

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
 Департамент программной инженерии

СОГЛАСОВАНО

Доцент департамента
 программной инженерии
 факультета компьютерных наук

канд. техн. наук

 Н. С. Белова

12 мая 2022 г.

УТВЕРЖДАЮ

Академический руководитель
 образовательной программы
 «Программная инженерия»

Профессор департамента программной
 инженерии, канд.техн.наук

_____ В.В.Шилов

12 мая 2022 г.

Подп. и дата	
Инв. №	
Взам. инв.	
Подп. и дата	
Инв. № подл	

БИРЖЕВОЙ БОТ

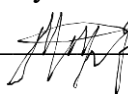
Программа и методика испытаний

ЛИСТ УТВЕРЖДЕНИЯ


RU.17701729.05.01-01 51 01-1-ЛУ

Исполнители:

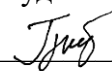
Студент группы БПИ208

 /Армаш В.В. /
 12 мая 2022 г

Студент группы БПИ201

 - /Ищенко А.С. /
 12 мая 2022 г

Студент группы БПИ208

 /Глинник А.М. /
 12 мая 2022 г

Москва 2022

УТВЕРЖДЕН

RU.17701729.05.01 -01 51 01-1-ЛУ

БИРЖЕВОЙ БОТ

Программа и методика испытаний

RU.17701729.05.01 -01 51 01-1

Листов 31

<i>Инва. № подл</i>	
<i>Подп. и дата</i>	
<i>Взам. инв.</i>	
<i>Инва. №</i>	
<i>Подп. и дата</i>	
<i>Подп. и дата</i>	

СОДЕРЖАНИЕ

ТЕРМИНОЛОГИЯ	4
1. ОБЪЕКТ ИСПЫТАНИЙ	7
1.1. Наименование программы	7
1.2. Краткая характеристика и область применения.....	7
2. ЦЕЛЬ ИСПЫТАНИЙ	8
3. ТРЕБОВАНИЯ К ПРОГРАММЕ.....	9
3.1. Требования к функциональным характеристикам	9
3.1.1. Требования к составу выполняемых функций программы	9
3.1.1.1. Функциональные требования к API для получения информации о ценах акций.....	9
3.1.1.2. Функциональные требования к API-модулю, эмулирующему работу торгового API	10
3.1.1.3. Функциональные требования к модели для прогнозирования цен на акции.....	10
3.1.1.4. Функциональные требования к модулю принятия решений:	11
3.1.1.5. Функциональные требования к основному модулю:	12
3.1.1.6. Функциональные требования к интерфейсному модулю:	12
3.1.2. Организация хранения данных.....	13
3.1.3. Организация входных данных.....	13
3.1.4. Организация выходных данных	13
3.2. Требования к временным характеристикам	13
3.3. Требования к интерфейсу	14
3.4. Требования к надежности	14
3.4.1. Требования к обеспечению надежного (устойчивого) функционирования программы	14
3.4.2. Требования к корректности данных	14
3.4.3. Отказы из-за некорректных действий оператора.....	14
4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ	15
4.1. Предварительный состав программной документации	15
4.2. Специальные требования к программной документации	15
5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ.....	16
5.1. Технические средства, используемые во время испытаний	16
5.2. Программные средства, используемые во время испытаний	16

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

5.3. Порядок проведения испытаний.....	16
6. МЕТОДЫ ИСПЫТАНИЙ.....	17
6.1. Испытания требований к программной документации	17
6.2. Испытания требований к функциональным характеристикам	17
6.2.1. Испытания требований к API для получения информации о ценах акций	17
6.2.2. Испытания требований к API-модулю, эмулирующему работу торгового API.....	19
6.2.3. Испытания требований к модели для прогнозирования цен на акции	19
6.2.4. Испытания требований к модулю принятия решений	21
6.2.5. Испытания требований к основному модулю.....	22
6.2.6. Испытания к интерфейсному модулю	22
6.3. Испытания требований к временным характеристикам	23
6.4. Испытаний требований к надежности.....	23
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	24
СРАВНИТЕЛЬНАЯ ТАБЛИЦА АНАЛОГОВ ПРОЕКТА «БИРЖЕВОЙ БОТ».....	27
Развернутый список требований к API для получения информации о ценах акций.....	28
СРАВНЕНИЕ СУЩЕСТВУЮЩИХ API И ОБОСНОВАНИЕ ВЫБОРА	29
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	31

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

ТЕРМИНОЛОГИЯ

- 1) API-модуль - программный интерфейс, посредством которого можно обратиться к данным. В данном случае— модуль, отвечающий за взаимодействие с API торговой площадки (биржи), осуществляющий периодические запросы к нему с целью получения актуальной информации о состоянии интересующих позиций.
- 2) Агрегация запроса, предагрегация данных — процесс осуществления действий с данными, приводящий их к состоянию, необходимому для работы программы. Пример: перевод времени из человекочитаемого формата в миллисекунды с момента ЕРОСН для корректности сравнения с другими временами и датами внутри программы.
- 3) Актор (здесь и далее в рамках проекта) - автономный асинхронный модуль, способный принимать сообщения и отвечать на них. Обладает базовым перечнем доступов и функций. Подробнее про работу с акторами можно прочитать в документации Thespian^[13] [14].
- 4) Акция (share, stock) - вид ценных бумаг.
- 5) Визуализация — представление информации пользователю в читаемом виде, это может быть как выводимое на экран значение, так и график.
- 6) Виртуальная транзакция — транзакция, не производящая действительных действий на бирже, но меняющая состояние виртуального счёта и баланса акций системы, и т.д.
- 7) Виртуальный (здесь и далее в рамках проекта) — имитирующий работу с подлинными сущностями, действующий в рамках тех же правил и ограничений, имеющихся с настоящими объектами. То есть, виртуальный счёт — счёт в широком понимании этого слова, но никак не привязанный к живым деньгам пользователя и имитирующий работу одного.
- 8) Восстановление после сбоя — автоматическое возобновление работы программы в случае критической ошибки, предполагает полное соответствие состояния работы программы и значений её данных оным до сбоя, т.е. отсутствие потери данных и т.п.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

- 9) Измерение эффективности модуля: среднее время обработки одной акции (время от получения запроса до выдачи результата)
- 10) Лог — краткий отчёт унифицированного вида о действиях и состоянии системы в данный момент времени.
- 11) Логгер — система непрерывного автоматического создания логов.
- 12) Модель акторов — модель параллельных вычислений, строящаяся вокруг понятия актора (англ. actor «актёр; действующий субъект»), считающегося универсальным примитивом параллельного исполнения. Актор в данной модели взаимодействует путём обмена сообщениями с другими акторами, и каждый в ответ на получаемые сообщения может принимать локальные решения, создавать новые акторы, посылать свои сообщения, устанавливать, как следует реагировать на последующие сообщения.
- 13) Модуль — независимая единица исполнения внутри программы (отдельный поток/процесс), действующий в соответствии с определённым стандартом входных и выходных данных, с возможностью полноценного, в том числе нагрузочного тестирования в отрыве от остальной программы, взаимозаменяемый с любыми другими модулями, действующими в рамках аналогичных стандартов ввода/вывода.
- 14) Наблюдаемые позиции (акции) - перечень позиций (акций), добавленных пользователем для наблюдения/отслеживания и совершения программой операций купли/продажи.
- 15) Позиция (также встречается под обозначениями „акция”, „метрика”) — интересующая нас единица торговли, в отношении которой проводится работа системы в том или ином виде. В зависимости от контекста может подразумевать как наименование позиции, так и конкретную единицу с таким названием (одну акцию, например), а также временной ряд с информацией о динамике изменения параметров данной позиции (количество лотов, текущая цена, и т.п.)
- 16) Полный перечень акций – перечень акций, получаемый с помощью выбранных API
- 17) Свободные денежные средства - те, которые остаются у владельца после уплаты всех текущих обязательств.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

- 18) Тик (англ. tick) — единица времени.
- 19) Токен (в контексте данного проекта) – набор символов, в котором зашифрована информация о владельце, правах доступ и прочей необходимой для авторизации в TINKOFF INVEST API ^[11] информации.
- 20) Торговый API — предоставляемый биржей API для осуществления операций торговли на ней, а также получения актуальной информации о статусе имеющихся позиций на ней.
- 21) Транзакция — процедура разового обмена информацией/деньгами с обратной связью, подтверждающей успешное её проведение. В рамках проекта мы рассматриваем транзакции, удовлетворяющие требованиям ACID.
- 22) Трейдинг - совершение торговых операций на биржевом рынке с целью получения прибыли.
- 23) Фондовая биржа - площадка для торговли ценными бумагами.
- 24) FIGI, figi, фиджи (англ.: Financial Instrument Global Identifier) — глобальный идентификатор финансового инструмента.
- 25) Open-source – проекты с открытым исходным кодом. Разрабатываются преимущественно на некоммерческих началах. Часто проекты публикуются именно так, чтобы появилась возможность улучшить качество продукта посредством его доработки и изучения совместными усилиями.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

1. ОБЪЕКТ ИСПЫТАНИЙ

1.1. Наименование программы

Наименование программы – «Биржевой бот».

Наименование программы на английском языке – «Stock bot».

1.2. Краткая характеристика и область применения

«Биржевой бот» - автоматизированная распределённая (модульная, многопоточная) система на базе машинного обучения для (виртуальной) торговли на бирже и составления кратковременных прогнозов поведения позиций.

Принято считать, что для получения прибыли на бирже нужно много знаний, интуиция опытного трейдера или проприетарные биржевые роботы с многомиллионными бюджетами разработки.

Проект призван развеять это предубеждение и показать, что даже самостоятельно разработанные автономные системы могут генерировать приятную прибыль при намного меньшем пороге входа.

Программа ориентирована на рынок ценных бумаг. «Биржевой бот» предоставляет пользователям возможность самостоятельно задавать список наблюдаемых акций для торговли, видеть графики состояния счёта и показатели прибыли, добавлять и изымать деньги с виртуального счета.

Целевой аудиторией проекта являются люди, которые непосредственно связаны с инвестициями – проект потенциально может развиваться в «боевого» бота, позволяющего людям автоматизировать торговлю на рынках ценных бумаг, криптовалют и т.п. логируемых позиций.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

2. ЦЕЛЬ ИСПЫТАНИЙ

Целью испытаний является проверка корректности выполнения программой функций, перечисленных в разделе «Требования к программе».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

3. ТРЕБОВАНИЯ К ПРОГРАММЕ

Программа должна соответствовать следующим требованиям, указанным в документе «“Биржевой бот”». Техническое задание».

3.1. Требования к функциональным характеристикам

3.1.1. Требования к составу выполняемых функций программы

Программа состоит из 2 частей (6 модулей).

Общие функциональные требования к программе:

- 1) Производство виртуальной торговли для выбранных акций
- 2) Добавление и удаление наблюдаемых акций для торговли ими
- 3) Добавление и изъятие денег с виртуального счёта
- 4) Предоставление информации о текущем размере средств (общее количество, вложенные и свободные) – отображение в интерфейсе программы
- 5) Предоставление информации о прибыли за выбранный период времени – отображение в интерфейсе программы
- 6) Визуализация состояния счета в реальном времени - график, обновляющийся каждую секунду, а также отображение текущего
- 7) Визуализация текущей работы системы и графика цен/прогноза цен в реальном времени
- 8) Эффективность при работе с большим количеством наблюдаемых позиций (не более трёх секунд на получение и обработку текущих цен, построение прогнозов и принятие решения без учёта времени ответа сервера для получения цен)

Core-часть: API для получения информации о ценах акций; API-модуль, эмулирующий работу торгового API; Модель для прогнозирования цен на акции в будущем; Модуль принятия решений; Основной модуль.

UI-часть: Интерфейсный модуль.

3.1.1.1. Функциональные требования к API для получения информации о ценах акций

- 1) Налаженное взаимодействие с выбранным API. Выбор пал на официальный клиент Tinkoff Invest^[12]. Мотивация выбора описана в приложении 4.
- 2) Обработка и предоставление полученной информации о позициях(акциях) в установленных форматах (подробнее о форматах и запрашиваемой информации в развернутом списке требований, предоставленном в приложении 3)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

- 3) Эффективность при работе с большим количеством наблюдаемых акций (агрегация ответа сервера не более секунды)
- 4) Передача информации о наличии неполадок на стороне сервиса, предоставляющего информацию о ценах (передача сообщений основному модулю о наличии неполадок)
- 5) Проверка входных данных об акциях, поступающих от основного модуля, на корректность и существование (т. е. решение ситуаций, когда поступает запрос на несуществующие позиции\запрос на продажу\покупку невозможного количества акций - считаем невозможными те данные, которые не кратны лоту или <0).
- 6) Восстановление после сбоя (см.п. «Требования к надежности»)

3.1.1.2. Функциональные требования к API-модулю, эмулирующему работу торгового API

В этом модуле логируется запись транзакций.

- 1) Совершение виртуальных операций купли/продажи на бирже в соответствии с принятым основным модулем решением (в рамках проекта совершение виртуальной операции мы осуществляем через логирование транзакций)
- 2) Предоставление информации о совершенных операциях в формате [время- позиция - количество единиц - тип операции(купля/продажа) - цена за единицу]
- 3) Запись истории транзакций в файл [статус транзакции: success\failure, информация об операции – см.п.2 текущего списка].
- 4) Обработка некорректных\неактуальных данных – передача оповещения основному модулю, продолжение работы программы (за счет игнорирования данных, признанных некорректными). Статус транзакции логируется как failure.

3.1.1.3. Функциональные требования к модели для прогнозирования цен на акции

- 1) Построение прогнозов цен для наблюдаемых позиций в соответствии с последними полученными данными о текущих ценах
- 2) Оценка точности предоставляемых прогнозов
- 3) Хранение модели прогнозирования для каждой наблюдаемой позиции
- 4) Предоставление основному модулю программы прогнозов и оценки их точности (в виде [<название>, <цена>, <timestamp>, <среднеквадратичное отклонения для данной позиции>])
- 5) Эффективность (не более секунды на построение прогноза по позиции, не более двух часов на обучение модели)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

3.1.1.4. Функциональные требования к модулю принятия решений:

- 1) Поиск и интеграция актор-системы с открытым исходным кодом и свободной лицензией, либо написание собственной в соответствии с установленными критериями (асинхронность, скорость, кроссплатформенность, устойчивость к падениям отдельных акторов и автоматический их перезапуск, возможность низкоуровневого логирования сообщений, возможность развёртывания на нескольких устройствах с сопутствующим общением модулей посредством встроенного либо внешнего http/grpc-клиента)
- 2) Поиск и интеграция тактового генератора и механизма реализации отложенных событий с открытым исходным кодом и свободной лицензией, либо написание собственного в соответствии с установленными критериями (асинхронность, нетребовательность в ресурсах процессора, обеспечение точности тактов/реализации событий с погрешностью не более секунды)
- 3) Выбор формата и способа взаимодействия модулей программы между собой.

В первую очередь необходимо реализовать базовый актор, бесшовно интегрирующийся с имеющейся актор-системой (об этом ниже, в требованиях к «основному модулю»), создать формат сообщений и наладить взаимодействие данного модуля с другими модулями настоящей программы.

Основные модели поведения модуля (неразрывны с другими модулями программы):

- 1) Получение прогнозов от модуля прогнозирования в определённом формате (временной ряд вида «время — предполагаемая цена») с последующей оценкой рисков и вероятностей ошибок (линейная регрессия вероятности по времени, квадратичная по цене)
- 2) Построение торговой стратегии по текущей позиции в соответствии с полученными прогнозами и рисками, а также комиссией по итогам п.1
- 3) В соответствии со стратегией из п.3 создание отложенных событий и отправка оных в соответствующую часть основного модуля, посредством которых осуществляется непосредственное управление торговым модулем (эмулятором торгового API) и следствием исполнения которых является совершение операций купли-продажи.
- 4) (Опционально) в зависимости от задержки исполнения модуля прогнозирования прямой запрос к модулю API о текущем состоянии наблюдаемой позиции с целью исключения действий по устаревшей стратегии и прогнозам.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

Обязательна проверка поступающих запросов на корректность, т.е. на актуальность и соответствие детерминированному формату сообщений, а также автоматический перезапуск модуля в случае его падения по каким-либо обстоятельствам (осуществляется актор-системой)

3.1.1.5 Функциональные требования к основному модулю:

Основной модуль должен содержать в себе актор-систему и осуществлять управление оной. Для данного проекта выбрана актор-система Thespian^{[13][14]}, поскольку та отвечает всем обязательным требованиям и является удобной в использовании. Дополнительно необходимо наличие тактового генератора и механизма реализации отложенных событий. Было принято решение сделать оные на базе минималистичной библиотеки schedule, так как её функционала достаточно для данного проекта.

Базовые механизмы и действия данного модуля (неразрывны с другими модулями программы):

- 1) Получение запроса от интерфейсного модуля на добавление/удаление наблюдаемых акций
- 2) В соответствии с п.1 создание и инициализация цепочки исполнения для каждой конкретной акции, состоящей из модуля прогнозирования и модуля принятия решений
- 3) Исполнение отложенных действий, т.е. отправка отложенных сообщений в необходимый момент времени тем акторам, которым они предназначались изначально
- 4) Осуществление передачи сообщений между модулями посредством актор-системы
- 5) Создание, управление, выключение и удаление акторов посредством актор-системы
- 6) Содержание информации о балансе и наблюдаемых позициях, а также предоставление данной информации по запросу от интерфейсного модуля, а также периодическая актуализация информации о балансе средств посредством модуля API
- 7) Исполнение всех операций, затребованных интерфейсным модулем, либо переадресация их ответственным за это модулям.

3.1.1.6. Функциональные требования к интерфейсному модулю:

- 1) Наличие пользовательского интерфейса, см. «приложение 1»
- 2) Предоставление пользователю возможности добавления и удаления наблюдаемых позиций (кнопки с окном для ввода названий)
- 3) Предоставление пользователю возможности добавления и изъятия средств с виртуального счёта (кнопки с окном для ввода суммы)
- 4) Отображение текущего количества средств на счёте

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

- 5) Отображение информации о прибыли (по умолчанию в процентах за последнюю неделю, пользователь может задать свой период)
- 6) Визуализация состояния счета (график количества средств на счёте за неделю и за день, пользователь может задать свои периоды)
- 7) Вывод на экран всех сообщений об ошибках, получаемых от основного модуля
- 8) Эффективность при работе с большим количеством наблюдаемых позиций (все данные, выводящиеся на экран, должны обновляться не дольше секунды)
- 9) Устойчивость к некорректным входным данным (например, при вводе некорректной суммы для добавления или изъятия средств) и вывод соответствующего сообщения на экран
- 10) Осуществление записи текущего состояния в файл и его последующее восстановления в случае сбоя

3.1.2. Организация хранения данных

Хранение данных каждый из перечисленных в п.4.1.1. модулей обеспечивает самостоятельно.

Организация хранения данных для каждого модуля описана в технических заданиях, касающихся конкретных модулей.

3.1.3. Организация входных данных

Формат организации входных данных определяется каждым модулем самостоятельно и описывается соответственно в пункте «Организация входных данных» в отдельных технических заданиях. Предварительный формат входных данных кратко описан для каждого модуля в п.4.1.1.

3.1.4. Организация выходных данных

Выходные данные системы описаны в пункте 4.1.1.6. («Функциональные требования к интерфейсному модулю»).

При условии, что входные данные были некорректными выходными данными будет являться сообщение об ошибке. Корректность данных, формат организации выходных данных определяется каждым модулем самостоятельно и описывается соответственно в пунктах «Требования к корректности данных», «Организация выходных данных» в отдельных технических заданиях.

3.2. Требования к временным характеристикам

Скорость работы системы зависит от скорости интернет-подключения и скорости работы алгоритма. Скорость работы алгоритма зависит от объема поступающих данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

Эффективной работа системы будет считаться, если агрегация ответа сервера длится не более секунды.

3.3. Требования к интерфейсу

Интерфейс программы должен быть реализован на русском языке.

Требования к интерфейсному модулю описаны в пункте «Функциональные требования к интерфейсному модулю».

Все исключительные ситуации, которые могут возникнуть в программе, должны быть обработаны так, чтобы было интуитивно понятно, в чем заключалась проблема.

3.4. Требования к надежности

3.4.1. Требования к обеспечению надежного (устойчивого) функционирования программы

При любых действиях пользователя и при любых входных данных (т. е. при условии работоспособности устройства и отсутствия попыток со стороны пользователя и/или ОС принудительно завершить программный процесс), система не должна завершаться из-за ошибки. Все исключения, возникающие в процессе работы программы, должны быть обработаны и не должны прерывать процесс.

3.4.2. Требования к корректности данных

Программа должна обеспечивать проверку корректности входных данных.

Корректность входных данных определяется для каждого модуля отдельно, она описывается в соответствующем техническом задании и в п. («Организация входных данных»)

3.4.3. Отказы из-за некорректных действий оператора

Отказ может возникнуть при условии повреждения исходного кода и/или при условии попытки принудительно завершить программный процесс.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

4.1. Предварительный состав программной документации

- 1) «Биржевой бот». Техническое задание (ГОСТ 19.201-78[2]);
- 2) «Биржевой бот». Программа и методика испытаний (ГОСТ 19.301-78[3]);
- 3) «Биржевой бот». Текст программы (ГОСТ 19.401-78[4]);
- 4) «Биржевой бот». Пояснительная записка (ГОСТ 19.404-79[5]);
- 5) «Биржевой бот». Руководство оператора (ГОСТ 19.505-79[6]);

4.2. Специальные требования к программной документации

- 1) Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78[1] и ГОСТами к каждому виду документа (см. п. 5.1.);
- 2) Пояснительная записка должна быть загружена в систему Антиплагиат через Smart LMS «НИУ ВШЭ». Лист, подтверждающий загрузку пояснительной записки, сдается в учебный офис вместе со всеми материалами не позже, чем за день до защиты курсовой работы;
- 3) Техническое задание и пояснительная записка, титульные листы других документов должны быть подписаны руководителем разработки и исполнителем Документация и программа сдается в электронном виде в формате .pdf или .docx. в архиве формата .zip или .rar;
- 4) Все документы перед защитой курсовой работы должны быть загружены в информационно-образовательную среду НИУ ВШЭ систему Smart LMS (Smart Learning Management System), в личном кабинете дисциплина - «Курсовой проект, 2 курс ПИ», одним архивом.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

5.1. Технические средства, используемые во время испытаний

Запуск и стабильная работа должна быть обеспечены при:

- 1) Unix-система
- 2) 8+ ядер/16+ потоков
- 3) 8+ Гб ОЗУ
- 4) 4+ Гб ПЗУ
- 5) Доступ к интернету.

5.2. Программные средства, используемые во время испытаний

Для функционирования системы требуются следующие программные совместимости:

- 1) Unix-система
- 2) Предустановленный Python версии 3.9 или выше
- 3) Доступ к жесткому диску
- 4) Доступ к интернету.

Тестирование проводилось на MacBook Pro (13-inch, M1, 2020).

5.3. Порядок проведения испытаний

Испытания должны проводиться в следующем порядке:

- 1) Проверка требований к программной документации;
- 2) Проверка требований к функциональным характеристикам;
- 3) Проверка требований к временным характеристикам;
- 4) Проверка требований к надежности.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

6. МЕТОДЫ ИСПЫТАНИЙ

Испытания представляют собой процесс установки соответствия программы и программной документации заданным требованиям.

Программа «Биржевой бот» поставляется в виде архива. Кроме необходимых для работы приложения файлов, в том числе исполняемых, архив содержит все необходимые документы.

6.1. Испытания требований к программной документации

Состав программной документации проверяется визуально, проверяется наличие полного комплекта документов в системе Smart LMS. Также, документация проверяется визуально на соответствие нормам ГОСТ. Все документы удовлетворяют представленным требованиям.

6.2. Испытания требований к функциональным характеристикам

6.2.1. Испытания требований к API для получения информации о ценах акций

1. Испытание работы токена Tinkoff API:

Когда указан существующий токен (сгенерированный в личном кабинете Tinkoff Invest), сервис предоставляет все запрашиваемые данные.

Когда токен указан неверно, обрабатывается и логируется ошибка на стороне сервиса:

```
{ } 20220512.json > ...
1
2   {
3     "status": "failure",
4     "share": "Seagen Inc",
5     "message": "Authentication failed"
6   },
7   {
```

Рисунок 1 – Ошибка авторизации

2. При условии отсутствия ошибок на стороне сервиса предоставляется полный список акций, точно так же можно получить информацию о конкретной акции:

```
{'name': 'MicroStrategy Inc', 'price': 0.001, 'lot': 1, 'currency': 'usd'}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

Рисунок 2 – Предоставление информации о запрашиваемой акции

Надо заметить, что на рисунке предоставлен не лог, поскольку программа логирует именно транзакции(или попытки их совершить).

Ниже изображен лог ситуации, когда без подключения к интернету попытались запросить информацию об акции и произвести транзакцию с ее участием:

```
{
  "status": "failure",
  "share": "Seagen Inc",
  "message": "DNS resolution failed for invest-public-api.tinkoff.ru: UNKNOWN: OS Error"
}
```

Рисунок 3 – Ошибка подключения. Причина: отсутствие подключения к интернету

3.Испытание запроса на совершение операции – покупка конкретной акции, количество 1, цена 0.01.

Модуль-эмулятор сделал запись о совершенной транзакции – она успешна, поскольку указана существующая акция и допустимое количество.

```
{
  "status": "success",
  "time": "17-44-55",
  "share": "MicroStrategy Inc",
  "operation": "buy",
  "count": 1,
  "price": 0.001,
  "currency": "usd"
}
```

Рисунок 4 – Запись об успешной покупке акции

5.Если ошибка случилась на стороне пользователя\программиста – запрошена несуществующая акция - запишется соответствующий лог:

```
{
  "status": "failure",
  "share": "Seagen Inc",
  "message": "No such share name"
},
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

Рисунок 5 – Ошибка ввода: некорректное название акции

Основываясь на данных наблюдениях, можно сделать вывод, что требования к функциональным характеристикам полностью удовлетворены (однако возможна последующая балансировка).

6.2.2. Испытания требований к API-модулю, эмулирующему работу торгового API

Испытания модуля-эмулятора, отвечающего за логирование транзакций, произведены успешно – записываются как успешные, так и не удачные транзакции, файл с датой создается верно. Выше представлены различные испытания API- модуля и положительная реакция эмулятора (запись принятия решений, запись ошибок на стороне пользователя\программиста и сервиса, запись статуса транзакций, запись транзакций за один день в один файл, генерация файла) подтверждает его успешное прохождение испытаний требований к выходным данным.

6.2.3. Испытания требований к модели для прогнозирования цен на акции

Чтобы результаты испытаний были нагляднее, следует уточнить структуру модуля:

Класс **Model** имеет 2 пользовательских метода: `learn` и `predict`.

(они оба принимают на вход свечи - массивы элементов вида `{time: datetime, price : float}` с временным шагом в `model_step`)

1. `learn` ничего не возвращает в случае успеха и в случае недостаточного количества данных для обучения вызывает исключение `ValueError('not enough data')`.

Также вычисляет и сохраняет в поле `MSE` массив среднеквадратичных отклонений в зависимости от дальности прогноза.

2. `predict` в случае успеха возвращает массив элементов вида `{time : datetime, price : float, MSE : float}` - массив прогноза, где `MSE` для каждой точки свой. Временные точки прогноза всегда отличаются на `model_step`, первая получается прибавлением `model_step` к последней точке из входных данных.

Вызывает исключение `ValueError` в случае недостаточного количества данных для прогноза и возвращает `None`, если модель ещё не обучена (имеет приоритет над `ValueError`)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

model_step в реализуемой версии равен 1 минуте.

3.enum-класс ModelActorResponseStatus со значениями

SUCCESS = 0,

NOT_ENOUGH_DATA = 1,

MODEL_IS_NOT_READY = 2,

UNDEFINED_ERROR = 3 для ответа на запросы.

4.enum-класс ModelActorMessageTarget имеет поля

UNSPECIFIED = 0,

LEARN = 1,

PREDICT = 2 для спецификации запроса

5.Класс ModelActorMessage имеет поля

message_type : ActorMessageType,

message_target: ModelActorMessageTarget (для ответа совпадает с target соответствующего запроса),

id (id запроса/ответа, id ответа совпадает с id соответствующего запроса),

args (аргументы запроса/ответа).

6.Класс ModelActor отнаследован от Actor, при создании создаёт объект класса Model и хранит в своём поле model. Имеет переопределённую функцию recieveMessage(msg :

ModelActorMessageTarget, sender) для принятия и обработки запроса.

В зависимости от message_target запроса вызывает соответствующий метод объекта model с аргументами args и возвращает:

1. для LEARN объект ModelActorResponseStatus (SUCCESS, если модель успешно обучилась, NOT_ENOUGH_DATA при недостатке данных для обучения, UNDEFINED_ERROR при получении исключения, отличного от ValueError('not enough data')).
2. для PREDICT результат метода predict в случае успеха, ModelActorResponseStatus.MODEL_IS_NOT_READY, если модель не обучена,

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

ModelActorResponseStatus.NOT_ENOUGH_DATA при недостатке входных данных для прогнозирования и ModelActorResponseStatus.UNDEFINED_ERROR в прочих случаях. Кроме того, формирует сообщение ModelActorMessage с args равным возвращаемому значению и отправляет его sender.

Результаты испытаний модуля:

1. Испытание обучения и прогноза на корректных данных – успешно:

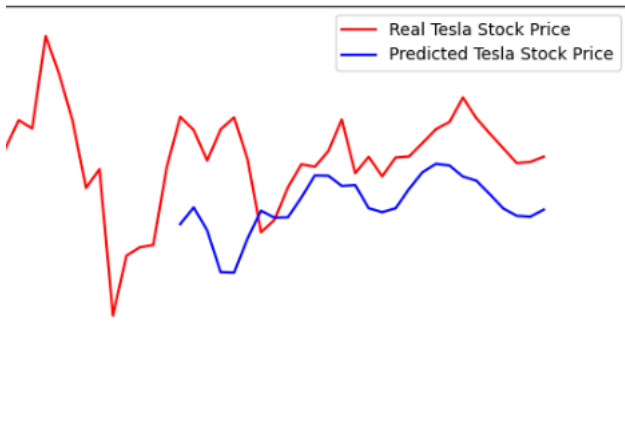


Рис.6 – Испытание прогноза на корректных данных

3. Попытка получения прогноза на необученной модели - ModelActor вернул ModelActorResponseStatus.MODEL_IS_NOT_READY

3. Попытка обучения/получения прогноза при недостаточном количестве входных данных – ModelActor вернул ModelActorResponseStatus.NOT_ENOUGH_DATA

4. При отсутствии спецификации запроса ModelActor вернул ModelActorResponseStatus.UNDEFINED_ERROR

Никаких непредвиденных ошибок и исключений при тестировании не возникло, все возможные случаи корректно обрабатываются. Результаты, возвращаемые как корректные, действительно такими являются.

6.2.4. Испытания требований к модулю принятия решений

1. Получение и обработка прогноза от основного модуля – успешно

2. Получение текущей цены от основного модуля – успешно

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

3. Вычисление, хранение и обновление решений – успешно
 4. Передача отложенного решения основному модулю в нужный момент – успешно
 5. Обновление решений в связи с изменением зарезервированного за позицией количества денег – успешно
 6. Проверка входных данных на актуальность: при отсутствии актуального прогноза решения не принимаются
- Модуль не вызывает никаких ошибок ни при каких обстоятельствах и возвращает только корректные решения.

6.2.5. Испытания требований к основному модулю

1. Создание position_handler для новой позиции – успешно в случае корректного названия акции, иначе вызывает ValueError
2. Удаление позиции – успешно, перед удалением отправляет запрос эмулятору о продаже всех лотов позиции
3. Добавление/изъятие виртуальных средств с распределением их по позициям - успешно в случае корректной суммы, иначе вызывает ValueError
4. Получение свечей и последних цен от API-модуля – успешно
5. Отправка запроса на обучение модели модулю прогнозирования и получения ModelActorResponseStatus.SUCCESS – успешно
6. Отправка запроса на получение прогноза и обработка результата – успешно
7. Отправка полученного прогноза модулю принятия решений – успешно
8. Получение решения от модуля принятия решений и отправка запроса на проведение операции эмулятору - успешно

6.2.6. Испытания к интерфейсному модулю

1. Отрисовка интерфейса при запуске программы – успешно

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

2. Вызов методов основного модуля при добавлении/изъятии пользователем виртуальных средств – успешно
3. Реакция на некорректную сумму для добавления – выводится сообщение “Проверьте корректность введенной суммы”
4. Вызов методов основного модуля при добавлении/удалении наблюдаемых позиций – успешно
5. Реакция на некорректное название акции – выводится сообщение “Проверьте корректность введенного названия”

6.3. Испытания требований к временным характеристикам

Проведены измерения выполнения запроса полного перечня акций – агрегация ответа сервера длилась ~ 0.845с, что в целом удовлетворяет установленным рамкам эффективной работы системы.

Со стороны модели прогнозирования требования к эффективности выполнены – не более 2 часов на обучение, секунды на прогнозирование.

Все функциональные методы основного модуля и модуля принятия решений удовлетворяют требованиям к временным характеристикам – не более одной секунды.

6.4. Испытаний требований к надежности

По результатам испытаний требований к функциональным характеристикам видно, что программа обеспечивает проверку корректности входных данных и не завершается аварийным способом.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 2) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 3) ГОСТ 19.301-77 Программа и методика испытаний. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 4) ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 5) ГОСТ 19.404-78 Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 6) ГОСТ 19.505-78 Руководство оператора. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 7) Tinkoff Invest Api Token// [Электронный ресурс] - Режим доступа: свободный, URL: <https://tinkoff.github.io/investAPI/token/> (5 мая 2022)
- 8) Тинькофф инвестиции // [Электронный ресурс] - Режим доступа: свободный, URL: <https://www.tinkoff.ru/invest/> (5 мая 2022)
- 9) Тинькофф клиент API // [Электронный ресурс] - Режим доступа: свободный, URL: <https://tinkoff.github.io/invest-python/> (5 мая 2022)
- 10) Thespian Actors User`s Guide// [Электронный ресурс] - Режим доступа: свободный, URL: <https://thespianpy.com/doc/using.html> (5 мая 2022)
- 11) Thespian Actors Library// [Электронный ресурс] - Режим доступа: свободный, URL: <https://github.com/kquick/Thespian>(5 мая 2022)
- 12) Finage.co.uk - предоставляет информацию в реальном времени для всех бирж в Соединенных Штатах и для внебиржевых рынков // [Электронный ресурс] - Режим доступа: свободный, URL: <http://finage.co.uk/> (5 мая 2022)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

- 13) Yahoo Finance API // [Электронный ресурс] - Режим доступа: свободный, URL:
<https://www.yahoofinanceapi.com> (5 мая 2022)
- 14) Alpha Vantage API // [Электронный ресурс] - Режим доступа: свободный, URL:
<https://www.alphavantage.co/documentation/> (5 мая 2022)
- 15) EXANTE API // [Электронный ресурс] - Режим доступа: свободный, URL:
<https://exante.eu/ru/technology/> (5 мая 2022)
- 16) Альфа-Банк, торговые роботы // [Электронный ресурс] - Режим доступа: свободный, URL:
<https://alfabank.ru/make-money/investments/torgovye-roboty/> (5 дек. 2021)
- 17) Обзор ботов для торговли криптовалютой // [Электронный ресурс] - Режим доступа: свободный, URL: <https://crypto.ru/boty-dlya-torgovli-kriptovalyutoy/> (5 дек. 2021)
- 18) Best Stock Trading Bots // [Электронный ресурс] - Режим доступа: свободный, URL:
<https://sourceforge.net/software/stock-trading-bots/> (5 дек. 2021)
- 19) Trade Ideas // [Электронный ресурс] - Режим доступа: свободный, URL:
<https://sourceforge.net/software/product/Trade-Ideas/> (5 дек. 2021)
- 20) Stoic AI // [Электронный ресурс] - Режим доступа: свободный, URL:
<https://sourceforge.net/software/product/Stoic/> (5 дек. 2021)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

ПРИЛОЖЕНИЕ 1

Общая концепция интерфейса программы

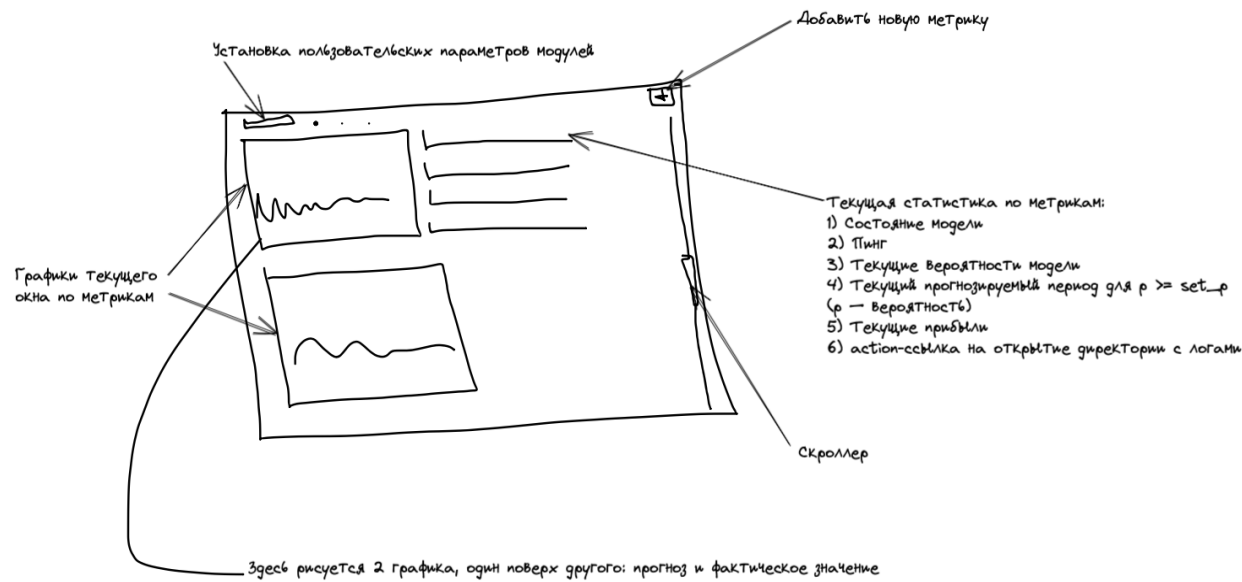


Рис. - Схема интерфейса.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

ПРИЛОЖЕНИЕ 2

СРАВНИТЕЛЬНАЯ ТАБЛИЦА АНАЛОГОВ ПРОЕКТА «БИРЖЕВОЙ БОТ»

	Tickeron	Trade Ideas ^[22]	Stoic AI ^[23]	Streak	Биржевой бот
Наличие открытого исходного кода	-	-	-	-	+
Модульность программы	-	-	-	-	+
Заявленное использование актор-системы как модели параллельных вычислений	-	-	-	-	+
Кроссплатформенность	+	+	+	+	***
Бесплатная подписка	-	-	+	-	+
Бесплатный пробный период	-	+	+	+	+
Операции с криптовалютой(криптотрейдинг)	+	-	+	+	-*
Отображение (в виде временного ряла) детального прогноза real-time	-	-	-	-	+
Возможность виртуальной торговли	-	-	-	+	+

*Благодаря модульности программы может быть добавлен соответствующий модуль

**Может быть собрано пользователем на любой системе на свой страх и риск

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

ПРИЛОЖЕНИЕ 3

Развернутый список требований к API для получения информации о ценах акций

Использование токена для работы с Tinkoff API^[12]. Токен генерируется в личном кабинете сайта Тинькофф Инвестиции^[11], срок жизни токена – 3 месяца с даты последнего использования. В проекте используется readonly token^[10] — он предназначен только для получения различной информации, такой как состояние портфеля, расписание торгов различных торговых площадок, текущие котировки, исторические данные и т.п., а выставление торговых поручений невозможно.

1. Предоставление модулям программы полного перечня акций.

Предоставляется: время запроса – общее для всех акций, figi\имя\цена\валюта\статус покупки\статус продажи\лот – для каждой акции.

2. Предоставление перечня акций, доступных к покупке.

Предоставляется: время запроса – общее для всех акций, figi\имя\цена\валюта\статус покупки\статус продажи\лот – для каждой акции. Отличие от предыдущего пункта - статус покупки у акций в предоставляемом перечне всегда true.

3. Предоставление перечня акций, доступных к продаже.

Предоставляется: время запроса – общее для всех акций, figi\имя\цена\валюта\статус покупки\статус продажи\лот – для каждой акции. Отличие от предыдущего пункта - статус продажи у акций в предоставляемом перечне всегда true.

4. Предоставление цены для конкретной акции за произвольный период в прошлом.

Произвольный период и шаг подаются как входные данные, шаг – минута\5минут\15 минут\час\день.

Предоставляется список время-цена за запрашиваемый период.

5. Получение информации об акции по ее названию.

Требование, существующее для выполнения запросов информации по конкретной акции – либо получение информации (цена\валюта\статус покупки\продажи\лот) на момент запроса, т. е. актуальной, либо получение информации по конкретной акции за произвольный период в прошлом.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

СРАВНЕНИЕ СУЩЕСТВУЮЩИХ API И ОБОСНОВАНИЕ ВЫБОРА

	Finage.co.uk ^[15]	Alpha Vantage API ^[17]	EXANTE ^[18]	Yahoo Finance API ^[16]	Tinkoff API ^[12]
Бесплатная подписка	+	+	+	+	+
Верификация данных пользователя (удостоверения личности) для использования сервиса	-	-	+	-	+
Лимиты на запрос (обращения к API)	+	-	-	+	-
Подробная документация и примеры использования	+	+	-	-	+
Возможность получения текущих цен акций	+	-	+	+	+
Возможность получения данных за	+	+	+	-	+

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

30
RU.17701729.05.01-01 51 01-1

прошедший период					
Возможность использования для реальной торговли на бирже (личное пользование)	-	-	+	-	+

Цветом в табличке обозначена польза для проекта. Зеленый – подходит, красный – не подходит. Насыщенный красный – категорически не подходит. По цветовой схеме видно, что под все предъявляемые требования идеально подошел Tinkoff API.

Следует отметить, что этот выбор не лишен недостатков: для его использования необходим открытый в банке инвестиционный счет, а публичный сервис на основе TINKOFF INVEST API сделать невозможно, т. к. это противоречит установленным условиям использования (TINKOFF INVEST API предоставляется только клиентам Тинькофф без права на ретрансляцию данных).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв.	Инв. №	Подп. и

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]