

# **Техническое задание**

## **Аналитическая логика, функциональные и технические требования**

### **Аналитическая логика и алгоритмы отбора**

- Система реализует многоэтапную процедуру отбора инструментов, включающую последовательную обработку данных, расчет показателей и формирование итоговых рекомендаций.
- На этапе первичной обработки выполняется загрузка данных из подключенных источников, проверка их полноты и корректности, приведение форматов к единому виду, а также устранение дублирующих записей.
- Далее осуществляется базовая фильтрация. Из выборки исключаются инструменты с неполными данными, недостаточной ликвидностью, неподходящими сроками до погашения или несоответствием установленным ограничениям.
- После фильтрации система рассчитывает ключевые показатели, включая доходность к погашению, текущую доходность, дюрацию, спреда, волатильность и агрегированный риск-показатель.
- На следующем этапе применяется скоринговая модель. Каждому инструменту присваивается интегральная оценка с учетом доходности, кредитного качества, ликвидности и других факторов.
- По итогам скоринга система формирует статус инструмента: к покупке, к наблюдению или к исключению.
- Для инструментов, попавших в итоговый список, рассчитывается рекомендуемая цена входа с учетом рыночных параметров и заданных сценариев.

### **Функциональные требования**

- Система обеспечивает загрузку данных из различных источников и их централизованное хранение.
- Реализуется автоматическая обработка и обновление информации, а также применение пользовательских параметров отбора.
- Поддерживается расчет аналитических показателей и ранжирование инструментов.
- Результаты отображаются через пользовательский интерфейс с возможностью выгрузки в Excel, CSV и PDF.
- Пользователь может задавать ограничения по валютам, срокам, ликвидности, рейтингам и другим параметрам.
- Итоговый отчет включает характеристики инструмента, расчетные показатели, рекомендации и обоснование принятого решения.

### **Технические требования**

- Архитектура системы строится по микросервисному принципу с использованием REST API.
- Предусматривается разделение компонентов обработки данных, аналитики и пользовательского интерфейса.
- Backend реализуется на Python или Java, frontend — на современных веб-фреймворках.
- Используются реляционные и NoSQL базы данных для хранения структурированной и исторической информации.
- Система должна быть масштабируемой, отказоустойчивой и обеспечивать логирование всех операций.

### **Показатели эффективности (KPI)**

- Не менее 95% данных должны проходить автоматическую валидацию без критических ошибок.
- Полнота обязательных полей должна составлять не менее 98%.
- Расчет аналитики по выборке до 10 000 инструментов должен выполняться в пределах нескольких минут.
- Результаты отбора должны быть воспроизводимыми при повторном запуске.
- Доля инструментов, подтверждаемых аналитиком как корректно отобранные, должна составлять не менее 80%.
- Система должна обеспечивать высокую доступность и стабильную работу.

### **Этапы реализации**

- Проектирование и согласование логики отбора и архитектуры.
- Разработка и настройка контура данных.
- Создание аналитического ядра и скоринговой модели.
- Разработка пользовательского интерфейса.
- Тестирование и пилотная эксплуатация.
- Внедрение и подготовка документации.

### **Ожидаемый результат**

- Сокращение времени анализа инструментов и повышение его качества.
- Снижение зависимости от ручной обработки данных.
- Стандартизация подхода к инвестиционному отбору.
- Повышение прозрачности принимаемых решений.
- Создание технологической базы для дальнейшего развития системы.

### **Требования к исполнителю**

- Наличие опыта разработки аналитических или финтех-систем.
- Понимание принципов обработки финансовых данных.
- Способность предложить архитектуру и реализовать систему.

- Наличие подтвержденного опыта и портфолио.
- Готовность к взаимодействию с заказчиком и доработке решения.