

Белорусский Государственный Университет
Факультет радиофизики и компьютерных технологий

Отчет по лабораторной работе №4

«Создание прототипа интерфейса. Проектирование API и разработка архитектуры ИС»

Подготовили:

Студенты 4 курс 5ПИ,

Равгейша Алексей

Бабарико Виолетта

Петров Егор

Преподаватель:

Ломако А.А.

Минск, 2025

1. Проектирование прототипа пользовательского интерфейса

На основе утвержденного технического задания и Use Case Diagram выбран ключевой сценарий "Генерация рекомендаций". Данный сценарий является центральным для функциональности системы "Интеллектуальная система по рекомендациям инвестиционных портфелей", поскольку он интегрирует данные о предпочтениях пользователя, анализе портфеля и внешних рыночных данных для формирования персонализированных инвестиционных предложений. Он соответствует основной цели проекта – помочь трейдерам в формировании оптимальных портфелей.

Прототип интерфейса разработан в формате wireframes для мобильного приложения (iOS/Android). Wireframes представлены реализованными в Figma и отрендерены для iPhone 16. Фокус на пяти последовательных экранах: подключение кошелька, небольшой опрос для построения рекомендаций и сам просмотр рекомендаций. Дизайн минималистичен.

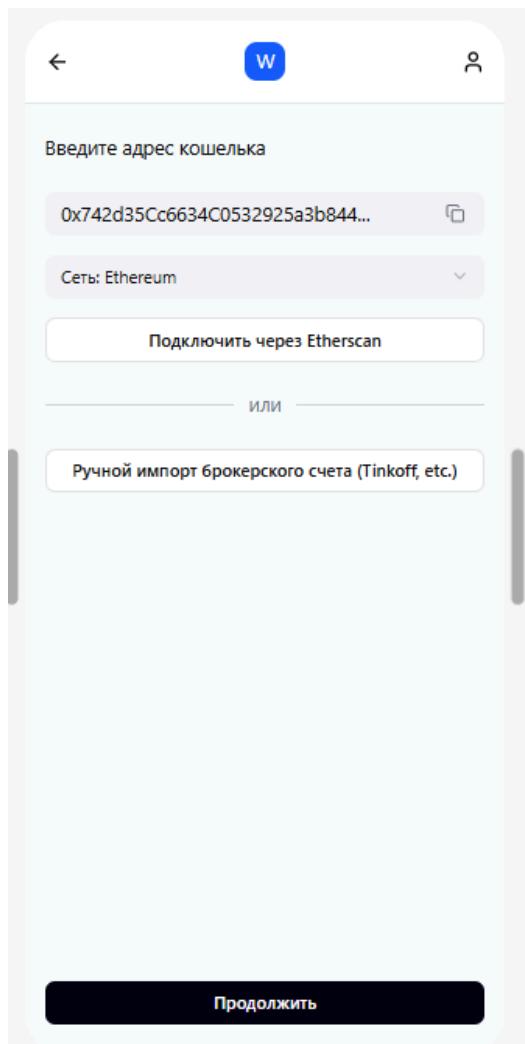


Рисунок 1. Фрейм подвязки кошелька

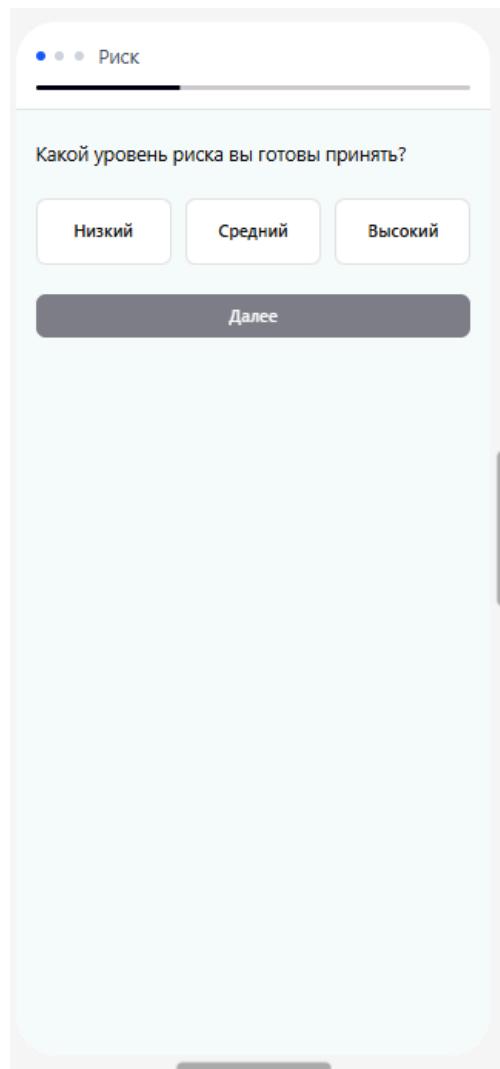


Рисунок 2. Первый фрейм опроса.

• • Сектора

Выберите интересующие сектора (до 3):

Выбрано: 0/3

DeFi AI & Tech

Green Energy Stablecoins

Meme Blue Chips

Далее

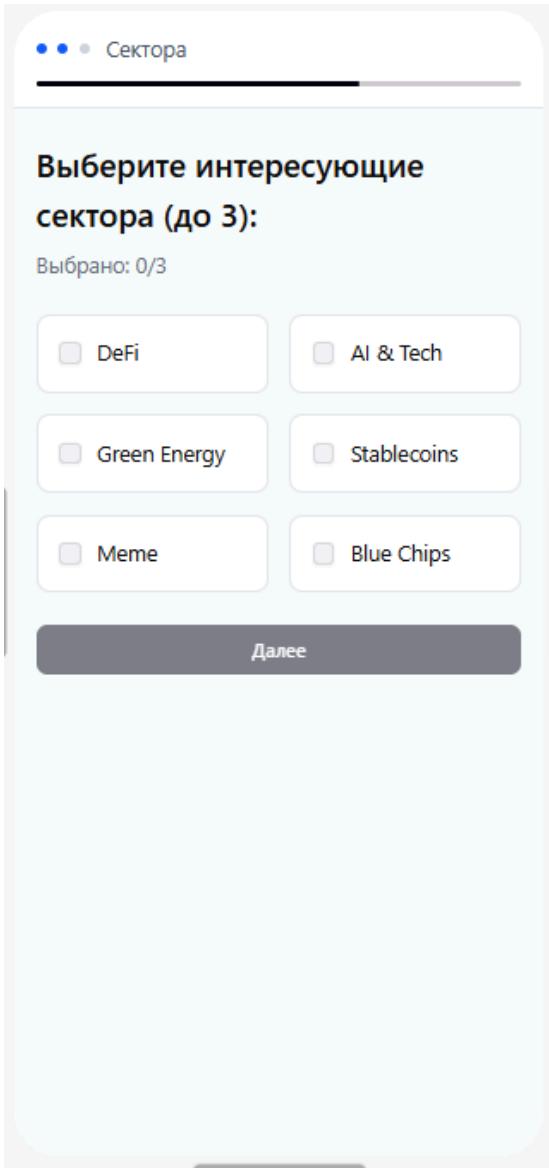


Рисунок 3. Второй фрейм опроса.

• • Сектора

Выберите интересующие сектора (до 3):

Выбрано: 3/3

DeFi AI & Tech

Green Energy Stablecoins

Meme Blue Chips

Далее

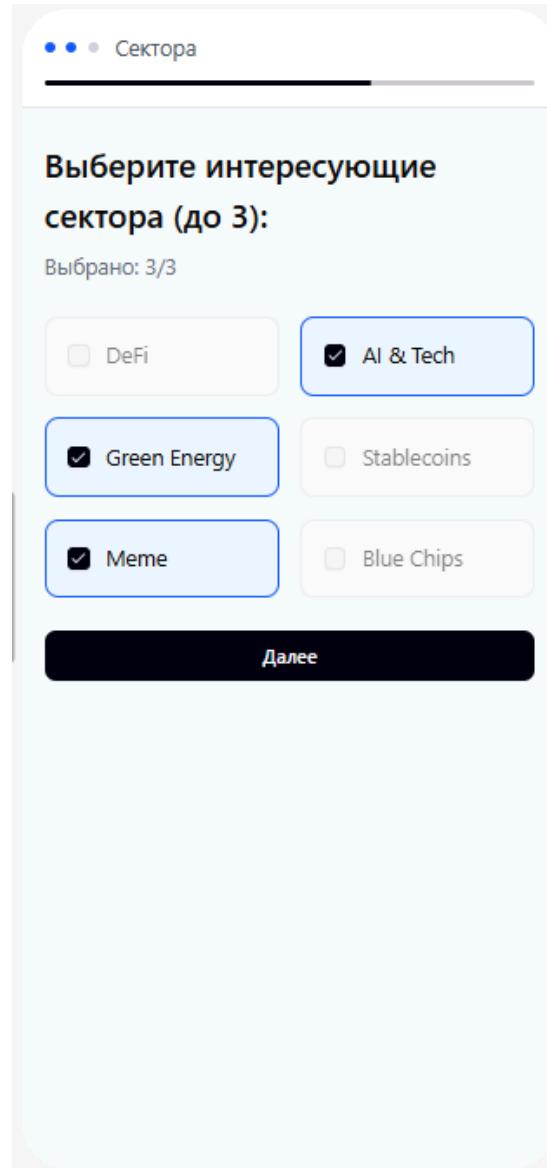


Рисунок 4. Второй фрейм опроса с выбором.

• • • Горизонт

На какой срок планируете инвестировать?

< 6 мес 6-12 мес > 12 мес

Сгенерировать рекомендацию

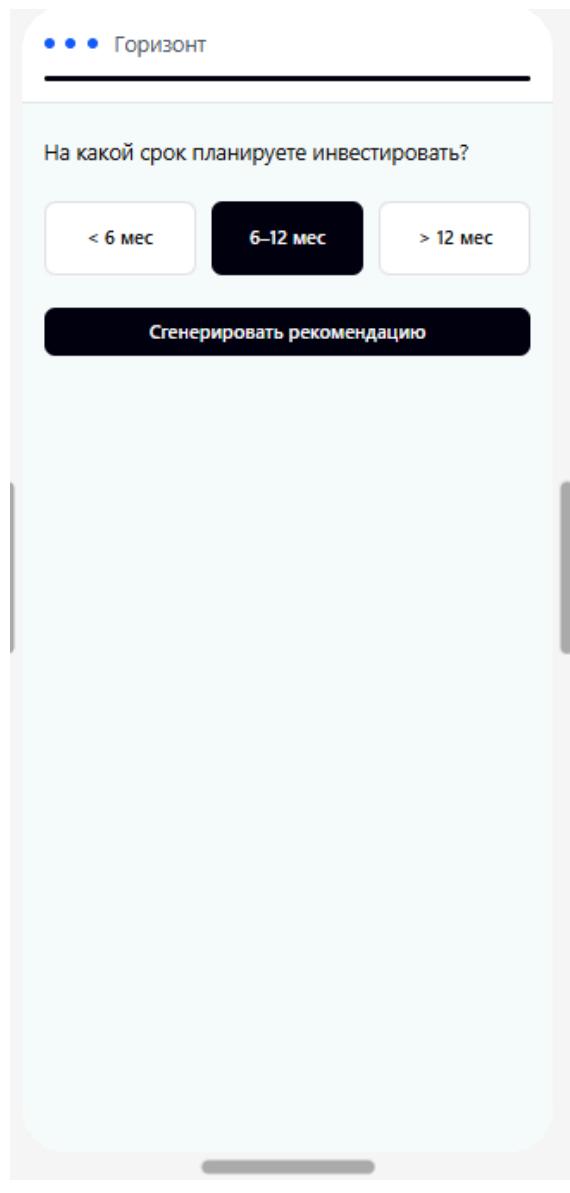


Рисунок 5. Третий фрейм опроса.

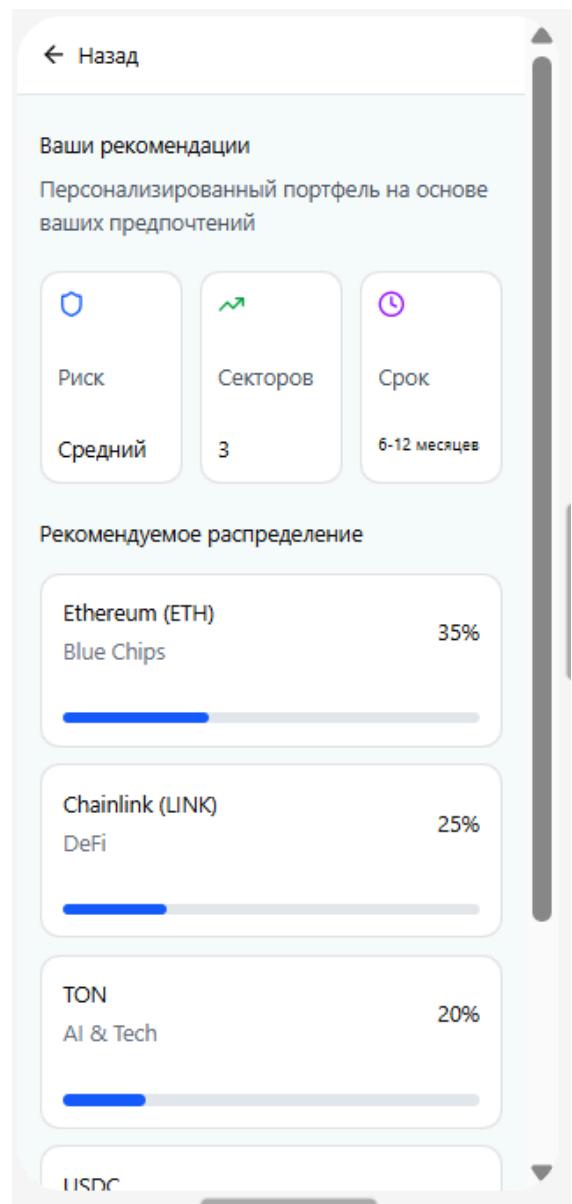
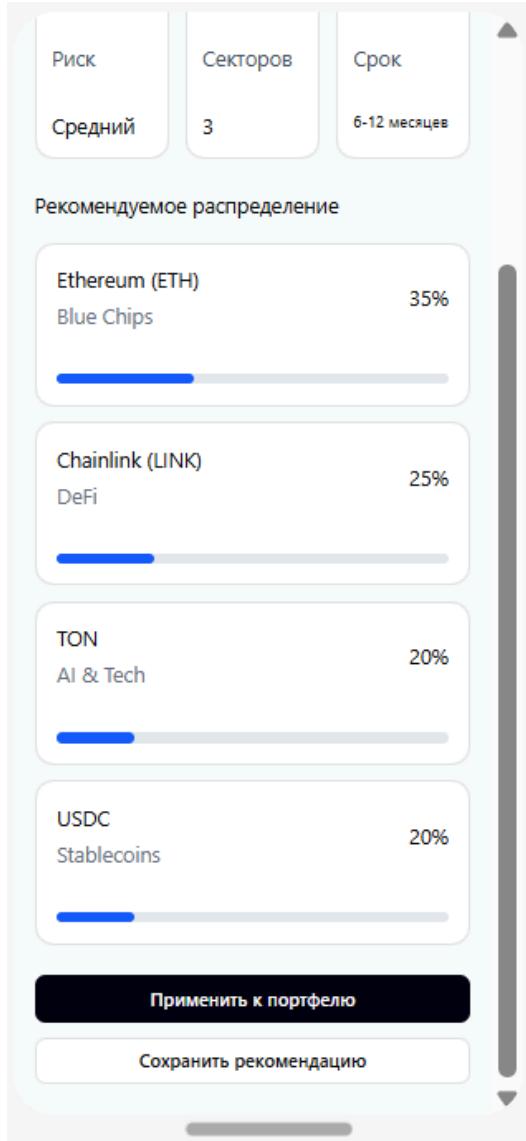


Рисунок 6. Итоговые рекомендации.



действий расположены в нижней части экрана – в зоне естественного доступа для большого пальца. Кнопка "Назад" с иконкой стрелки и текстом соответствует общепринятым мобильным паттернам (iOS, Android), что снижает порог входа для новых пользователей. Использование понятных иконок (щит – риск, график – сектора, часы – срок) усиливает визуальную логику и не требует дополнительного объяснения.

Предотвращение ошибок реализовано через информативность и контроль действий: уровень риска и срок инвестиций отображаются текстом, а не только цветом (например, красный цвет может означать высокий риск, а зеленый – низкий), что исключает ошибки восприятия; каждая рекомендация сопровождается названием и точным процентом распределения; кнопка "Применить к портфелю" не выполняется мгновенно – предполагается переход к подтверждению, что защищает от импульсивных транзакций. Пользователь всегда может вернуться назад или сохранить рекомендацию без обязательного действия.

Закон Хика (время выбора растет с увеличением числа альтернатив) учтен путем ограничения количества вариантов: вместо десятков активов пользователю предлагается всего 4 тщательно подобранных позиции, что сокращает время анализа и принятия решения. На экране присутствует не более 3 интерактивных зон (навигация, список активов, кнопки), что соответствует оптимальному диапазону для мобильных интерфейсов и предотвращает "паралич выбора", особенно у начинающих инвесторов.

Закон Фиттса (скорость достижения цели зависит от размера и расстояния до цели) реализован через увеличенные тач-зоны:

- Кнопки "Применить к портфелю" и "Сохранить рекомендацию" занимают всю ширину экрана и имеют высоту не менее 56 px – это соответствует минимальным требованиям для удобного нажатия большим пальцем;
- Основная информация (риск, срок, распределение) размещена в верхней половине экрана, в зоне естественного обзора;
- Прогресс-бары и карточки имеют достаточную высоту и отступы, что упрощает скроллинг и взаимодействие одной рукой.

На основе прототипа сформулированы следующие обязательные требования к финальному интерфейсу с учетом доступности (WCAG 2.1 AA):

1.1. Визуальная доступность

- Контраст текста к фону – не менее 4.5:1 для обычного текста, 3:1 для крупных заголовков.
- Цвет не должен быть единственным носителем информации: уровень

риска и сектора сопровождаются текстом.

- Поддержка системного высокого контраста и темной темы.

1.2. Управление с клавиатуры и screen reader

- Все элементы интерактивны с клавиатуры (Tab, Enter).
- Видимый фокус (обводка) при навигации.
- Семантические метки для screen reader:
 - Кнопка "Назад" — «Вернуться к предыдущему экрану»;
 - Прогресс-бар — «Ethereum: 35% от портфеля»;
 - Кнопка "Применить" — «Применить рекомендованный портфель».

1.3. Адаптивность и масштабирование

- Поддержка увеличения текста до 200% без горизонтальной прокрутки.
- Все тач-зоны – не менее 44×44 px.
- Адаптация под экраны от 320 px (iPhone SE) до 414 px (Android).

1.4. Подтверждение критических действий

- При нажатии "Применить к портфелю" – обязательное модальное окно с текстом: «*Вы уверены? Это приведёт к покупке активов через партнёра. Риск: [уровень]*» Кнопки: "Отмена" и "Подтвердить".

1.5. Локализация и читаемость

- Все тексты вынесены в систему локализации (i18n).
- Шрифт: системный sans-serif (SF Pro / Roboto), размер основного текста – 16–17 px.
- Междустрочный интервал – 1.5.

1.6. Тестирование доступности

- Проверка через Lighthouse (Accessibility ≥ 95), axe DevTools.
- Тестирование с VoiceOver (iOS) и TalkBack (Android).
- Проверка на дальтонизм (симуляторы Color Oracle).

2. Проектирование API.

Для организации взаимодействия компонентов информационной системы «Интеллектуальная система по рекомендациям инвестиционных портфелей» выбран REST-подход на основе протокола HTTP/HTTPS. Данный выбор обусловлен следующими факторами:

- Простота и стандартизация — REST использует стандартные HTTP-методы (GET, POST, PUT, PATCH, DELETE), что упрощает

интеграцию с веб- и мобильными клиентами.

- Масштабируемость — stateless-природа REST позволяет горизонтально масштабировать серверную часть без сохранения сессий.
- Кэшируемость — GET-запросы могут кэшироваться на уровне CDN или браузера, снижая нагрузку на сервер.
- Совместимость с внешними сервисами — интеграция с API CoinMarketCap, Investing.com, Binance, Tinkoff и др. осуществляется через REST, что делает внутренний API унифицированным.
- Поддержка JSON — основной формат обмена данными, совместимый с JavaScript, Python, мобильными SDK.

Выбрано взаимодействие между Frontend (Web/Mobile App) и Backend Server — ключевой поток данных в системе. Спецификация API охватывает основные сценарии: аутентификация, получение профиля, анализ портфеля, генерация рекомендаций, уведомления и выполнение транзакций.

2.1. Прототип спецификации REST API

Базовый URL: <https://api.investadvisor.pro/v1>

Аутентификация: Используется JWT Bearer Token, передаваемый в заголовке “Authorization: Bearer” Токен выдается при регистрации/входе и имеет TTL = 24 часа.

2.1.1. Регистрация и вход

- POST /auth/register

Описание: Регистрация нового пользователя

Тело запроса (JSON):

```
{  
    "email": "string",  
    "password": "string",  
    "name": "string"  
}
```

Ответ (200 OK):

```
{  
    "user_id": "integer",  
    "access_token": "string",  
}
```

```
    "expires_in": 86400  
}
```

- POST /auth/login

Описание: Аутентификация пользователя

Тело запроса: аналогично /register

Ответ: аналогично /register

2.1.2. Профиль пользователя

- GET /profile

Описание: Получение профиля пользователя

Ответ (200 OK):

```
{  
    "user_id": 123,  
    "email": "user@example.com",  
    "name": "Иван Иванов",  
    "risk_tolerance": "medium",  
    "sectors_interest": ["DeFi", "AI", "Green Energy"],  
    "investment_horizon": 12,  
    "registration_date": "2026-03-15T10:00:00Z"  
}
```

- PATCH /profile/survey

Описание: Обновление результатов адаптивного опроса

Тело запроса:

```
{  
    "risk_tolerance": "low|medium|high",  
    "sectors_interest": ["string"],  
    "investment_horizon": "integer (months)"  
}
```

Ответ (200 OK):

```
{ "status": "updated" }
```

2.1.3. Подключение кошельков и счетов

- POST /wallets

Описание: Добавление криптокошелька

Тело запроса:

```
{
  "address": "0x742d35Cc6634C0532925a3b844Bc454e4438f44e",
  "blockchain": "ETH"
}
```

Ответ (201 Created):

```
{
  "wallet_id": 456,
  "address": "0x742d35Cc6634C0532925a3b844Bc454e4438f44e",
  "connected_at": "2026-03-16T12:00:00Z"
}
```

- POST /accounts/import

Описание: Ручной импорт брокерского счета (JSON-выгрузка)

Тело запроса:

```
{
  "broker": "Tinkoff",
  "holdings": [
    {
      "symbol": "AAPL",
      "quantity": 10.5,
      "avg_price": 175.30
    }
  ]
}
```

2.1.4. Анализ портфеля

- GET /portfolio

Описание: Получение текущего портфеля

Ответ (200 OK):

```
{  
    "portfolio_id": 789,  
    "total_volume_usd": 15420.50,  
    "assets": [  
        {  
            "symbol": "BTC",  
            "type": "crypto",  
            "quantity": 0.52,  
            "current_price": 68200.00,  
            "value_usd": 35464.00,  
            "source": "etherscan"  
        },  
        {  
            "symbol": "AAPL",  
            "type": "stock",  
            "quantity": 10,  
            "current_price": 196.00,  
            "value_usd": 1960.00,  
            "source": "manual"  
        }  
    ],  
    "last_update": "2026-03-17T08:00:00Z"  
}
```

2.1.5. Генерация рекомендаций

- POST /recommendations/generate

Описание: Запуск генерации рекомендаций

Тело запроса (опционально):

```
{  
    "portfolio_id": 789,  
    "force_refresh": true  
}
```

Ответ (202 Accepted):

```
{  
    "rec_id": 101,  
    "status": "processing",  
    "estimated_time": 15  
}
```

- GET /recommendations/{rec_id}

Описание: Получение готовой рекомендации

Ответ (200 OK):

```
{  
    "rec_id": 101,  
    "generated_at": "2026-03-17T08:05:00Z",  
    "predicted_income_12m": 18.7,  
    "recommended_assets": [  
        {  
            "symbol": "SOL",  
            "type": "crypto",  
            "allocation_pct": 25,  
            "reason": "high growth in DeFi, similar to top 5%  
portfolios"  
        },  
        {  
            "symbol": "NVDA",  
            "type": "stock",  
            "allocation_pct": 30,  
            "reason": "AI sector momentum, LSTM forecast +12.4%"  
    ]  
}
```

```
    in 6m"
}
],
"simulation_url": "/simulations/101"
}
```

2.1.6. Уведомления

- GET /notifications

Описание: Получение списка уведомлений

Параметры: ?limit=20&offset=0

Ответ:

```
{
  "notifications": [
    {
      "notif_id": 301,
      "type": "price_alert",
      "message": "BTC вырос на 7% за час — рассмотрите ребалансировку",
      "sent_at": "2026-03-17T09:30:00Z",
      "read": false
    }
  ],
  "total": 5
}
```

2.1.7. Выполнение транзакций

- POST /transactions/execute

Описание: Инициация транзакции через партнера

Тело запроса:

```
{
  "rec_id": 101,
```

```

    "asset_symbol": "SOL",
    "amount_usd": 3000,
    "partner": "Binance"
}

```

Ответ (201 Created):

```

{
    "trans_id": 501,
    "status": "redirect",
    "redirect_url":
"https://binance.com/checkout?ref=investadvisor_501",
    "commission_earned": 45.00
}

```

2.2. HTTP-методы и коды ошибок

Метод	Использование
GET	Чтение данных (профиль, портфель, рекомендации)
POST	Создание (регистрация, кошелек, транзакция)
PATCH	Частичное обновление (опрос)
PUT	Не используется
DELETE	Не используется (логика архивации)

Коды ошибок (JSON-ответы)

```

// 400 Bad Request
{
    "error": "invalid_request",
    "message": "Поле 'email' обязательно"
}
// 401 Unauthorized

```

```
{  
    "error": "unauthorized",  
    "message": "Неверный или истекший токен"  
}  
  
// 403 Forbidden  
{  
    "error": "forbidden",  
    "message": "Доступ запрещен: недостаточно средств в  
портфеле"  
}  
  
// 404 Not Found  
{  
    "error": "not_found",  
    "message": "Рекомендация с rec_id=999 не найдена"  
}  
  
// 409 Conflict  
{  
    "error": "conflict",  
    "message": "Кошелек уже подключен"  
}  
  
// 422 Unprocessable Entity  
{  
    "error": "validation_failed",  
    "details": [  
        { "field": "investment_horizon", "issue": "должно быть  
> 0" }  
    ]  
}
```

```
}

// 429 Too Many Requests
{
    "error": "rate_limit",
    "message": "Превышен лимит: 100 запросов/минуту",
    "retry_after": 60
}

// 500 Internal Server Error
{
    "error": "server_error",
    "message": "Внутренняя ошибка сервера"
}
```

2.3. Заголовки и безопасность

Все запросы и ответы используют заголовок Content-Type: application/json, что гарантирует корректную интерпретацию данных на стороне клиента и сервера. Для защиты от злоупотреблений введено ограничение скорости запросов (rate limiting): не более 100 запросов в минуту с одного IP-адреса и не более 500 запросов в сутки на одного аутентифицированного пользователя. Политика CORS настроена таким образом, что запросы разрешены только с доверенных доменов — app.investadvisor.pro и localhost:* в режиме разработки. Все соединения осуществляются исключительно по протоколу HTTPS с использованием TLS 1.3, запросы по HTTP автоматически отклоняются с кодом 403. Дополнительно применяется валидация входных данных на уровне сервера с использованием JSON Schema, что предотвращает SQL-инъекции и другие уязвимости.

2.4. Выводы

Разработанная спецификация REST API полностью покрывает функциональность MVP системы, обеспечивая последовательное взаимодействие между фронтендом и бэкендом — от регистрации пользователя до выполнения инвестиционных транзакций через партнеров. Использование стандартных HTTP-методов, четких JSON-схем и единого

формата ошибок упрощает разработку клиентских приложений и тестирование. Безопасность реализована на нескольких уровнях: аутентификация через JWT, шифрование трафика, ограничение скорости и строгая валидация. Архитектура API является stateless, что позволяет легко масштабировать систему при росте нагрузки. В дальнейшем возможно развитие в сторону добавления WebSocket для мгновенных уведомлений, внедрения GraphQL для гибких запросов к рекомендациям, а также генерации интерактивной документации через OpenAPI (Swagger), что повысит удобство интеграции для сторонних разработчиков и партнеров.

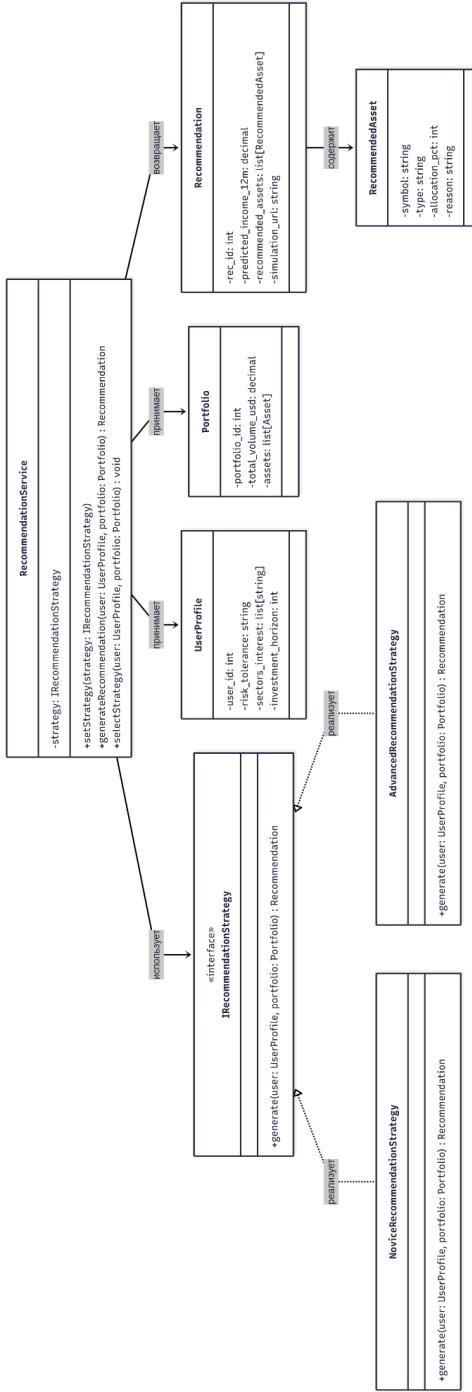
3. Проектирование архитектуры

В разрабатываемой информационной системе центральной задачей является генерация персонализированных инвестиционных рекомендаций, которые должны адаптироваться под уровень пользователя, определяемый объёмом портфеля, результатами адаптивного опроса и поведенческими паттернами. Система различает два ключевых сценария: для новичков предлагаются готовые сбалансированные портфели, а для опытных трейдеров — сложные рекомендации на основе коллаборативной фильтрации, LSTM-прогнозов, анализа волатильности и симуляций «что если». В будущем планируется расширение: интеграция с внешними ML-моделями, стратегии на основе новостного анализа или институциональных движений.

Паттерн **Стратегия** выбран как оптимальное решение, поскольку он позволяет инкапсулировать каждый алгоритм рекомендаций в отдельный класс, обеспечивая:

- Упрощение разработки — логика генерации рекомендаций вынесена из основного сервиса; добавление новой стратегии не требует изменения существующего кода.
- Повышение масштабируемости — стратегии могут выполняться асинхронно, а выбор стратегии осуществляется динамически на основе профиля пользователя.
- Улучшение тестируемости — каждая стратегия тестируется изолированно (unit-тесты), переключение между ними — в интеграционных тестах с моками.
- Гибкость в эксплуатации — возможен runtime-выбор стратегии, A/B-тестирование, временное отключение сложных алгоритмов при высокой нагрузке.

Паттерн применяется в компоненте RecommendationService. Сервис взаимодействует с API-эндпоинтами /recommendations/generate и /recommendations/{rec_id}, описанными в спецификации API. Выбор стратегии происходит на основе /profile и /portfolio.



Интерфейс `IRecommendationStrategy` определяет единый метод `generate()`, принимающий объекты `UserProfile` (данные из `/profile`) и `Portfolio` (данные из `/portfolio`) и возвращающий объект `Recommendation`, соответствующий формату ответа `/recommendations/ {rec_id}`.

Конкретные реализации:

- `NoviceRecommendationStrategy` — формирует 3–5 преднастроенных портфелей на основе `risk_tolerance` и `sectors_interest`.
- `AdvancedRecommendationStrategy` — выполняет коллаборативную фильтрацию по профилям пользователей, LSTM-прогнозирование доходности (интеграция с внешними данными), генерацию поля

- reason с объяснением.
- Класс RecommendationService методом selectStrategy() определяет стратегию.

К преимуществам применения паттерна Стратегия можно отнести:

- Спецификации API — форматы ответов (predicted_income_12m, recommended_assets, reason) унифицированы через общий интерфейс.
- Гибкость расширения — добавление NewsSentimentStrategy или InstitutionalFlowStrategy не нарушит существующий код.
- Поддержка А/В-тестирования — можно случайным образом назначать стратегии группам пользователей через конфигурацию.
- Оптимизация производительности — NoviceStrategy выполняется синхронно, AdvancedStrategy — асинхронно с уведомлением через /notifications.
- Тестируемость — мокирование стратегий в тестах API, проверка корректности reason и allocation_pct.

4. Заключение

Разработанные элементы информационной системы — пользовательский интерфейс, REST API и архитектурные решения (паттерн Стратегия) — представляют собой взаимосвязанную, целостную экосистему, обеспечивающую эффективное функционирование интеллектуальной платформы по рекомендациям инвестиционных портфелей. Их взаимодействие строится на принципах разделения ответственности, масштабируемости, гибкости и безопасности, что критично для MVP и дальнейшего развития.

Пользовательский интерфейс выступает точкой входа для трейдера: он инициирует регистрацию, подключает кошельки, проходит опрос, просматривает портфель и получает рекомендации. Все действия пользователя транслируются в HTTP-запросы к REST API (/auth/register, /profile, /portfolio, /recommendations/generate и др.). API, в свою очередь, выступает единым шлюзом между фронтендом и бэкендом, обеспечивая стандартизованный, документированный и безопасный обмен данными. Использование JWT-аутентификации, валидации JSON-схем, rate limiting и HTTPS гарантирует, что данные пользователя (включая адреса кошельков и результаты опроса) передаются и обрабатываются с соблюдением приватности и GDPR-подобных стандартов.

Архитектурные решения, в частности паттерн Стратегия, напрямую влияют на качество и скорость ответа API. Выбор стратегии происходит на основе данных, полученных через API (/profile, /portfolio), и определяет

содержимое ответа `/recommendations/{rec_id}`. Например, для новичка интерфейс отображает простую карточку с готовым портфелем, а для опытного — детализированную симуляцию с графиками и обоснованием.

В свою очередь, масштабируемость архитектуры (асинхронная генерация рекомендаций, кэширование GET-запросов, stateless API) позволяет интерфейсу оставаться отзывчивым даже при высокой нагрузке. Например, `202 Accepted` при запуске генерации с последующим `polling /recommendations/{rec_id}` предотвращает зависания UI, а уведомления через `/notifications` интегрируются в push-систему и Telegram-бот — всё это отражается в интерфейсе в реальном времени.

API выступает связующим звеном, обеспечивая: единообразие данных между UI и бэкендом; гибкость расширения — добавление новых эндпоинтов (например, `/ab-test/strategy`) без ломания существующего frontend’а; безопасность и контроль — валидация, аутентификация, ограничения.

Исполнитель



Равгейша А. Д.

(подпись)

(Ф.И.О)

Исполнитель



Бабарико В. Д.

(подпись)

(Ф.И.О)

Исполнитель



Петров Е. А.

(подпись)

(Ф.И.О)