

DOI 10.58351/2949-2041.2025.22.5.011

Майдан Алексей Эдгардович, студент
Московский авиационный институт,
Москва

ИССЛЕДОВАНИЕ СУЩЕСТВУЮЩИХ РЕШЕНИЙ ДЛЯ АНАЛИЗА ЦЕННЫХ БУМАГ И РАЗРАБОТКА СОБСТВЕННОГО СЕРВИСА

Аннотация: В статье рассматриваются существующие решения для анализа ценных бумаг, выявляются требования к новому сервису и описывается его проектирование и разработка.

Ключевые слова: Сервисы анализа ценных бумаг, разработка сервиса анализа ценных бумаг.

В настоящее время существует множество решений, предназначенных для учёта и анализа инвестиционного портфеля. Эти инструменты можно условно разделить на две категории: Telegram-боты и специализированные веб-сервисы. По результатам поиска были выделены наиболее популярные из них.

Среди Telegram ботов это:

Cherry Bot

BonkBot

Среди онлайн платформ это:

- PortfolioAnalyst
- Zephyr Portfolio Analytics
- Portfolio Visualizer

Был проведён сравнительный анализ данных сервисов. Результаты анализа приведены в таблице 1.

Таблица 1

Результаты сравнения сервисов аналитики ценных бумаг

Характеристики	Название платформы				
	Cherry AI	BonkBot	Portfolio Analyst	Zephyr Portfolio Analytics	Portfolio Visualizer
Платформа	Telegram бот	Telegram бот	Онлайн платформа	Онлайн платформа	Онлайн платформа
Поддержка традиционных активов	-	-	+	+	+
Поддержка криптовалют	+	+	-	-	-
Сложность	Низкая	Средняя	Средняя	Высокая	Высокая
Требование наличия счета у организации	-	-	+	-	-



Анализ существующих решений показывает, что ни одно из них не предоставляет комплексного подхода, объединяющего:

1. Поддержку традиционных финансовых инструментов (акции, облигации и другие активы Московской биржи).
2. Продвинутое аналитические возможности для оценки эффективности портфеля.
3. Удобство и доступность использования через популярный мессенджер Telegram.

Таким образом, разрабатываемый сервис призван заполнить существующую нишу между простыми ботами, ориентированными на криптовалюты, и сложными веб-сервисами, предоставляя пользователям удобный и функциональный инструмент для анализа инвестиционного портфеля, ориентированный на традиционные финансовые рынки.

Требования к сервису анализа ценных бумаг

Функциональные требования описывают действия, которые система должна выполнять. Для разрабатываемого сервиса анализа ценных бумаг эти требования включают:

1. Регистрация пользователей: Система должна автоматически идентифицировать новых пользователей Telegram и добавлять их в базу данных при первом взаимодействии с ботом.

Добавление ценных бумаг в портфель: Пользователь должен иметь возможность добавлять биржевые активы (по тикеру) в свой персональный портфель.

Удаление ценных бумаг из портфеля: Система должна позволять пользователю удалять ранее добавленные ценные бумаги.

Проверка существования ценной бумаги: Перед добавлением система должна валидировать тикер, проверяя его наличие на бирже (через API Московской биржи).

Получение текущего списка активов: Пользователь может запросить список активов, добавленных в его портфель, с отображением в виде текстового сообщения.

Расчёт коэффициента Шарпа: Система должна предоставлять пользователю аналитическую информацию по каждому активу в портфеле, включая значение коэффициента Шарпа на основе месячной и годовой доходности и текущей ключевой ставки [1].

Нефункциональные требования определяют свойства системы, которые не связаны напрямую с конкретной функциональностью, но оказывают влияние на архитектурные и технологические решения. Для рассматриваемой системы выделены следующие нефункциональные требования:

1. Независимость бизнес-логики от внешних источников данных: Система должна иметь возможность изолированной работы бизнес-логики, независимо от выбранных API или баз данных.

Модульность и расширяемость: Архитектура должна обеспечивать простоту внедрения новых источников данных, расчётных алгоритмов и каналов взаимодействия (например, веб-интерфейс или email-уведомления).

Высокая тестируемость: Компоненты бизнес-логики должны поддаваться модульному и интеграционному тестированию без необходимости запуска всей системы или обращения к внешним сервисам.

Устойчивость к ошибкам внешних сервисов: В случае недоступности API или базы данных система должна корректно обрабатывать исключения и уведомлять пользователя о сбое.

Минимальные требования к развёртыванию: Приложение должно быть развёрнуто с использованием контейнеризации (Docker), с возможностью локального и удалённого запуска без сложной предварительной настройки окружения.



Поддержка параллельной обработки запросов: Сервис должен быть способен обслуживать несколько пользователей одновременно без потери производительности и взаимных блокировок.

Конфиденциальность пользовательских данных: Система не должна передавать данные о пользователях третьим лицам. Вся информация о пользователе должна храниться исключительно локально и использоваться только в рамках предоставленного функционала.

Высокая производительность: Сервис должен быстро обрабатывать даже сложные запросы пользователей и способен выдерживать большие количества одновременных запросов.

Безопасность: сервис должен обеспечивать защиту конфиденциальности и целостности данных

Простота и удобство взаимодействия: Сервис должен быть легко доступен и удобен для взаимодействия.

Выбор языка программирования

Для реализации backend-логики был выбран язык Go, так как он обладает высокой производительностью, низким потреблением ресурсов и встроенной поддержкой многопоточности (через goroutines), что критически важно при обслуживании параллельных запросов от пользователей. Дополнительно, Go упрощает кросс-компиляцию и развёртывание, что способствует стабильной работе в контейнеризированной среде (Docker).

Выбор интерфейса для взаимодействия пользователя с сервисом

В качестве пользовательского интерфейса был выбран Telegram, поскольку:

1. Он обеспечивает широкое распространение и привычен пользователям [2];

Telegram API стабилен, хорошо документирован и предоставляет удобные механизмы взаимодействия (inline-кнопки, команды, callback-запросы);

Отсутствует необходимость разработки и поддержки собственного пользовательского интерфейса (веб или мобильного), что ускоряет время выхода продукта и снижает трудозатраты на UI/UX.

Проектирование архитектуры сервиса анализа ценных бумаг

Спроектирована архитектура платформы, с использованием стандартного шаблона проектирования backend сервисов, что обеспечивает высокий уровень организации и совместимости с инструментами Go-разработки, что:

- способствует читаемости и предсказуемости кода;
- упрощает подключение новых разработчиков;
- позволяет масштабировать проект по мере его роста;
- обеспечивает отделение инфраструктурных и исполняемых компонентов от бизнес-логики.

В сочетании с архитектурными принципами (в частности, Clean Architecture), подобная структура формирует надёжную основу для создания промышленного backend-сервиса, устойчивого к изменениям и удобного в сопровождении [3].

Итоговый результат сервиса анализа ценных бумаг представлен на рисунке 1.



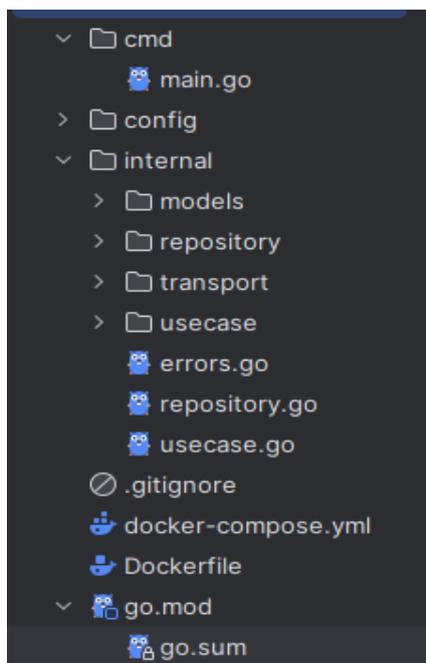


Рис. 1. Архитектура сервиса анализа ценных бумаг

Алгоритмы взаимодействия пользователя с портфелем

Когда пользователь запускает бота (вводит команду /start), сервис регистрирует его в базе данных, если ранее пользователь ещё не взаимодействовал с системой. После успешной регистрации пользователю отображается главное меню с текущими активами и возможными действиями.

Основная логика взаимодействия происходит через кнопки меню и ответы на сообщения бота. Пользователь может выбрать действие: добавить актив, удалить актив или рассчитать коэффициент Шарпа. При выборе добавления актива бот запрашивает у пользователя тикер – пользователь должен отправить сообщение в ответ на запрос. Бот проверяет, существует ли такой актив на Московской бирже и есть ли он уже в портфеле пользователя. В случае успеха актив добавляется в базу данных, и пользователь получает подтверждение. На рисунке 2 представлена блок-схема алгоритма добавления актива.

Удаление актива работает аналогичным образом: пользователь выбирает соответствующую команду, вводит тикер в ответ на сообщение бота, и система проверяет наличие данного актива в базе. При подтверждении актив удаляется, и пользователь получает уведомление об успешном удалении. Все ошибки, такие как попытка добавить несуществующий тикер или удалить отсутствующий в портфеле актив, обрабатываются и сопровождаются понятными сообщениями.

Для расчёта коэффициента Шарпа бот запрашивает у внутреннего сервиса данные о доходности всех активов пользователя за месяц и год, а также актуальную ключевую ставку. После получения данных производится расчёт коэффициента, и результат возвращается пользователю в виде таблицы. Это позволяет инвестору оценить риск и эффективность своих вложений [4]. На рисунке 3 представлен пример расчета коэффициента Шарпа для портфеля пользователя.

Таким образом, всё взаимодействие с пользователем построено вокруг чётких сценариев и реализовано в стиле “вопрос-ответ”, что делает работу с ботом простой даже для неподготовленного пользователя.



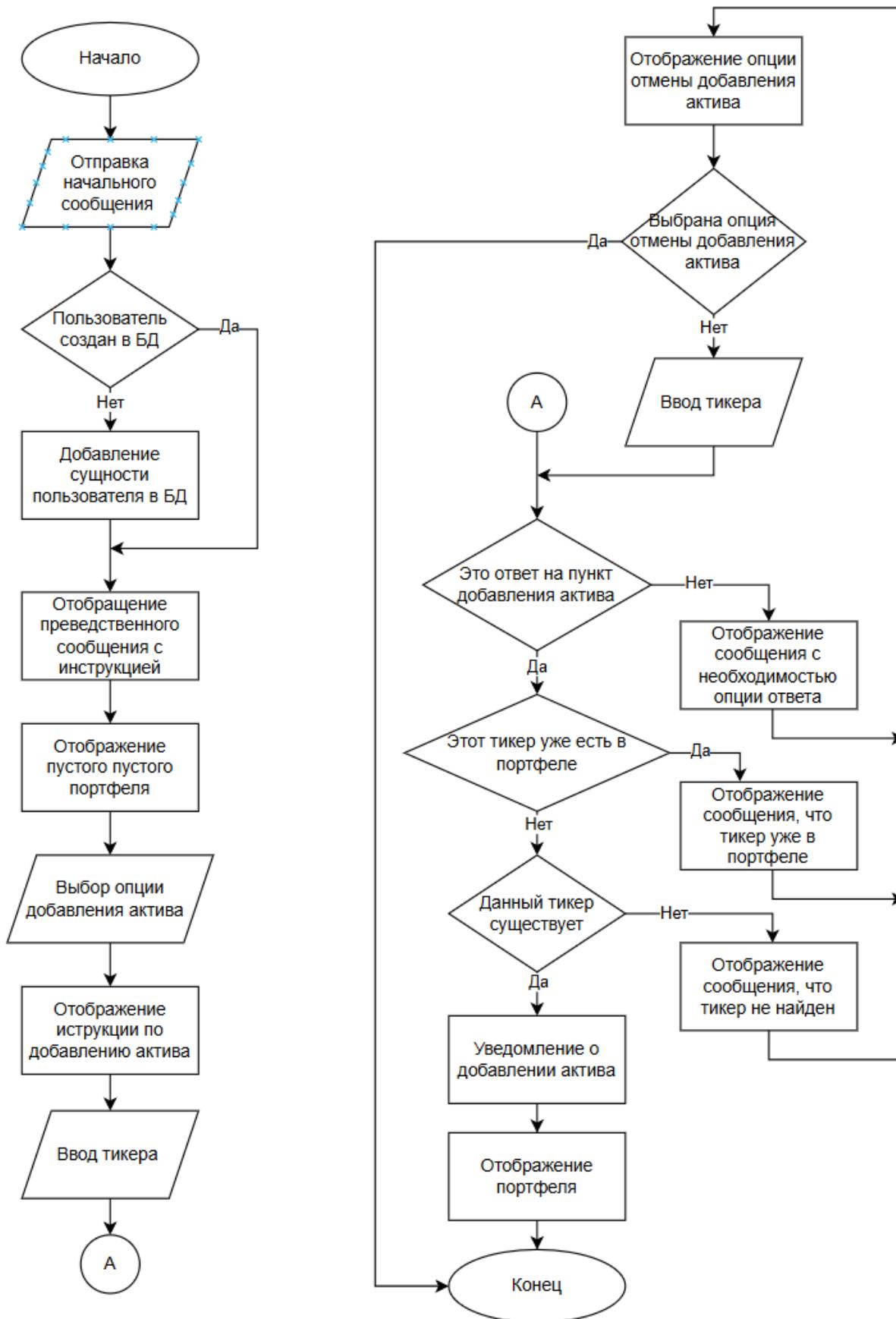


Рис. 2. Блок-схема алгоритма добавления актива



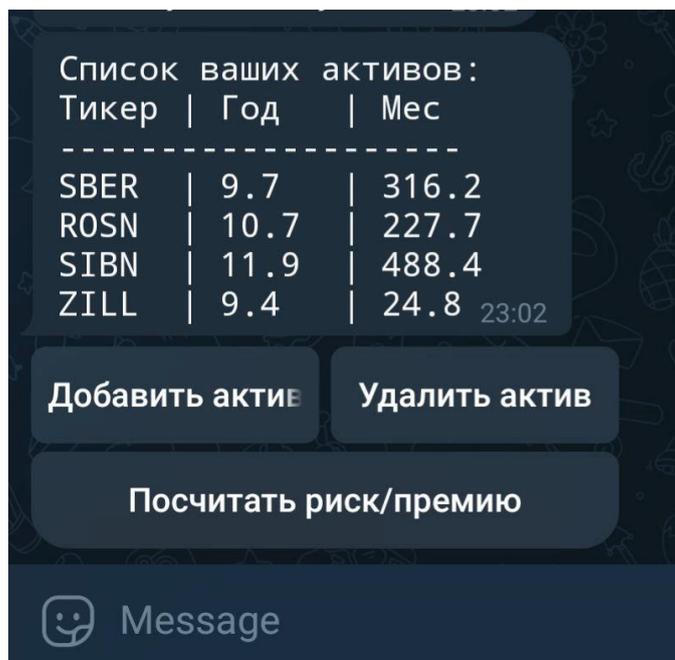


Рис. 3. Пример расчета коэффициента Шарпа

Аналитические возможности сервиса

Разработанная система предоставляет пользователю ряд аналитических инструментов, способствующих принятию обоснованных инвестиционных решений. Эти инструменты позволяют оценивать текущую эффективность активов, выявлять риски и корректировать стратегию в соответствии с рыночной ситуацией.

К аналитическим возможностям системы относятся:

1. Автоматическое получение данных о ключевой ставке Центробанка РФ, которая используется в расчётах в качестве безрисковой доходности. Это исключает необходимость ручного ввода и делает анализ актуальным и адаптированным к текущим экономическим условиям;

Табличное представление результатов анализа, выводимое непосредственно в интерфейсе Telegram, позволяет быстро сопоставить значения коэффициента Шарпа для всех активов портфеля. Благодаря форматированию с фиксированной шириной шрифта обеспечивается высокая читаемость и наглядность данных

Обработка исключений при отсутствии данных – в случае, если невозможно рассчитать показатель (например, из-за отсутствия торгов по активу или нулевой волатильности), система уведомляет пользователя и сохраняет корректность интерфейса, предотвращая искажение восприятия информации

Возможность оперативного пересчёта показателей с учётом новых данных – пользователь может в любой момент инициировать повторный расчёт, чтобы получить обновлённую картину состояния портфеля

Наличие подобных аналитических функций превращает систему в полноценный помощник в сфере личных финансов. Пользователь получает не просто список активов, но и возможность глубоко проанализировать каждый из них с точки зрения эффективности, а также своевременно принимать решения об изменении состава портфеля. Это особенно важно в условиях высокой волатильности рынка и необходимости быстрой адаптации инвестиционной стратегии.



Список литературы:

1. Инвестиции / У. Ф. Шарп, Г. Дж. Александер, Д. Дж. Бэйли; пер. с англ. – М.: Инфра-М – 2001;
2. Bot Interfaces in Messaging Apps: A Systematic Review / D. Morrison, J. Smith, L. Wang / International Journal of Human–Computer Interaction. – 2021.
3. Clean architecture: a craftsman's guide to software structure and design / R. C. Martin. / Prentice Hall – 2017;
4. The Cross-Section of Expected Stock Returns / Е. Ф. Фама, К. Р. Френч / Journal of Finance. – 1992;

