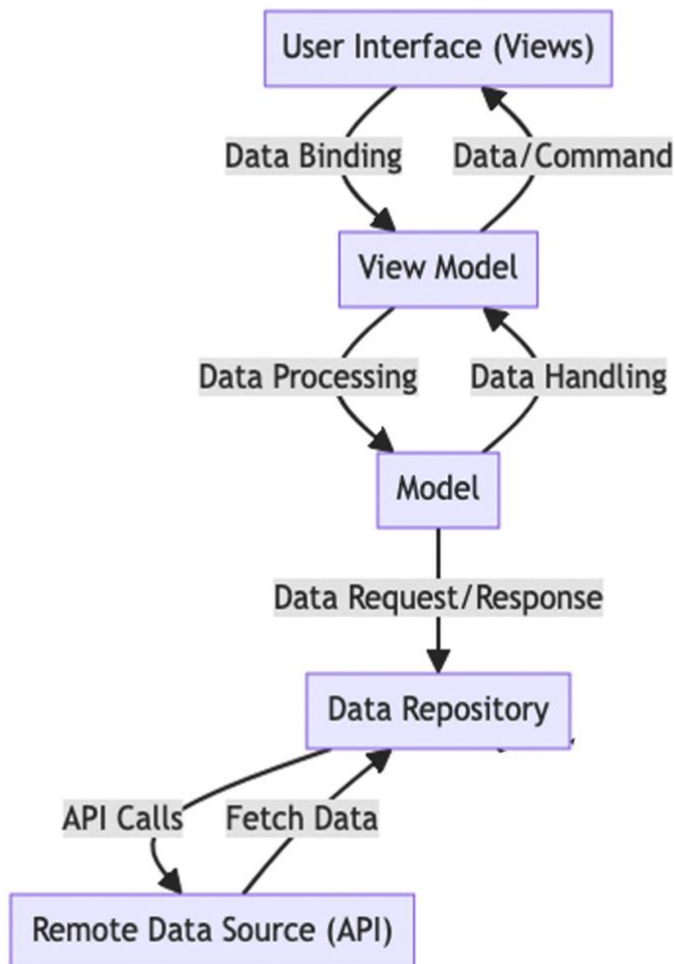
Нагрузочное тестирование JunitTest: Нагрузочное тестирование с использованием юнит тест в котлине

Архитектура системы

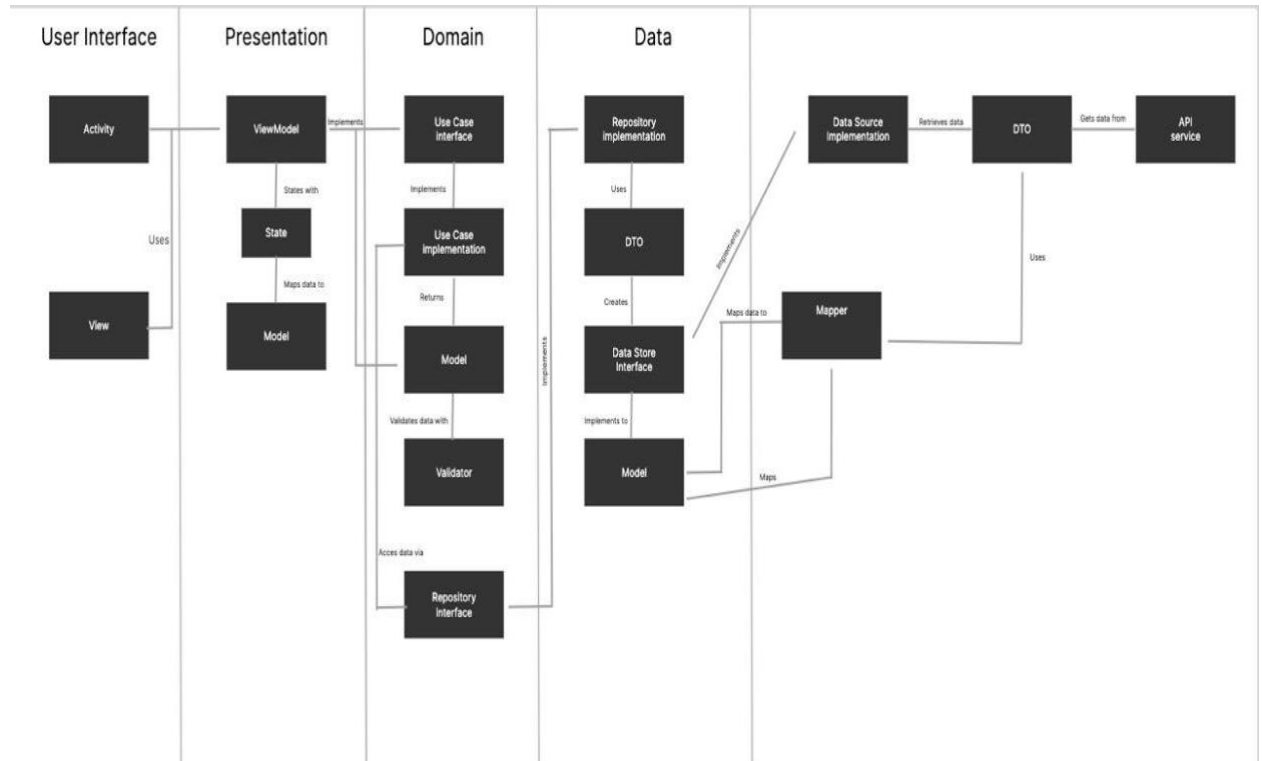
Стек технологий

Kotlin, Gradle, Jetpack Compose, Retrofit, FirebaseAuth

Архитектура



MVVM



Описание MVVM:

Model: Это уровень данных, содержащий бизнес-логику и структуры данных приложения. Он отвечает за извлечение, хранение и обработку данных.

View: Представление - это уровень пользовательского интерфейса. Здесь пользователь взаимодействует с приложением. Он отображает данные из **ViewModel** и отправляет пользовательские команды **ViewModel** для выполнения действий.

ViewModel: **ViewModel** действует как посредник между представлением и моделью. Он обрабатывает логику представления и предоставляет данные из модели таким образом, чтобы представление могло легко их использовать.

В MVVM представление привязывается к свойствам и командам **ViewModel**, которая, в свою очередь, взаимодействует с моделью для извлечения или обновления данных. Затем **ViewModel** уведомляет представление об изменениях данных с помощью привязки данных, чтобы представление могло обновляться соответствующим образом.

User interface: Это будет включать **Activity** и **View**, напрямую взаимодействующие с **ViewModel**.

Presentation: Это будет включать **ViewModel**, который привязывается к **View** и взаимодействует с уровнем домена для вариантов использования.

Domain: Этот уровень содержит варианты использования и модели, которые инкапсулируют бизнес-логику.

Data: Это будет включать реализацию репозитория, который взаимодействует с уровнем домена и источниками данных, такими как локальная база данных или удаленный API.

Бизнес-процессы

Аутентификация пользователя: управляет входом пользователя в систему и безопасностью, гарантируя, что только авторизованные пользователи получают доступ к своим брокерским счетам.

Извлечение данных: обеспечивает извлечение и непрерывное обновление данных о счете, портфеле и транзакциях с серверов брокера.

Анализ данных: выполняет углубленный анализ портфеля, транзакций и рыночных данных, чтобы предоставить пользователям ценную статистику и углубленную аналитику.

Управление: позволяет пользователям эффективно размещать свои инвестиционные транзакции и контролировать их, усиливая контроль над своим финансовым портфелем.

Тестирование

✓ CryptoTestApi.kt

✓ CryptoApiTest

- loadCryptoList(): List<CryptoModel>
- `1000 API calls and filter crypto`: Unit
- `filter cryptos above random price threshold`: Unit
- `count cryptos below random price`: Unit
- `find maximum price in crypto list`: Unit
- `average price of cryptos`: Unit
- `check specific currency`: Unit

```
class CryptoApiTest {
    private fun loadCryptoList(): List<CryptoModel>{
        val jsonFile = File("/Users/makok/Downloads/Fintech
3/app/src/test/java/com/example/fintech/data.json")
        val jsonContent = jsonFile.readText()
        val mapper = jacksonObjectMapper()
        return mapper.readValue(jsonContent)
    }
    @Test
    fun `1000 API calls and filter crypto`() {
        val cryptoList = loadCryptoList()
        repeat(1000){
            val expensiveCoins = cryptoList.filter{
                val price = it.price.toDoubleOrNull() ?: 0.0
                price > 100.0
            }
            assert(expensiveCoins.isNotEmpty())
        }
        val lastExpensiveCoins = cryptoList.filter{
            val price = it.price.toDoubleOrNull() ?: 0.0
            price > 100
        }
        lastExpensiveCoins.forEach{ coin ->
            println("Coin: ${coin.currency}, Price: ${coin.price}")
        }
    }
    @Test
    fun `filter cryptos above random price threshold`() {
        val cryptoList = loadCryptoList()
        val randomThreshold = Random.nextDouble(0.1, 0.5)
        val filteredList = cryptoList.filter { it.price.toDouble() > randomThreshold }
```

```

        filteredList.forEach { crypto ->
            assertTrue("Failed at price ${crypto.price}", crypto.price.toDouble() >
randomThreshold)
        }
    }
}

```

@Test

```

fun `count cryptos below random price`() {
    val cryptoList = loadCryptoList()
    val randomPrice = Random.nextDouble(0.05, 0.3)
    val expectedCount = cryptoList.count { it.price.toDouble() < randomPrice }
    val actualCount = cryptoList.filter { it.price.toDouble() < randomPrice }.size
    assertEquals(expectedCount, actualCount)
}

```

@Test

```

fun `find maximum price in crypto list`() {
    val cryptoList = loadCryptoList()
    val maxPrice = cryptoList.maxByOrNull { it.price.toDouble() }?.price?.toDouble() ?: 0.0
    val expectedMaxPrice = cryptoList.map { it.price.toDouble() }.maxOrNull() ?: 0.0
    assertEquals(expectedMaxPrice, maxPrice, 0.0)
}

```

@Test

```

fun `average price of cryptos`() {
    val cryptoList = loadCryptoList()
    val averagePrice = cryptoList.map { it.price.toDouble() }.average()
    val expectedAverage = cryptoList.sumOf { it.price.toDouble() } / cryptoList.size
    assertEquals(expectedAverage, averagePrice, 0.0)
}

```

@Test

```

fun `check specific currency`() {
    val cryptoList = loadCryptoList()
    val randomCrypto = cryptoList.random().currency
    val containsCurrency = cryptoList.any { it.currency == randomCrypto }
    assertTrue(containsCurrency)
}
}

```

Заключение

В данной курсовой работе было проведено исследование и разработка приложения "Финтек", предназначенного для помощи начинающим инвесторам в изучении криптовалют. Основная цель проекта была достигнута: создан удобный, функциональный инструмент, который предоставляет пользователям актуальную информацию о криптовалютах, их создателях и используемых технологиях, а также позволяет осуществлять конвертацию криптовалют и анализировать графики их изменения стоимости.

В ходе работы над проектом были получены ценные знания и практические навыки в области разработки мобильных приложений

Список литературы

Официальная документация по Jetpack Compose

Android Developers. (2023). Jetpack Compose. <https://developer.android.com/jetpack/compose>. 2023

Статья о трансформации кода из Java в Kotlin

Baeldung. (2023). "Convert Java to Kotlin". <https://www.baeldung.com/kotlin/convert-java-to-kotlin>. 2023

Документация FirebaseAuth

Firebase. (2023). FirebaseAuth <https://firebase.google.com/docs/auth> 2023

Документация Retrofit

Square. (2023). Retrofit <https://square.github.io/retrofit> 2023

Официальная документация по Kotlin.

JetBrains. (2023). Kotlin Documentation <https://kotlinlang.org/docs/home.html> 2023

Приложения

6:27

Hey there,
Create an Account

First Name

Last Name

Email

Password

☐ By continuing you accept our [Privacy Policy](#) and [Term of Use](#)

Register

Already have an account? [Login](#)

6:27

Login
Welcome Back

Email

Password

[Forgot your password](#)

Login

Don't have an account yet? [Register](#)

6:28

Hey, there!

Crypto encyclopedia

Crypto Converter

Crypto Chart's

6:29

Home

1. Bitcoin (BTC)	active
2. Ethereum (ETH)	active
3. Tether (USDT)	active
4. Binance Coin (BNB)	active
5. XRP (XRP)	active
6. Solana (SOL)	active
7. USDC (USDC)	active
8. Lido Staked Ether (STETH)	active
9. Cardano (ADA)	active
10. Dogecoin (DOGE)	active
11. TRON (TRX)	active

6:29

Home

1. Bitcoin (BTC)

Bitcoin is a cryptocurrency and worldwide payment system. It is the first decentralized digital currency, as the system works without a central bank or single administrator.

Tags

Segwit Cryptocurrency Proof Of Work Payments SHA256 Mining Lightning Network

Team members

Satoshi Nakamoto
Founder

Wladimir J. van der Laan

6:30

Crypto Converter

BTC 100.04 -0%
Bitcoin

DOGE 16.8 +110%
Dogecoin

ETH 0.05 +11%
Ethereum

USDT 1.0 +0%
USD

Send 0.686 EUR

Get ETH



11/10