

Технический проект системы управления инвестиционными портфелями

1. Постановка задачи

Создать систему управления инвестиционными портфелями, которая будет удобна для разных типов пользователей: инвесторов, советников, администраторов и провайдеров финансовых продуктов.

2. Требования и ограничения

2.1. Макросущности

- Агенты системы: инвесторы, советники (люди и ИИ), администраторы, официальные лица.
- Различные типы инвесторов: нужно учитывать их профили риска.
- Внутренние данные: финансовые инструменты, портфели, стратегии, сделки, исторические рыночные цены, корпоративные действия, отчеты, логи, данные личных кабинетов, данные KYC, техническая документация, пользовательская документация, внутренние корпоративные знания и документы.
- Внешние источники данных: биржевые цены, рекомендации ИИ советников. Автоматическая подгрузка некоторых данных из внешних источников вовнутрь. Интеграции с внешними сервисами (например, API брокеров, банков, платёжных систем, KYC/AML провайдеров)

2.2. Компоненты:

- (I) Формулы в Airtable (логика учета портфеля, расчет доходности).
- (II) Пользовательские интерфейсы (инвесторы, советники).
- (III) Скрипты-процессы (поток событий: клиент, аллокация, трейды).
- (IV) Исторические данные (цены, static instrument data, corp actions).
- (V) Скрипты-агенты (оптимизация: пред-расчеты, кэширование).
- (VI) Скрипты-агенты (проверка целостности данных).
- (VII) Скрипты-агенты (real-time аналитика, AI Advisors).
- (VIII) API с внешними провайдерами (рынки, цены, исполнение сделок).
- (IX) Скрипты-агенты (регуляторные функции: комплаенс, репортинг).

2.3. Производительность

- Время отклика для основных операций пользователя (просмотр портфеля, открытие списка инструментов) не должно превышать 2 секунд.
- Генерация стандартных отчетов не должна занимать более 10 секунд.
- Система должна выдерживать одновременную работу X пользователей (уточняется на этапе проектирования нагрузки).
- Фоновые задачи по загрузке и расчету данных должны завершаться в приемлемые временные окна (например, ночные расчеты должны успевать до начала рабочего дня).

2.4. Надежность

- Доступность системы: 99.0% (уточняется).
- Механизмы резервного копирования и восстановления данных.
- Обработка ошибок и отказов компонентов системы.
- Защита от основных веб-уязвимостей (OWASP Top 10).
- Шифрование чувствительных данных (пароли, API-ключи).
- Разграничение доступа на основе ролей.
- Ведение логов безопасности.
- Хранение, отказоустойчивость и резервное копирование всех внутренних данных.
- >Login через OAuth (Гугл, Эппл), OTP (одноразовые пароли) и MFA (приложение-аутентификатор).

2.5. Удобство Использования (Usability)

- Мобильное приложение.
- Единообразие элементов управления.
- Наличие подсказок. (По клику, тапу, всплывающих)
- Уведомления (в том числе пуш)
- Качественный ввод данных. Простота интерфейсов для разных типов пользователей.
- Адаптивный дизайн для корректного отображения на различных устройствах (десктоп, планшет, мобильные устройства для веб-версии).
- Основной акцент на понятное представление финансовой информации.
- Минимизация количества кликов для выполнения частых операций.
- Пользователи должны понимать, какие данные обновляются автоматически, по каким принципам, в какие моменты времени. Например: если биржевые цены на конкретный день могут

обновляться с задержкой в несколько дней, это должно объясняться пользователям.

- Локализация и поддержка разных языков (?).

2.6. Примеры потоков информации

- Советник создает портфель для Инвестора → добавляет позиции → система автоматически подгружает цены и рассчитывает стоимость.
- Советник создает стратегию → задает модельный портфель → система автоматически подгружает цены и рассчитывает варианты реальных портфелей и их стоимость.
- Система автоматически и регулярно подгружает биржевые цены и обновляет агрегированные таблицы.
- Ребалансировка портфелей проводится как вручную, так и автоматически в ответ на некоторые события (?)
- Все действия и изменения фиксируются для аудита и отчетности.

2.7. Примеры пользовательских сценариев

- "Как инвестор, я хочу добавить новый портфель и видеть его стоимость в реальном времени."
- "Как советник, я хочу создать стратегию и видеть, сколько инвесторов ее выбрали."
- "Как администратор, я хочу заблокировать пользователя при подозрении на мошенничество."
- "Как инвестор, я хочу получить уведомление, если мой портфель сильно отклонился от стратегии."
- "Как пользователь, я хочу видеть историю всех своих операций и изменений в портфеле."

2.8. Ограничения

- Версионирование. Бюджеты на разработку первой и каждой последующей версии системы.
- Требования финансовых и айтишных регуляторов по безопасности.

3. Ключевые концепции и их взаимосвязи

3.1 Стадии проекта

В дальнейшем тексте используется следующая терминология:

- **Pre-MVP** относится к тем элементам системы, которые уже существуют на данный момент (июнь 2025 г.) в виде прототипа Airtable.
- **Basic MVP** относится к продукту, пригодному для демонстрации базового функционала системы, а именно просмотра портфелей, стратегий, трейдов, отчетов. Учет клиентов и их портфелей, сверка с депозитариями и аллокацией по стратегиям. Реализация: low-code Airtable + парсинг данных депозитариев/провайдеров цен.
- **Full MVP** относится к продукту, удовлетворяющему всем требованиям из пункта 2 и фичи: риски, рекомендации, AI советник. Реализация: Supabase + UI + back-end + mobile app.
- **MVP** относится к случаям, когда функции Basic MVP и Full MVP совпадают в каком-то аспекте системы.
- **Advanced** (он же **Post-MVP**) относится к продукту профессионального уровня (уточняется). Обучение на поведении клиентов. Передача торговых приказов по API (в т.ч. авто-сгенерированных).

3.2 Торговые стратегии Simvai — ключевая ценность

MVP (Minimum Viable Product):

- Прототип приложения: регистрация, настройка, 1 стратегия-бот, клиентские отчеты.
- Реализация нескольких алгостратегий.
- **CRM:** Информация о клиентах (каналы привлечения, стоимость, выручка, маржа). Интеграция CRM & PTS.

Advanced:

- Встроенный инструмент для генерации простых стратегий и их бэктестинга через „виртуальные“ TrackingPortfolios (см. ниже).
- Маркетплейс стратегий (уточняется).

3.3 Инструменты собираются в Позиции.

- **Позиции** составляют **Портфели**.
- **ModelPortfolio:**
 - Абстракция, привязана к **Стратегии** (задается **Советником**).
 - Содержит ModelPositions (Инструмент + Вес, сумма весов = 100%).
 - Ребалансируется через ModelRebalancing (модификация весов).
 - StrategicModelPortfolio: еще большая абстракция, базовый набор весов.
- **TrackedPortfolio:**
 - Консолидация одного или нескольких RealPortfolio.

- Нужен для учета стратегии на нескольких счетах клиента.
- Содержит TrackedPositions (консолидация RealPositions).
- Сравнивается с ModelPortfolio.
- Ребалансируется через PortfolioRebalancing (приведение к весам ModelPortfolio).
- TrackedPortfolioCurrency: обычно Preferred Reporting Currency клиента.
- Может быть привязан к **Стратегии** как к бенчмарку или для репликации.
- **RealPortfolio:**
 - Л е ж и т на конкретном DepoAccount (принадлежит **Инвестору в Депозитарии**).
 - Launch Date, RealPortfolio_Currency (MVP: совпадает с валютой DepoAccount).
 - Набор **Позиций**, меняющийся во времени.

3.4 Разделы/Экраны (GUI)

- **Для Инвестора:**
 - Дашборд (обзор всех портфелей, ключевые P&L, предстоящие события).
 - Раздел "Мои Портфели" (список портфелей, детальный просмотр портфеля: состав, позиции, аналитика, история операций).
 - Раздел "Стратегии" (просмотр доступных стратегий, подписка на стратегию).
 - Раздел "Торговля" (создание заявок, просмотр истории заявок и сделок).
 - Настройки профиля.

- **Для Советника:**
 - Дашборд (обзор управляемых стратегий, их эффективность, количество подписчиков).
 - Раздел "Мои Стратегии" (создание, редактирование, настройка аллокаций, просмотр модельной P&L).
 - Раздел "Клиенты" (просмотр инвесторов, следующих его стратегиям, их портфелей - с их согласия).
- **Для Администратора:**
 - Управление пользователями.
 - Управление справочниками (инструменты, депозитарии).
 - Мониторинг системы и фоновых задач.
 - Настройки платформы.

4. Декомпозиция проекта на логические модули

4.1. Модуль пользователей и ролей

- **Что хранит:** Данные пользователей, их роли (инвестор, советник, админ), профили, настройки, связку с персоной.
- **Что делает:** Регистрация, вход, смена пароля, управление профилем, назначение ролей, блокировка/разблокировка.
- **Взаимодействие:** Все остальные модули используют идентификатор пользователя для контроля доступа и персонализации.
- **Примеры сценариев:**
 - Инвестор регистрируется (или его регистрирует его Советник), выбирает тип персоны, настраивает профиль.
 - Администратор блокирует пользователя при подозрении на мошенничество.
- **Pre-MVP, Basic MVP:**

- Управление пользователями и их ролями непосредственно в таблицах Airtable.
- Личный кабинет пользователя read-only, есть только возможность просматривать портфели и формировать отчеты.
- Ручное назначение ролей администратором в Airtable. "Полуавтоматическое" может означать использование простого скрипта Airtable, но основной метод – ручное назначение.
- **Full MVP:**
 - Самостоятельная регистрация пользователей с подтверждением по SMS и через Google OAuth.
 - Личный кабинет пользователей.

4.2. Модуль финансовых инструментов

- **Что хранит:** Справочник инструментов, идентификаторы (ISIN, тикеры и др.), историю цен, корпоративные действия.
- **Что делает:** Добавление/редактирование инструментов.
- **Взаимодействие:** Используется в портфелях, стратегиях, торговых операциях, расчетах P&L.
- **Примеры сценариев:**
 - Администратор добавляет новый инструмент.
 - Система автоматически обновляет цены по расписанию.
- **Pre-MVP:**
 - Ручное добавление и редактирование инструментов в Airtable. "Полуавтоматическое" может означать импорт из CSV-файла, подготовленного вручную.
 - Хранение истории цен и корпоративных действий.
- **Basic MVP**
 - Полноценная реализация структуры PriceEvents и Prices (см. раздел 9.4).

- Обновление цен из внешних источников через Airtable Automations.

4.3. Модуль портфелей

- **Что хранит:** Портфели пользователей (Depo Accounts, Real Portfolios, Tracked Portfolios), их состав (позиции), историю изменений, снимки состояния.
- **Что делает:** Создание портфеля, добавление/удаление позиций, расчет стоимости и P&L, ведение истории.
- **Взаимодействие:** Использует данные инструментов, пользователей, стратегий, торговых операций, фоновых расчетов.
- **Примеры сценариев:**
 - Инвестор создает портфель, добавляет позиции.
 - Система ежедневно обновляет стоимость портфеля.
- **Pre-MVP:**
 - Создание и управление портфелями (DepoAccounts, RealPortfolios, TrackedPortfolios, ModelPortfolios) и позициями в таблицах Airtable.
- **Basic MVP:**
 - Расчет базовой стоимости портфеля и P&L и некоторых других метрик с использованием формул Airtable.

4.4. Модуль стратегий

- **Что хранит:** Стратегии, их параметры, модельные портфели.
- **Что делает:** Создание/редактирование стратегий, задание аллокаций, расчет модельной доходности.
- **Взаимодействие:** Инвесторы могут выбрать стратегию для портфеля, фоновые задачи используют для расчетов.
- **Примеры сценариев:**

- Советник создает стратегию, задает веса инструментов.
- Инвестор выбирает стратегию для своего портфеля.
- **Pre-MVP:**
 - Ручное создание и управление стратегиями и модельными портфелями в Airtable.
 - Задание аллокаций (весов инструментов) в ModelPositions.
- **Basic MV, Full MVP:**
 - Расчет модельной доходности через „виртуальный" TrackedPortfolio со статусом model_backtest.
- **Advanced:**
 - Маректплейс стратегий, возможность (для инвесторов) подписки на стратегии с более сложной логикой.

4.5. Модуль торговли

- **Что хранит:** Торговые поручения, сделки, статусы исполнения.
- **Что делает:** Создание поручений, отслеживание исполнения.
- **Взаимодействие:** Использует портфели, инструменты, интеграции с брокерами, влияет на позиции в портфелях.
- **Примеры сценариев:**
 - Инвестор создает заявку на покупку.
 - Система фиксирует исполнение сделки и обновляет портфель.
- **Pre-MVP, Basic MVP, Full MVP:**
 - Ручное создание торговых поручений (TradeOrders) в Airtable.
 - Присвоение статусов исполнения поручений вручную. "Простые автоматизации" могут включать email-уведомления при ручном изменении статуса TradeOrder администратором, но не автоматическое исполнение или сложную логику обработки статусов.

- **Advanced:**
 - Интеграция с API брокеров для автоматической передачи торговых поручений и получения статусов их исполнения.
 - Поддержка различных типов ордеров и условий исполнения.
 - Алгоритмическая торговля на основе сигналов от стратегий или AI-агентов

4.6. Модуль аналитики (ядро фронт-энда)

- **Что хранит:** ничего.
- **Что делает:** визуализирует данные, предоставляет аналитику по запросу пользователей.
- **Взаимодействие:** Использует все основные модули для сбора данных.
- **Примеры сценариев:**
 - Инвестор просматривает отчет по доходности портфеля.
 - Советник анализирует эффективность своей стратегии.
- **Basic MVP (Airtable):**
 - Базовые дашборды и визуализации, созданные с помощью Airtable Interfaces или Page Designer.
- **Full MVP:**
 - **Web dashboards (React).** Рекомендуется использование Lovable или похожих сервисов быстрой разработки фронт-энда.
- **Advanced:**
 - Использование специализированных баз данных **ClickHouse** или **TimescaleDB** для хранения больших массивов данных аналитики.
 - Расчет сложных метрик в реальном времени.

4.7. Модуль фоновых задач (ядро бэкенда)

- **Что хранит:** Логи фоновых операций, статусы интеграций, историю загрузок данных.

- **Что делает:** Автоматический сбор рыночных данных, оптимизирование баз данных для быстрых запросов к ним.
- **Взаимодействие:** Обновляет базы данных и инициирует события в других модулях.
- **Basic MVP (Airtable):**
 - Использование Airtable Automations (включая шаги "Run a script" с JavaScript) для выполнения задач по расписанию или по триггерам (например, обновление P&L, создание позиций после сделок).
 - Ограничения Airtable Automations: короткое время выполнения (обычно ~30 секунд), лимиты на API-вызовы, частоту запусков. Не подходят для длительных или ресурсоемких операций.
 - Оптимизация баз данных в Airtable сводится к правильной структуре таблиц, использованию вьюшек.
- **Full MVP (Supabase):**
 - Внедрение полноценной системы для непрерывного выполнения фоновых задач с использованием **Supabase edge functions** или других вариантов облачных функций.
 - Масштабируемость и отказоустойчивость бэкенда.
- **Advanced:**
 - Работа AI-агентов.
 - Взаимодействие с продвинутой аналитикой.

4.8. Модуль уведомлений

- **Что хранит:** Шаблоны уведомлений, настройки пользователей, историю отправленных уведомлений.
- **Что делает:** Отправляет push, email и SMS уведомления, позволяет настраивать типы уведомлений.

- **Взаимодействие:** Интегрируется со всеми модулями для отправки важных событий пользователям.
- **Примеры сценариев:**
 - Система отправляет уведомление о исполнении торгового поручения.
 - Пользователь получает alert о критическом отклонении портфеля.
- **Full MVP** (на более ранних этапах модуль отсутствует):
 - Отправка email-уведомлений на основе триггеров в Airtable (например, изменение статуса TradeOrder, создание новой записи).
 - Поддержка push-уведомлений.
 - Гибкая настройка предпочтений по уведомлениям для пользователей.

4.9. Модуль документов и отчетности

- **Что хранит:** документы, отчеты, договоры, регламенты.
- **Что делает:** позволяет обмениваться юридическими документами.
- **Взаимодействие:** использует модуль аналитики для экспорта отчетов.
- **Примеры сценариев:**
 - Экспортируются ежемесячные отчеты.
 - Пользователь загружает скан паспорта.
 - Советники обмениваются знаниями.
- **Реализация (Nextcloud + Airtable):**
 - Хранение документов (сканы, договоры и т.д.) в Nextcloud.
 - Генерация базовых отчетов (например, состав портфеля, P&L) с использованием Airtable Page Designer или Scripting App, с возможностью экспорта в PDF.

- Связывание ссылок на документы из Nextcloud в Airtable при необходимости.

5. Детализация баз данных и логики Simvai MVP

5.1. (I) Instruments: Финансовые инструменты - статические данные

- **MVP:** 10 инструментов (включая USD cash, EUR cash).
- Инструмент статичен (раз созданный, не умирает), хотя все может меняться (вкл. true ISIN).
- Цена, corp actions - отдельные базы.

Структура базы Instruments:

1. **IDs:** (Блок, 7 полей: 4 string, 2 flag + другие) 1.1. Primary key (24 alphanumeric, последние 12 = ISIN). 1.2. ISIN 1.3. trueISIN 1.4. flag if internal ISIN = true ISIN 1.5. flag if ISIN ever changed 1.6. PreviousISIN 1.7. Другие идентификаторы: Решено использовать фиксированные поля для других типов ID, даже если некоторые из них будут часто оставаться незаполненными. Конкретный набор этих полей требует определения (например, OtherID_BBG_UniqueID, OtherID_Reuters_RIC и т.д.).
2. **Tickers:** (Блок, 6 полей string) 2.1-2.4: TickerBloomberg, TickerRefinitiv, TickerMicex, TickerOther 2.5: PrimaryTicker.Ticker 2.6: PrimaryTicker.Source (Bloomberg, Refinitiv, ...)
3. **AssetClass (equity, fixed income, alternative, FX):** (Блок, 3 поля) 3a) Grouping1: (cash, equity, bond, alternative) 3b) Grouping2: (зависит от Grouping1, как в Wize: Public equity, ...) 3c) Grouping3: (зависит от Grouping1 и Grouping2)

4. **Wrapper:** (2 поля) 4.1. Type: (Outright, Derivative, Private_Contract, Collective) 4.1.1. Outright: (stock, bond, cash, digital currency, physical gold/Real Estate/lifestyle assets) 4.1.2. Derivative: (futures, options, swaps, CDS, CLNs, structured notes) 4.1.3. Private_Contract: (Private Loan, SAFE, digital currency in DeFi) 4.1.4. Collective: Типы коллективных инвестиций включают: - Regulated collective: ETF (может быть одновременно и Transferable security), UCITS fund, mutual fund - Semi-regulated collective: open-ended fund, AMC, AIF - Non-regulated: closed-ended fund 4.2. Liquidity: (Intra-day, daily, weekly, monthly, quarterly, annually, 2-3y, 3-5y, 6-10y, 10+)
5. **Description data:** (Блок, 6 простых полей)
- Short_Name
 - Full_Name
 - Country
 - Region
 - Sector
 - Sub-sector (зависит от Sector)
 - USD Amount Outstanding (заполняется скриптом-агентом)
6. **Flag isVirtual2settle:** (если виртуальный инструмент на нерассчитанную сделку, обычно создается Депозитарием)
- InstrumentID (плейсхолдер для инструмента)
7. **Transferability:** (single select)
- Transferable Financial security: (stock, bond, ETF, structured note)
 - Collective investment: (Regulated: ETF, UCITS, mutual fund; Semi-regulated: open-ended fund, AMC, AIF; Non-regulated: closed-ended fund)
 - Currency: (cash on account, money-market fund, physical gold, digital currency on personal wallet)

- Derivative Contracts: (futures, options, warrants, swap, CDS - exchange-traded or OTC)
- Private Contracts: (Private Loan, SAFE, digital currency in DeFi)
- NON-FINANCIAL ASSETS: (Physical Real estate, Lifestyle assets)

8. **CorpActions_Link:** (список всех CorpActions, включая изменение amount outstanding, выплату купонов/дивидендов)

5.2. (V) Масштабы данных (Исторические данные)

- **Пользователи:** 10,000s
- **Инструменты:** 1,000s
 - 500 акций S&P 500
 - top 30 crypto coins
 - top 10 currencies
 - 200 ETFs
 - ~200 ключевых акций вне США
- **Глубина истории:** 5-10 лет
- **Частота ценовых данных:**
 - каждый день: минимум за 5 лет
 - каждый час: для ликвидных инструментов в реальных портфелях
 - каждый месяц: 5+ лет назад

5.3. (2) CorpActions: История корпоративных действий по отслеживаемым инструментам

- Отслеживание может начаться, когда инструмент появляется в реальном или модельном портфеле.

5.4. (3) Prices: Ценовые векторы и PriceEvents

Для эффективного управления ценовыми данными и курсами валют вводится понятие PriceEvent (событие цены), которое представляет собой

логический момент времени или пакет обновлений, к которому относятся одна или несколько ценовых записей.

Требования по частоте обновления цен и курсов:

1. Цены всех отслеживаемых инструментов: ежедневно (цена закрытия).
2. Цены инструментов, находящихся в реальных портфелях пользователей: каждый час.
3. Курсы всех используемых в системе валют (ожидается до 10 валют): каждый час.
4. Курсы валют и цены инструментов, непосредственно задействованных в торговой операции (сделке): фиксируются на момент каждой сделки.

Хранение валют: Валюты (например, USD, EUR) трактуются как финансовые инструменты и хранятся в общей таблице Instruments. Курсы обмена валют (например, EUR/USD) сохраняются как цены для соответствующего инструмента-валюты (EUR) в базовой валюте (USD) в таблице Prices.

A. Таблица PriceEvents

Фиксирует каждое событие получения или обновления цен.

- PriceEventID (Primary Key, INTEGER/UUID): Уникальный идентификатор события цены.
- EffectiveTimestamp (DATETIME): Точное время (с точностью до секунд или миллисекунд, если необходимо), на которое актуальны цены/курсы в данном событии. Это основной timestamp, используемый для запросов типа "какая цена была на момент времени X?".
- NominalDate (DATE): Номинальная дата, к которой относятся цены в данном событии (например, дата закрытия торгов). Обычно это дата, извлеченная из EffectiveTimestamp. Используется для реализации

логики `status = latest/hist` при сравнении цен на одну и ту же дату.

- **EventType (STRING):** Тип события, определяющий контекст полученных цен. Примеры значений:
 - `'DailyClose'` (ежедневные цены закрытия по всем инструментам)
 - `'HourlyInstrument'` (ежечасное обновление цен по инструментам в активных портфелях)
 - `'HourlyFX'` (ежечасное обновление курсов валют)
 - `'TradeRelated'` (цены и курсы, зафиксированные в момент совершения сделки)
 - `'ManualUpdate'` (цены, введенные или скорректированные вручную)
- **SourceDescription (STRING, опционально):** Текстовое описание источника данных для данного события (например, "Feed X", "API Брокера Y", "Расчет по сделке #123", "Корректировка администратором").
- **SystemEntryTimestamp (DATETIME):** Системное время, когда данное событие `PriceEvent` было зарегистрировано в базе данных.

В. Таблица Prices (обновленная структура)

Хранит конкретные ценовые записи, каждая из которых связана с определенным `PriceEvent`.

- **PriceRecordID (Primary Key, INTEGER/UUID):** Уникальный идентификатор конкретной ценовой записи.
- **PriceEventID_FK (FOREIGN KEY):** Ссылка на `PriceEventID` из таблицы `PriceEvents`, указывающая, к какому событию относится данная цена.

- `InstrumentID_FK` (FOREIGN KEY): Ссылка на `InstrumentID` из таблицы `Instruments`. Указывает, для какого финансового инструмента (акции, облигации, валюты и т.д.) приведена цена.
- `PriceValue` (DECIMAL): Числовое значение цены или курса. Точность должна быть достаточной для всех типов инструментов и валют.
- `PriceCurrencyID_FK` (FOREIGN KEY): Ссылка на `InstrumentID` (из таблицы `Instruments`, где инструмент является валютой), указывающая валюту, в которой выражена `PriceValue`. Например, для акции Apple цена будет в USD, для курса EUR/USD `InstrumentID_FK` будет EUR, `PriceValue` - курс, а `PriceCurrencyID_FK` - USD.
- `isClosePrice` (BOOLEAN): Флаг (true/false), указывающий, является ли данная цена официальной ценой закрытия. Особенно актуально для `PriceEvents` с `EventType='DailyClose'`.
- `Status` (STRING): Принимает значения 'latest' (актуальная) или 'hist' (историческая). Логика управления статусом:
 - Когда через новый `PriceEvent` поступает новая ценовая запись для комбинации (`InstrumentID_FK`, `PriceEvents.NominalDate`), эта новая запись получает статус 'latest'.
 - Если для той же комбинации (`InstrumentID_FK`, `PriceEvents.NominalDate`) уже существует другая запись в `Prices` со статусом 'latest', то статус этой старой записи изменяется на 'hist'.
 - `SystemUpdateTimestamp` у старой записи при смене статуса на hist не изменяется.
- `SystemUpdateTimestamp` (DATETIME): Системное время создания или последнего изменения (например, смена статуса) данной

записи PriceRecord. Соответствует полю "timestamp of update" из предыдущего описания.

5.5. (4) Depositaries (=Banks/Brokers)

- **Структура:** Name, contact details (вкл. имена), working hours, ...
- **MVP:** 2 (виртуальных). Позже тестирование Interactive Brokers и/или SingleBroker по API.

5.6. (5) Scripts (agents, processes)

- AI_Advisor(s), AI_RiskMonitor_RealTime, AI_RiskMonitor_Daily, AI_Rebalancer, AI_Reporter. (В MVP эти агенты представляют собой концептуальные заготовки или не реализуются; их полноценная AI-логика является целью Post-MVP).

5.7. (6) Personas: Профили инвесторов

- **Структура:** PersonaName, PersonaDescription, Strategies_Link (автоматически: подходящие/рекомендованные), RecommendedReportingFrequency (monthly, quarterly, weekly).
- **MVP:** 3 профиля (Busy Professional, Cautious Rent-seeker, Market Geek), по 1 рекомендованной стратегии для каждого.
- **Свойства Persona:**
 - Предлагаемые стратегии (риск-доходность, ликвидность)
 - Уровень детализации отчетности
 - Жесткость риск-мониторинга
 - Цели в \$
 - Инвестиционный горизонт
 - Предпочтительная валюта отчетности
 - Предпочтительный язык
 - Список TrackedPortfolios

5.8. (7) Пользователи

- **Структура:** Name, Address, Email..., Role (Инвесторы, Советники, Администраторы), status (active, pending, suspended, test), personal info, account credentials.
- **Инвесторы:** имеют DepoAccounts, RealPortfolios, TrackedPortfolios, Persona; access level (free, standard, premium); detail level (опции, репортинг).
- **Советники:** имеют Strategy_Class, ModelPortfolios. (Могут быть реальные пользователи и AI).
- **Администраторы.**
- **MVP:** 4 пользователя (1 админ, 1 советник, 2 инвестора).
- **Основное свойство пользователя-инвестора:** 1 Persona из 3 (Busy Professional, Cautious Rent-seeker, Market Geek).

5.9. (11) Иерархия Позичий (по повышению уровня абстракции)

- DepoPositions → RealPositions → TrackedPositions → ModelPositions
- У всех: Timestamp of update, status latest/hist.
- У ModelPositions все timestamp of update одновременны при каждой ребалансировке (модификация через ModelRebalancing).

(11a) DepoPositions

- **Структура:** InstrumentID, DepoVolume, Date, Time, DepoReportDate, In_DepoAccount, status (latest/hist), DepoCurrency, DepoFXrate, DepoPrice, DepoAmount.
- Дублирует информацию из отчетов депозитариев (для реконсиляции).
- **Скрипт-агент:** Проверка $\text{DepoPrice} * \text{DepoVolume} * \text{DepoFXrate} = \text{DepoAmount}$.

- **Скрипт-агент:** Ежедневная загрузка и парсинг депо-отчетов. (В MVP – ручной импорт данных отчетов депозитариев администратором, автоматизация - Post-MVP).
- Меняются от новых отчетов, трейды не влияют.
- **(11b) RealPositions**
 - **Структура:** InstrumentID, volume, pending_volume, Date, Time, In_RealPortfolio, status (latest/hist).
 - НЕ содержит цены, НЕ содержит DepoAccount.
 - Содержит PendingCashflows (CorpAction влияют, учтены недорассчитанные сделки, flag isVirtual2settle = True у инструмента).
 - status latest/hist: Новые позиции latest, существующая с тем же InstrumentID и RealPortfolio меняет статус на hist.
 - volume: >0 (лонг), <0 (шорт).
 - Меняются от: Trades (статус старых latest -> hist), CorpAction (сплиты меняют объем).
 - **MVP:** ~10 позиций, включая изменения.
- **(11c) TrackedPositions**
 - **Структура:** InstrumentID, volume, pending_volume, Date, Time, In_TrackedPortfolio, underlying RealPositions (list), status (latest/hist), split_status (yes/no).
 - split_status: yes, если соответствует нескольким RealPositions.
 - Содержит PendingCashflows.
 - TrackedPositions с InstrumentID=cash могут быть split_status=yes из-за pending cashflows.
- **(11d) ModelPositions**
 - **Структура:** InstrumentID, MPweight, In_ModelPortfolio, Date, Time.

- Не содержит цен, чистая абстракция (front office).
- Одна из ModelPositions всегда Cash (вес может быть 0).

9.10. Иерархия Портфелей (по повышению уровня абстракции)

- DepoAccounts → RealPortfolios → TrackedPortfolios → ModelPortfolios
- **Общие свойства:** status (pre-launch, active, closed, suspended, wind-down), access rights (view, trade, transfer in/out), Timestamp of update.
- НЕТ ссылок на трейды/ордера (нужны только для реконсиляции скриптами).

(12a) DepoAccounts

- **Структура:** UserID (Investor), Depository, AccCurrency, status, etc.
- Содержит ссылку на RealPortfolio.
- **MVP:** DepoAccounts <-> RealPortfolios. AccCurrency = PortfolioCurrency (принудительно).
- RealPortfolio НЕ МОЖЕТ быть на нескольких DepoAccounts (MVP: 1 DepoAccount = 1 Real Portfolio).
- TrackedPortfolio может быть комбинацией нескольких RealPortfolios.
- **(12b) RealPortfolios**
 - **Структура:** UserID (Investor), DepoAccount, LaunchDate, PortfolioCurrency.
 - Содержит RealPositions.
 - НЕ привязан к Strategy / ModelPortfolios (только реальные позиции).
 - Нужен для реконсиляций.

- **Пользовательский интерфейс (View):** asset statement. По умолчанию не виден пользователю (флажок "показать отдельные DepoAccounts").
- **MVP:** 3 портфеля (2 для 1 инвестора).
- **(12c) TrackedPortfolios**
 - **Структура:** UserID (Investor), LaunchDate, ReportingCurrency, RealPortfolios_Link (list), split_status, AdvisoryStatus, BenchmarkStrategy_Link, Strategy_Link, ModelPortfolio_Link.
 - `split_status`: yes, если соответствует нескольким RealPortfolios.
 - У пользователя всегда есть хотя бы один TrackedPortfolio (консолидация BCEX RealPortfolios).
 - **MVP:** 4 портфеля (3 для 1 инвестора: 2 реальных, 1 консолидированный).
 - Пользователь МОЖЕТ создать TrackedPortfolio с любой прошлой даты.
 - **Пользовательский интерфейс (Form):** привязка к стратегии, к бенчмарку.
 - Содержит TrackedPositions (консолидация RealPositions, кроме бэктеста).
 - Содержит ЦЕНЫ (область - Middle Office, client reporting).
 - **С к р и п т - а г е н т (ночной):** Собирает TrackedPositions из RealPositions + Trades.
 - У с л о в и е :
Ежедневный PriceEvent с isClosePrice=True по всем инструментам.

- Расчеты: $\text{Volume} * \text{Current Price} = \text{Current Amount in InstrumentCurrency}$; $+ \text{CurrentFXrate} = \text{CurrentAmount in ReportingCurrency}$.
- Даты: 1-й покупки, средней покупки (ресет при полной продаже).
- AvgEntryPrice (ресет при полной продаже).
- **Финансовый Результат (P&L):**
 - Received Income (дивиденды, купоны)
 - Accrued Income (дивиденды, купоны)
 - $\text{RealizedPnL} = (\text{Цена продажи} - \text{Средняя цена}) * \text{Объем трейда} + \text{Received Income}$
 - $\text{UnrealizedPnL} = (\text{Last price} - \text{Средняя цена}) * \text{Объем позиции} + \text{Accrued Income}$
 - $\text{TotalPnL} = \text{RealizedPnL} + \text{UnrealizedPnL}$
 - В валюте позиции и портфеля.
- AdvisoryStatus: (self_advised / strategy_full / strategy_advisory / user_backtest / model_backtest).
 - strategy_full может генерировать авто-ордера.
- Привязан к ModelPortfolio.
- **Пользовательский интерфейс (View):** сравнение весов TrackedPortfolio с ModelPortfolio.
- **Скрипт-процесс:** RebalanceTrackedToModel (приведение весов).
 - Выход: предложенный список ордеров. (Если AdvisoryStatus=Strategy_full и есть доступ у AI.AutoRebalancer (Post-MVP), ордера идут на исполнение).
- **Пользовательский интерфейс (View):** asset statement с Position PnL.

- **Пользовательский интерфейс (View+скрипт):** анализ доходности (абсолютный, FX effect, против Strategy, против BenchmarkStrategy), анализ рисков.
- **(12d) ModelPortfolios**
 - **Структура:** StrategyID, LaunchDate.
 - Содержит ModelPositions, сумма MPweights = 100%.
 - **Скрипт-агент:** Проверяет условие суммы весов.
 - **Интерфейс-процесс:** RebalanceModelWeights (получает новые веса по стратегии).
 - Ручной итеративный процесс: уточнение/изменение текущих весов, добавление позиций, проверка суммы весов (приведение к 100% равномерно / за счет кэша / вручную).
 - **Доходность ModelPortfolio:** Скриптом создается TrackedPortfolio (AdvisoryStatus = model_backtest), без привязки к RealPortfolio. Виртуальные позиции создаются на каждую дату в ModelRebalancings (исполнение по price isClosePrice=True).
- **9.10.5. (12e) StrategicModelPortfolios**
 - То же, что ModelPortfolio, но веса зафиксированы (обычно equal-weight).
 - При запуске стратегии веса ModelPortfolio переинициализируются извне.

5.11. TradeOrders: Заявки на сделки

- **Структура:** InstrumentID, In_DepoAcc, In_RealPortfolio, volume, buy/sell, market/limit/stop, limitprice, execution_status, Trades_Link, avg_done, amount_done, timestamp of update.

- `execution_status`: (pending, placed, done, partiallyDonePlaced, partiallyDoneCancelled, cancelled, rejected).
- `avg_done`, `amount_done` - посчитаны из порожденных трейдов.
- **MVP** (`status latest/hist`): не надо хранить (согласно разделу 12.2, достаточно текущего статуса).
- Ордер: (а) по счету `DepoAccount` (важно для back office/регулятора), (б) по портфелю `RealPortfolio` (важно для middle office, генерируется стратегией/советником).
- **MVP**: Ордер от `RealPortfolio`.
- **Скрипт-агент**: `Compliance Pre-Trade`.
- Почти то же, что трейд, но до исполнения.
- НЕ СУЩЕСТВУЕТ `Trades` БЕЗ `TradeOrder`.
- Одна `TradeOrder` может порождать 0+ `Trades`.
- `TradeOrder_Price` (вместо цены исполнения).
- `TradeOrder_Type`: (market (цена не указана), limit, stop).
- `TradeOrder_status`: (исполнена, исполнена частично/остаток снят, исполнена частично/остаток на исполнении, отменена...). Если частично - сколько исполнено, средняя цена.

5.12. Trades: Сделки

- **Структура**: `InstrumentID`, `In_DepoAcc`, `In_RealPortfolio`, `volume`, `buy/sell`, `ExecutionPrice`, `TradeOrder_Link`, `status` (pending/settled/reversal).
- **MVP**: 5 `Trades` из 10 `TradeOrders`.
- **Скрипт-агент**: `AML/MAR Compliance`.
- **Скрипт-процесс**: Создает `RealPosition` (latest), меняет статус предыдущей на hist.
- Иницируется: User (Investor/Advisor) или `AI Agent` (`AI_Rebalancer`).

- (Изменение ModelPortfolio - это ModelRebalancing, не Трейд. Иницируется Советником (человек/AI)).

Детализация полей Trades:

1. RealPortfolio_ID (включает User_ID & DepoAccount_ID)
2. Instrument_PrimaryID
3. buy or sell
4. объем бумаг
5. объем в валюте инструмента
6. валюта инструмента (дублирование)
7. Ave_Execution_Price (средняя цена исполнения)
8. дата и время исполнения (последние, если несколько)
9. курс валюты инструмента при исполнении (средняя)
10. сумма в валюте RealPortfolio_Currency
11. Initiator (Investor, Advisor: Advisor_ID, AI_Rebalancer)
12. TradeOrder_Link

5.13. PortfolioInOut: Вводы-выводы средств

- **Структура:** UserID (Investor), In_DepoAccount, In_RealPortfolio, in/out, type (partial/full).
- **Скрипт-процесс:** Создает RealPosition в кэше в валюте портфеля (latest), меняет статус предыдущей на hist.
- **MVP:** 3 начальных ввода, 1 дополнительный ввод, 1 частичный вывод.

5.14. Strategies: Стратегии

- **Структура:** создатель UserID (Advisor), LaunchDate, PersonalID (подходящие/рекомендованные), StrategicModelPortfolio_Link, ModelPortfolio_Link, StrategyClass, flag usedAsBenchmark.
- **MVP:** 3 стратегии, по 1 каждого StrategyClass: Buy & Hold, Algorithmic, Discretionary.
- **status:** active, closed, test, suspended.

- **LaunchDate:** может быть давно (при наличии бэктеста).
- **Strategy_currency:** USD (по умолчанию).
- **strategy asset class:** Equity, Bonds, Equity-Bonds mix, digital currencies, Global Asset Class Rotation. **MVP:** текстовое описание.
- Содержит **ModelPositions** (Инструмент, % портфеля (сумма=100%), дата).
- **Доходность Стратегии:** Считается с помощью виртуального **TrackedPortfolio** (**AdvisoryStatus=model_backtest**).
- **Бенчмарки:** Некоторые стратегии **usedAsBenchmark=yes**. Всегда есть **CashStrategy** (0% годовых) - бенчмарк по умолчанию.
- Привязка к бенчмарку с любой даты (позже **Launch Date**), может меняться задним числом.

5.15. **ModelRebalancings:** Ребалансировки модельного портфеля

- **Структура:** **ModelPortfolioID**, **Date**, **Time**, **ModelPositions_Links** (модельные позиции с соотв. датой/временем).

5.16. **Важная задача: Reconciliation (стадия Advanced)** - проверка целостности данных

1. **ReconcileRealPortfolioVsDepoAccount:** Проверка совпадения портфеля по данным депозитария с нашим учетом (построчно по позициям).
2. **ReconcileRealPortfolioVsTrades:** Проверка, что портфель однозначно восстанавливается по сделкам из **Trades** и **CorpActions**.
3. **ReconcileDepoPrices:** Проверка совпадения цен одного инструмента в разных депозитариях (точное совпадение для цен типа **Close**).

6. Скрипты системы

В системе предусмотрены различные типы скриптов для автоматизации процессов и поддержания целостности данных. Они делятся на скрипты-процессы и скрипты-агенты, они же **фоновые задачи**. В данном разделе собраны вместе все упомянутые ранее скрипты.

6.1. Скрипты-процессы

Скрипты-процессы отвечают за управление потоками событий и выполнение конкретных операций в системе.

1. Обработка корпоративных действий (сплитов):

- **Описание:** При обнаружении корпоративного действия типа "сплит" для финансового инструмента, скрипт автоматически создает новые ценовые записи со статусом latest и изменяет статус существующих ценовых записей на hist. Timestamp of update для новых цен устанавливается на момент работы скрипта.
- **Затрагиваемые базы:** Prices, CorpActions.

2. Создание/обновление позиций после сделок (Trades):

- **Описание:** После фиксации сделки (Trade), скрипт создает новую реальную позицию (RealPosition) со статусом latest. Если для данного инструмента и портфеля уже существует позиция со статусом latest, ее статус изменяется на hist.
- **Затрагиваемые базы:** Trades, RealPositions.

3. Обработка вводов/выводов средств (PortfolioInOut):

- **Описание:** При операции ввода или вывода средств, скрипт создает позицию (RealPosition) в кэше в валюте портфеля со

статусом `latest`. Статус предыдущей аналогичной позиции (если есть) изменяется на `arc`.

- **Затрагиваемые базы:** PortfolioInOut, RealPositions.

4. Ребалансировка отслеживаемого портфеля к модельному (RebalanceTrackedToModel):

- **Описание:** Скрипт-процесс, инициирующий приведение весов отслеживаемого портфеля (TrackedPortfolio) в соответствие с весами модельного портфеля (ModelPortfolio).
- **Выход:** Генерирует предложенный список торговых ордеров. Если `AdvisoryStatus` портфеля установлен как `Strategy_full` и у `AI.AutoRebalancer` (Post-MVP) есть доступ к портфелю, ордера передаются на исполнение.
- **Затрагиваемые базы/модули:** `TrackedPortfolios`, `ModelPortfolios`, `TradeOrders`, Модуль Торговли.

5. Управление потоком событий (общие):

- **Описание:** Категория скриптов, отвечающих за логику потока событий в системе, таких как появление нового клиента, модификация аллокации в стратегиях, обработка трейдов по портфелю.
- **Затрагиваемые базы / модули:** Пользователи, Стратегии, Портфели, Торговля и др.

6.2. Скрипты-агенты задачи

Скрипты-агенты выполняют мониторинг, оптимизацию и обеспечивают регуляторные функции. Они же на стадии Full MVP - **фоновые задачи**.

1. Оптимизация данных (пред-расчеты, кэширование):

- **Описание:** Общая категория скриптов, направленных на повышение производительности системы путем выполнения предварительных расчетов и кэширования часто запрашиваемых данных.

2. Проверка целостности данных (Reconciliation):

- **Описание:** Набор скриптов для сверки данных между различными частями системы и внешними источниками. (Детально описаны в разделе 9.16, не для MVP).
 - ReconcileRealPortfolioVsDepoAccount
 - ReconcileRealPortfolioVsTrades
 - ReconcileDepoPrices

3. Real-time аналитика по портфелям и рынкам, AI Advisors:

- **Описание:** Скрипты, предоставляющие аналитическую информацию в реальном времени и реализующие функции AI-советников. (В MVP эти агенты представляют собой концептуальные заготовки или не реализуются; их полноценная AI-логика является целью Post-MVP).
- **Конкретные реализации (имена агентов):**
 - AI_Advisor(s)
 - AI_RiskMonitor_RealTime
 - AI_RiskMonitor_Daily
 - AI_Rebalancer (также участвует в процессе RebalanceTrackedToModel, полноценная автоматизация - Post-MVP)
 - AI_Reporter

4. Регуляторные функции (комплаенс, регуляторный репортинг):

- **Описание:** Скрипты, обеспечивающие соблюдение регуляторных требований, включая AML/MAR комплаенс и подготовку отчетов для регуляторов. (Полная автоматизация и сложные проверки -

Post-MVP. В MVP могут быть реализованы простейшие проверки формулами или ручные чек-листы).

- **Конкретные реализации:**
 - **Compliance Pre-Trade:** Скрипт-агент, выполняющий проверки на соответствие регуляторным требованиям перед размещением торгового ордера.
 - **AML/MAR Compliance (для Trades):** Скрипт-агент, проверяющий сделки на предмет отмывания денег и манипулирования рынком.

5. Проверка данных депозитарных позиций:

- **Описание:** При обработке данных из отчетов депозитариев, скрипт проверяет корректность парсинга ($\text{DepoPrice} * \text{DepoVolume} * \text{DepoFXrate} = \text{DepoAmount}$).
- **Затрагиваемые базы:** DepoPositions.

6. Ежедневная загрузка и парсинг депо-отчетов:

- **Описание:** Скрипт-агент, который каждый день автоматически загружает отчеты из депозитариев и разбирает их для обновления DepoPositions. (В MVP – ручной импорт данных отчетов депозитариев администратором, автоматизация - Post-MVP).
- **Затрагиваемые базы:** DepoPositions, внешние источники данных (депозитарии).

7. Ночной сбор TrackedPositions и расчет P&L:

- **Описание:** Ежедневно (ночью) собирает TrackedPositions на основе RealPositions и Trades. Рассчитывает финансовый результат (P&L) для TrackedPortfolios.

- **Условие:** Требуется наличия ежедневного PriceEvent с isClosePrice=True по всем инструментам.
- **Расчеты:** $\text{Volume} * \text{Current Price} = \text{Current Amount in InstrumentCurrency}$; $+ \text{CurrentFXrate} = \text{CurrentAmount in ReportingCurrency}$. Расчет Received Income, Accrued Income, RealizedPnL, UnrealizedPnL, TotalPnL.
- **Затрагиваемые базы:** TrackedPositions, RealPositions, Trades, Prices.

8. Проверка суммы весов в модельном портфеле:

- **Описание:** При модификации ModelPortfolio скрипт-агент постоянно проверяет, что сумма весов (MPweights) всех позиций равна 100%.
- **Затрагиваемые базы:** ModelPortfolios, ModelPositions.

9. Заполнение USD Amount Outstanding в Instruments:

- **Описание:** Скрипт-агент, который заполняет поле USD Amount Outstanding в базе данных Instruments.
- **Затрагиваемые базы:** Instruments.

7. Пояснения по реализации MVP

7.1. MVP Basic

- **Бэкенд:** Airtable (v1). Все основные данные и бизнес-логика MVP реализуются в Airtable.
- **CRM:** Airtable CRM (v1) для данных о клиентах и взаимодействиях. Nextcloud (текущая версия, например, Hub 9

v30.0.9 или новее) используется для хранения документов (Модуль 4.11).

- **Фронтенд:** чисто вьюшки Airtable, показывающие портфели и отчеты.
- **Основные функции прототипа:** Регистрация пользователей, базовая настройка, 1 стратегия-бот, генерация основных клиентских отчетов.
- **Сущности и логика:** Реализация ключевых сущностей (Инструменты, Портфели, Пользователи и т.д.) и основной логики их взаимодействия в Airtable. Логика см. Раздел 4 „Логические Модули“; детализация см. Раздел 5 „Детализация“
- **Пользовательские интерфейсы:** Разработка черновых (базовых, функциональных) макетов пользовательских интерфейсов для основных сценариев. — для последующей реализации в Фулл МВП;
- **Фоновые задачи и скрипты:** реализуются через Airtable Automations (включая шаги "Run a script" с JavaScript) и формулы Airtable. См. Раздел 6 „Скрипты“

7.2. MVP Full:

- **Бэкенд & Фронтенд:** Переходим на **Supabase + React**.
- Полноценные, масштабируемые фоновые задачи
- **Reconciliation (Сверка данных)**
- Полная автоматизация загрузки и парсинга депозитарных отчетов из внешних источников.
- С л о ж н ы е A I -
функции: AI_Advisor(s), AI_RiskMonitor_RealTime, AI_Reba
lancer (для автоматического исполнения ордеров).

- Мобильное приложение (уточняется)

7.3. Детализация пользовательскому интерфейсу (Фронтенд)

На основе примеров из раздела 3.5 и с учетом знания детализации сущностей, MVP фронтенд должен включать следующие ключевые экраны и функции:

- **Для Инвестора:**
 - **Дашборд:** Просмотр списка своих TrackedPortfolios с их текущей расчетной стоимостью (на основе последних загруженных цен).
 - **Раздел "Мои Портфели":**
 - Список TrackedPortfolios пользователя.
 - Детальный просмотр одного TrackedPortfolio: состав (TrackedPositions), их объемы, последние известные цены, расчетная стоимость каждой позиции и всего портфеля.
 - Возможность создать новый (пустой) TrackedPortfolio (указав название и валюту отчётности) . Привязка RealPortfolios к TrackedPortfolio осуществляется администратором или через скрипт в Airtable.
 - **Раздел "Стратегии":**
 - Просмотр списка доступных Strategies (название, описание, создатель-советник).
 - Возможность "подписаться" на стратегию (выбрать стратегию для своего TrackedPortfolio как Strategy_Link или Bench

markStrategy_Link - фактическая привязка и логика выполняются в Airtable, фронтенд иницирует запрос).

- **Раздел "Торговля":**
 - Форма для ручного создания TradeOrder: выбор RealPortfolio, инструмента из справочника, указание объема, направления (buy/sell), типа заявки (для MVP достаточно "market order").
 - Просмотр списка своих TradeOrders и их текущих статусов (статусы обновляются вручную администратором в Airtable, или через простые автоматизации).
- **Настройки профиля:** Просмотр базовых данных своего профиля (например, имя, email). Редактирование не входит в MVP. μ
- **Для Советника:**
 - **Дашборд:** Список созданных им Strategies.
 - **Раздел "Мои Стратегии":**
 - Список созданных Strategies.
 - Возможность создать новую Strategy (название, описание).
 - Для созданной Strategy возможность определить/отредактировать ее ModelPortfolio: выбор инструментов из справочника и указание их целевых весов (MPweight). Сумма весов должна контролироваться (в идеале на фронтенде, но как минимум проверка в Airtable).
- **Для Администратора:**
 - **Управление пользователями:** Список всех пользователей, возможность изменить роль пользователя, статус (active, suspended).
 - **Управление инструментами:** Список всех Instruments, возможность добавить новый инструмент (ручной ввод всех

необходимых полей согласно структуре в Airtable). Возможность вручную обновить цену для инструмента.

- **Управление портфелями:** Просмотр списка RealPortfolios и TrackedPortfolios. Возможность вручную создать RealPortfolio и привязать его к DemoAccount и пользователю. Возможность привязать RealPortfolios к TrackedPortfolio.
- **Управление заявками и сделками:** Просмотр списка TradeOrders. Возможность вручную изменить статус TradeOrder и создать связанную с ней Trade (с указанием цены исполнения).

Данный перечень является основой для MVP фронтенда. Детальный дизайн и UX/UI должны быть проработаны отдельно.

8. Открытые вопросы и дальнейшие уточнения

Данный раздел предназначен для фиксации вопросов, требующих дополнительного обсуждения и уточнений для дальнейшего проектирования и разработки системы.

1. **Детализация полей для "Другие идентификаторы"** в Instruments: Необходимо определить конкретный и исчерпывающий набор полей для альтернативных идентификаторов (например, OtherID_BBG_UniqueID, OtherID_Reuters_RIC, OtherID_CUSIP и т.д.). Какие из них являются наиболее приоритетными для MVP и последующих версий?
2. **Локализация и поддержка разных языков (Раздел 2):** Указано с вопросительным знаком. Требуется ли поддержка нескольких? Если да, то каких языков, и какие части системы должны быть локализованы (интерфейс, отчеты, документация)?
3. **Масштабируемость (Раздел 2, 7.2):** "Система должна выдерживать одновременную работу X пользователей". Необходимо определить целевое значение X для MVP и для будущих версий (например, через 1 год, 3 года). Каковы ожидания по росту числа портфелей, инструментов, сделок?

4. **Доступность системы (Раздел 7.3):** "99.0% (уточняется)". Требуется подтвердить или скорректировать этот показатель для MVP, учитывая использование Airtable и Vercel. Каковы процедуры восстановления в случае сбоев?
5. **AI_Advisor(s) и другие AI-агенты (Раздел 9.6, 11.2.3):** Каковы конкретные ожидания от AI-компонентов в первых версиях после MVP? Какие типы советов/аналитики должны предоставлять AI_Advisor(s) (например, рекомендации по ребалансировке, подходящие инструменты)? Какие конкретные риски должен отслеживать AI_RiskMonitor (например, превышение лимитов по концентрации, отклонение от стратегии)?
6. **"1 стратегия-бот" для PTS MVP (Раздел 8.1.3):** Какая именно стратегия должна быть реализована этим ботом для демонстрации? Должна ли она быть связана с одной из трех MVP стратегий (Buy & Hold, Algorithmic, Discretionary из раздела 9.14)? Каков ожидаемый результат работы этого бота?
7. **"Клиентские отчеты" для PTS MVP (Раздел 8.1.3):** Какие конкретно отчеты должны быть доступны клиентам в MVP? Какой уровень детализации и какие ключевые метрики должны в них присутствовать (например, отчет о составе портфеля, отчет о P&L за период, история операций)?
8. **Интеграция с API брокеров (Раздел 9.5):** "Позже тестирование Interactive Brokers и/или SingleBroker по API". Когда планируется это тестирование? Какие функции API будут приоритетными для первых этапов интеграции (например, получение данных о позициях, истории сделок, рыночных цен; или передача торговых ордеров)?
9. **"Черновые интерфейсы" для PTS MVP (Раздел 8.1.3):** Есть ли существующие макеты (wireframes), прототипы (mockups) или дизайн-концепции интерфейсов, на которые можно ориентироваться? Какие ключевые экраны и пользовательские сценарии должны быть реализованы в MVP фронтенде (помимо общих примеров из раздела 7.5 и нового раздела 12.4)?
10. **Регуляторные функции: комплайенс, репортинг (Раздел 8.1.2.X, 11.2.4):** Какие конкретные регуляторные требования (юрисдикция, тип отчетности, требования к аудиту) должны быть учтены в первую очередь, даже если их полная автоматизированная реализация выходит за рамки MVP?
11. **Источники данных для цен и корпоративных действий в MVP:** Откуда будут поступать данные о ценах и корпоративных

действиях для 10 MVP инструментов? Будет ли это ручной ввод, импорт из файлов, или тестовая интеграция с каким-либо бесплатным API?

12. **Детализация UI для "Ребалансировка отслеживаемого портфеля к модельному" (Раздел 9.10.3, 12.4):** Как должен выглядеть интерфейс для пользователя, где ему показывается предложенный список ордеров? Какие действия он может совершить с этим списком (например, утвердить все, отклонить все, утвердить выборочно)?
13. **Хранение истории status latest/hist для позиций (Раздел 9.9):** требуется явное дублирование записей со сменой статуса (Текущее описание предполагает явное дублирование).
14. И т.д.