

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук
Кафедра информационных технологий управления

Техническое задание к проекту «Ведение домашнего бюджета»
в соответствии с ГОСТ 34.602-89

Исполнитель _____ А.А. Пустовалов, 3 курс, д/о
Исполнитель _____ В.Г. Новиков, 3 курс, д/о
Исполнитель _____ Е.О. Бордюжа, 3 курс, д/о
Заказчик _____ В.С. Тарасов, старший преподаватель

Воронеж 2023

Содержание

Содержание.....	2
1 Понятие и термины	4
2 Общие сведения.....	7
2.1 Полное наименование системы и название веб-приложения	7
2.2 Наименование исполнителя и заказчика приложения.....	7
2.2.1 Наименование исполнителя	7
2.2.2 Наименование заказчика	7
2.3 Перечень документов, на основании которых создается сайт.....	7
2.4 Плановые сроки начала и окончания работ	7
2.5 Порядок оформления и предъявления заказчику результатов работ по созданию веб-приложения	7
3 Назначение и цели создания веб-приложения	9
3.1 Назначение веб-приложения	9
3.2 Цели создания веб-приложения	9
3.3 Характеристика объектов автоматизации	9
4 Требования к веб-приложению и программному обеспечению	10
4.1 Требования к веб-приложению в целом.....	10
4.1.1 Требования к структуре и функционированию веб-приложения	10
4.1.2 Требования к защите информации	11
4.1.3 Требования к оформлению и верстке страниц.....	11
4.1.4 Требования к архитектуре	11
4.1.5 Требования к используемым технологиям	12
4.1.6 Требования к численности и квалификации персонала, обслуживающего приложение	14

4.2 Требования к функциям, выполняемым веб-приложением	14
5 Языковые версии веб-приложения	15
6 Группы пользователей	16
7 Дизайн сайта	18
8 Навигация по веб-приложению	19
9 Описание страниц сайта	20
9.1 Запуск приложения	20
9.2 Главная страница	20
9.2.1 Окно добавления трат и поступлений	20
9.2.2 Окно просмотра статистики	20
9.3 Профиль	21
9.4 Боковое меню	21
9.5 Центр уведомлений	21
10 Перспективы развития, модернизации веб-приложения	22
11 Требования к документированию	23
12 Порядок контроля и приемки работы	24
Приложение	25
Реквизиты и подписи сторон	31

1 Понятие и термины

- Авторизация – это процесс проверки прав пользователя на осуществление определенных действий на сайте;
- База данных – это упорядоченный набор структурированной информации или данных, которые хранятся в электронном виде в компьютерной системе;
- Бюджет – это расходы и доходы конкретного человека или группы лиц;
- Веб-приложений, веб-сервис, интернет-сервис, онлайн-сервис, проект – это программное обеспечение, которое размещено на удаленном сервере и доступно через браузеры в интернете;
- Демо-режим – это режим работы приложения, в котором доступна ограниченная функциональность;
- Заглушка – это временное решение отображения информации на веб-сайте, состоящее только из картинки или одной веб-страницы;
- Клиент – это объект, запрашивающий информацию по сети;
- Транскомпиляция – это конвертация кода одного языка в другой. С помощью специального транскомпилятора один язык программирования общего назначения можно перенести на любой другой язык программирования общего назначения;
- Личный кабинет, профиль – это раздел сервиса, в котором пользователь может получить доступ к персональным данным;
- Мониторинг – это отслеживание расходов и доходов по категориям;
- Развертывание – это все действия, которые делают систему готовой к использованию;
- Регистрация – это способ сообщить сервису данные о себе и в обмен получить доступ к дополнительным ресурсам на сайте, которые недоступны гостям;

- Резиновая верстка – это подход к веб-разработке, в рамках которого создаются масштабируемые сайты, способные подстраиваться под разрешение текущего экрана;
- Сервер – это отдельный класс компьютерных устройств, предназначенных для обработки запросов от различных узлов сети;
- СУБД – это система управления базы данных;
- Фреймворк – это программная среда, облегчающая разработку и объединение разных компонентов большого программного проекта;
- Хеширование – это представляет собой преобразование любого объема информации в уникальный набор символов, который присущ только этому массиву входящей информации;
- Frontend – это клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса;
- Backend – это программно-аппаратная часть сервиса;
- REST API – это стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов;
- React – это JavaScript библиотека для создания пользовательских интерфейсов;
- SPA – это одностраничное веб-приложение, которое работает на одной HTML-странице, обновляя данные на ней;
- GitHub – это крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки;
- HTTP, HTTPS – это широко распространённый протокол передачи данных, предназначенный для передачи документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам;
- HTML – это стандартизированный язык разметки документов для просмотра веб-страниц в браузере;

- Header – это визуальный элемент, расположенный в верхней части страницы;
- MVC архитектура – это способ организации кода, который предполагает выделение блоков, отвечающих за решение разных задач;
- SQL-запросы – это наборы команд для работы с реляционными (табличными) базами данных;
- SQL-инъекция – это один из способов взлома сайта.

2 Общие сведения

2.1 Полное наименование системы и название веб-приложения

Полное наименование: приложение для мониторинга бюджета.

Название веб-приложения: MoneyPie.

2.2 Наименование исполнителя и заказчика приложения

2.2.1 Наименование исполнителя

Студент 3-го курса Пустовалов Артем Алексеевич, кафедра информационных систем и технологий управления предприятием.

Студент 3-го курса Новиков Виктор Григорьевич, кафедра информационных систем и технологий управления предприятием.

Студент 3-го курса Бордюжа Екатерина Олеговна, кафедра информационных систем и технологий управления предприятием.

2.2.2 Наименование заказчика

Старший преподаватель Тарасов Вячеслав Сергеевич, кафедра программирования и информационных технологий.

2.3 Перечень документов, на основании которых создается сайт

На основании Технического Задания, документа «Стратегия повышения финансовой грамотности в Российской Федерации на 2017-2023 годы» и закона «Об информации, информационных технологиях и о защите информации» от 27.07.2006 №149-ФЗ создается данное веб-приложение.

2.4 Плановые сроки начала и окончания работ

Плановый срок начала работ – март 2023 г.

Плановый срок окончания работ – июнь 2023 г.

2.5 Порядок оформления и предъявления заказчику результатов работ по созданию веб-приложения

Предъявление результатов работ заказчику осуществляется в следующем виде:

- Работающее веб-приложение согласно Техническому Заданию;
- Документация к веб-приложению;

- Презентация и видео с демонстрацией функциональности веб-приложения;
- Исходный код веб-приложения;
- Защита проекта.

3 Назначение и цели создания веб-приложения

3.1 Назначение веб-приложения

Назначением приложения является автоматизация учета и контроля финансов пользователя.

3.2 Цели создания веб-приложения

Данное приложение создается для получения прибыли от премиум-подписок.

3.3 Характеристика объектов автоматизации

Объектом автоматизации является процесс группировки расходов и доходов пользователя (или пользователей, если ведется совместный бюджет) по категориям с возможностью просмотра статистики по каждой из них.

4 Требования к веб-приложению и программному обеспечению

4.1 Требования к веб-приложению в целом

Приложение должно удовлетворять следующим требованиям:

- Обеспечить работу в Google Chrome, Firefox и Microsoft Edge;
- Предоставить возможность регистрации и авторизации;
- Предоставить уже авторизованному пользователю пригласить друга для ведения совместного бюджета;
- Предоставить статистику для анализа расходов и доходов по категориям. Авторизованный пользователь может смотреть, сколько денег он тратит и зарабатывает в каждой категории, например, в категории "Продукты", "Здоровье" или "Развлечения";
- На данном этапе разработки не будет интеграции с платежными сервисами, поэтому все финансовые операции, к примеру оформление премиум подписки, будут реализованы в виде заглушек.

4.1.1 Требования к структуре и функционированию веб-приложения

При разработке клиентской части будет использоваться библиотека React, которая позволяет разрабатывать SPA, а также предлагает компонентный подход, при котором интерфейсы собираются из отдельных элементов (компонентов).

Приложение должно запускаться в следующих браузерах:

- Google Chrome (версия 90.0.4430.93 или новее);
- Firefox (версия 90.0.2 или новее);
- Microsoft Edge (версия 90.0.818.39 или новее).

Для связи клиента и сервера следует использовать REST API, которое представляет из себя интерфейс, используемый двумя компьютерными системами для безопасного обмена информацией через сеть Интернет.

В качестве основного языка программирования будет использован JavaScript. Разработка будет вестись на языке TypeScript, который является обратно совместимым с JavaScript и транскомпилируется в последний.

После транскомпиляции программу на TypeScript можно выполнять в любом современном браузере как JavaScript код или использовать совместно с серверной платформой Node.js. Код транскомпилятора, транслирующего TypeScript в JavaScript, распространяется под лицензией Apache. Его код представлен в публичном доступе на сервисе GitHub.

В качестве СУБД будет использоваться реляционная СУБД PostgreSQL.

4.1.2 Требования к защите информации

Разрабатываемое веб-приложение должно быть защищено от SQL и XSS инъекций. Пароли пользователей в базе данных должны хешироваться.

4.1.3 Требования к оформлению и верстке страниц

Страницы должны быть сверстаны с использованием принципов резиновой верстки, которые позволяют перестроить и адаптировать интерфейс к различным размерам экрана.

Также должна быть реализована кроссбраузерная поддержка веб-приложения, которая заключается в том, что разрабатываемый веб-сайт должен отображаться и функционировать во всех перечисленных ниже браузерах идентично. Под идентичностью функционирования подразумевается: отсутствие некорректной работы, отсутствие ошибок в верстке и способность отображать материал с одинаковой степенью читабельности.

Обеспечить поддержку следующих браузеров:

- Google Chrome (версия 90.0.4430.93 или новее);
- Firefox (версия 90.0.2 или новее);
- Microsoft Edge (версия 90.0.818.39 или новее).

4.1.4 Требования к архитектуре

Серверная архитектура должна быть реализована в соответствии с подходом MVC, который подразумевает разделение данных и логики приложения на три отдельных части, а именно модель (model), вид (view) и контроллер (controller). Контроллер обрабатывает входящие запросы на сервер. Модель запрашивает из базы данных информацию, нужную для выполнения конкретных запросов. Вид определяет результат запроса, который получает пользователь. Для связи клиент-сервер следует реализовать REST API.

4.1.5 Требования к используемым технологиям

При разработке Frontend части приложения будут использованы следующие технологии:

- JavaScript (JS) – язык программирования, который выполняется внутри браузера и позволяет внедрять в сайт различные функции на стороне клиента;
- TypeScript (TS) – язык программирования, представленный Microsoft и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript, самый популярный сценарий использования – введение в код на этапе разработки четкой системы типов, которая позволяет избежать различных ошибок в типизации при транскомпиляции в JS. Является обратно совместимым с JS и транскомпилируется в него же;
- React – библиотека с открытым исходным кодом, позволяющая создавать одностраничные приложения, работает как с JS, так и с TS.

При разработке Backend части приложения будут использованы следующие технологии:

- Основной язык – TypeScript, исполняемый на сервере в среде Node.js в связке с фреймворком Nest.js. Nest.js – это платформа для создания серверных приложений на Node.js. Сам фреймворк построен с

использованием TypeScript и полностью поддерживает его (при этом позволяет разработчикам использовать чистый JavaScript). Позволяет реализовать MVC архитектуру;

- Приложение будет оперировать реляционной БД, в качестве СУБД будет использоваться СУБД с открытым исходным кодом PostgreSQL;
- В качестве инструмента развертки приложения будет использоваться Docker, который позволяет автоматизировать процесс развертывания и управления приложениями;
- Для документации разрабатываемого REST API будет использоваться Swagger, предоставляющий набор инструментов, который позволяет автоматически описывать API на основе его кода.

Сервисы, используемые в процессе разработки:

- Git – распределенная система контроля версий, позволяющая отслеживать, вносить и удалять изменения;
- GitHub – крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git;
- Miro – платформа для визуализации гипотез и моделей, предоставляющая возможности для совместной работы распределенных команд;
- Trello – облачная программа для управления проектами небольших групп, позволяет наглядно отслеживать задачи каждого участника разработки;
- Figma – онлайн-сервис для разработки и прототипирования интерфейсов с возможностью организации совместной работы в режиме реального времени.

4.1.6 Требования к численности и квалификации персонала, обслуживающего приложение

В случае необходимости исправления ошибок в работе системы, актуализации информации о финансовой грамотности в базе данных или развития приложения требуются:

- Один или более разработчиков со знаниями PostgreSQL, TypeScript и Nest.JS для работы с базой данных и серверной частью приложения;
- Один или более разработчиков со знаниями TypeScript и React.

4.2 Требования к функциям, выполняемым веб-приложением

Разрабатываемое веб-приложение должно обладать следующими функциональными возможностями:

- Возможность регистрации и авторизации для пользователей;
- Добавление записей расходов и доходов по категориям;
- Просмотр статистики расходов и доходов по каждой категории за определенный период;
- Реализация возможности приглашения друга для ведения совместного бюджета;
- Предоставление пользователю информации о финансовой грамотности.

5 Языковые версии веб-приложения

На данном этапе разработки предусмотрена только русскоязычная версия приложения.

6 Группы пользователей

В системе существуют такие группы пользователей как: неавторизованный, авторизованный, премиум-пользователь и участник группы.

Разрабатываемое приложение предоставляет возможность вести совместный бюджет группе пользователей. Вместимость группы по умолчанию составляет два человека, но если хотя бы один участник группы является премиум-пользователем, то вместимость группы увеличивается до пяти человек, что является максимальной вместимостью.

Неавторизованный пользователь – посетитель веб-сайта, узнавший о сервисе из поисковой выдачи или любым другим способом. Для неавторизованного пользователя должна быть реализована следующая функциональность:

- Регистрация в веб-приложении;
- Вход в демо-режим, в котором доступна ограниченная функциональность веб-приложения, а именно: просмотр трат, внесенных заранее (данные траты являются предзаписанными неизменяемыми данными), возможность фильтрации таких трат.

Авторизованный пользователь – пользователь, прошедший процесс авторизации. Для авторизованного пользователя должна быть реализована следующая функциональность:

- Приглашение одного друга в группу для ведения совместного бюджета;
- Добавление доходов и расходов. При внесении трат и поступлений можно выбрать одну из предложенных категорий. Без премиум-подписки пользователь ограничен количеством таких категорий;
- Оформление премиум подписки;
- Просмотр статистики трат;
- Фильтрация трат по категориям;

- Возможность выхода из аккаунта.

Премиум-пользователь – авторизованный пользователь, который приобрел подписку на сервис, открывающую дополнительные функциональные возможности. Для премиум-пользователя должна быть реализована та же функциональность, что и для авторизованного, а также:

- Увеличение количества возможных участников в группе, в которой состоит премиум-пользователей до пяти человек;
- Неограниченное количество используемых категорий.

Участник группы – авторизованный пользователь, ведущий совместный бюджет. Разрешается состоять не более чем в одной группе. Данному пользователю доступна следующая функциональность:

- Добавление трат в общий бюджет;
- Добавление трат в личный бюджет. Траты в личном бюджете являются приватными, то есть они не видны другим пользователям группы;
- Просмотр статистики трат;
- Фильтрация трат по категориям;
- Возможность выхода из группы.

7 Дизайн сайта

Дизайн сайта должен быть выполнен в минималистичном стиле, элементы интерфейса сглажены. Основной шрифт Montserrat и различные его начертания. Основные цвета: базовые (#141516, #FFFFFF), второстепенные (#363636, #A29581), акцентные (#F5A932, #F2DCBB), дополнительные (#59345C, #A17BA4). Допускается использование градиентов и теней, выдержанных в общей палитре.

8 Навигация по веб-приложению

Приложение использует для навигации меню в верхней части экрана. Кнопка меню открывает боковую панель, которая позволяет перейти в следующие разделы сайта: «профиль», «главная страница», «статистика», «премиум-подписка».

Для пользователя, который не состоит в группе начальным экраном будет являться вкладка «личный бюджет». Для пользователя, состоящего в группе, начальная вкладка по умолчанию «общий бюджет».

9 Описание страниц сайта

9.1 Запуск приложения

На странице содержатся следующие элементы:

- Форма регистрации;
- Кнопка «Зарегистрироваться»;
- Ссылка «Уже есть аккаунт? Войти»;
- Кнопка закрытия формы;
- Поля для ввода информации, необходимой для регистрации пользователя в системе.

При первом запуске приложения неавторизованный пользователь попадает в демо-режим, из которого он может перейти на страницу регистрации.

Если регистрация прошла успешно, после нажатия на кнопку «Зарегистрироваться» происходит переход на страницу входа. Если при регистрации возникла какая-то ошибка, пользователь остается на той же странице и получает сообщение об ошибке.

Если пользователь уже зарегистрирован, то при нажатии на ссылку «Уже есть аккаунт? Войти» происходит переход на страницу авторизации.

9.2 Главная страница

После авторизации пользователь попадает на главную страницу, которая представляет собой список недавних трат. На ней находится кнопка добавления трат и поступлений, кнопка просмотра статистики, а также две вкладки личный и общий бюджет.

9.2.1 Окно добавления трат и поступлений

В данном окне можно добавить трату или поступление, выбрать категорию, дату и сумму.

9.2.2 Окно просмотра статистики

В верхней части окна находится header с возможностью перейти в меню или вернуться на домашнюю страницу. Также представлена статистика трат и

поступлений по категориям в текстовом виде, а также график расходов и доходов.

9.3 Профиль

На данной странице содержатся:

- Управление подпиской (возможность оформить, продлить или отказаться от нее);
- Кнопка добавить пользователя в группу;
- Кнопка выхода из аккаунта;
- Кнопка выхода из группы.

9.4 Боковое меню

Данный элемент навигации позволяет перейти в следующие разделы сайта: профиль, главная страница, статистика, премиум подписка.

9.5 Центр уведомлений

Кнопка «центр уведомлений» находится в правом верхнем углу header. При нажатии открывает окно, в которое приходят уведомления и советы по финансовой грамотности.

10 Перспективы развития, модернизации веб-приложения

- Развитие демо-режима путем внедрения возможности добавлять пользовательские траты, а не только просматривать предзаписанные;
- Приглашение неавторизованных пользователей;
- Добавление ограничения для конкретной категории трат, выбранной пользователем. Если лимит превышен система сигнализирует об этом сообщением;
- Добавление для премиум-пользователя возможности создания авторской категории трат;
- Добавление светлой темы в приложение;
- Переход от резиновой верстки к адаптивной;
- Локализация на иные языки (прежде всего английский);
- Добавление поддержки различных валют;
- Интеграция с банковскими API.

11 Требования к документированию

Документирование системы ведется в рамках Технического Задания в соответствии с ГОСТ 34.602-89.

Предоставление курсового проекта осуществляется на основе данного Технического задания.

Отслеживание рабочего процесса и управлением задачами участников проекта производится при помощи инструмента Trello.

Описание основных сценариев работы приложения осуществляется в сервисе Miro.

Документирование API обеспечивается с помощью инструмента Swagger.

12 Порядок контроля и приемки работы

Контроль разработки системы осуществляется путём регулярных встреч каждые две недели с представителем заказчика Зениным К.В. для обсуждения промежуточных результатов работы в рамках рубежных аттестаций. Готовая система с полной документацией будет представлена заказчику в обозначенную им дату. Заказчик определит соответствие системы всем требованиям и осуществит её приём.

Исполнитель должен предоставить следующее:

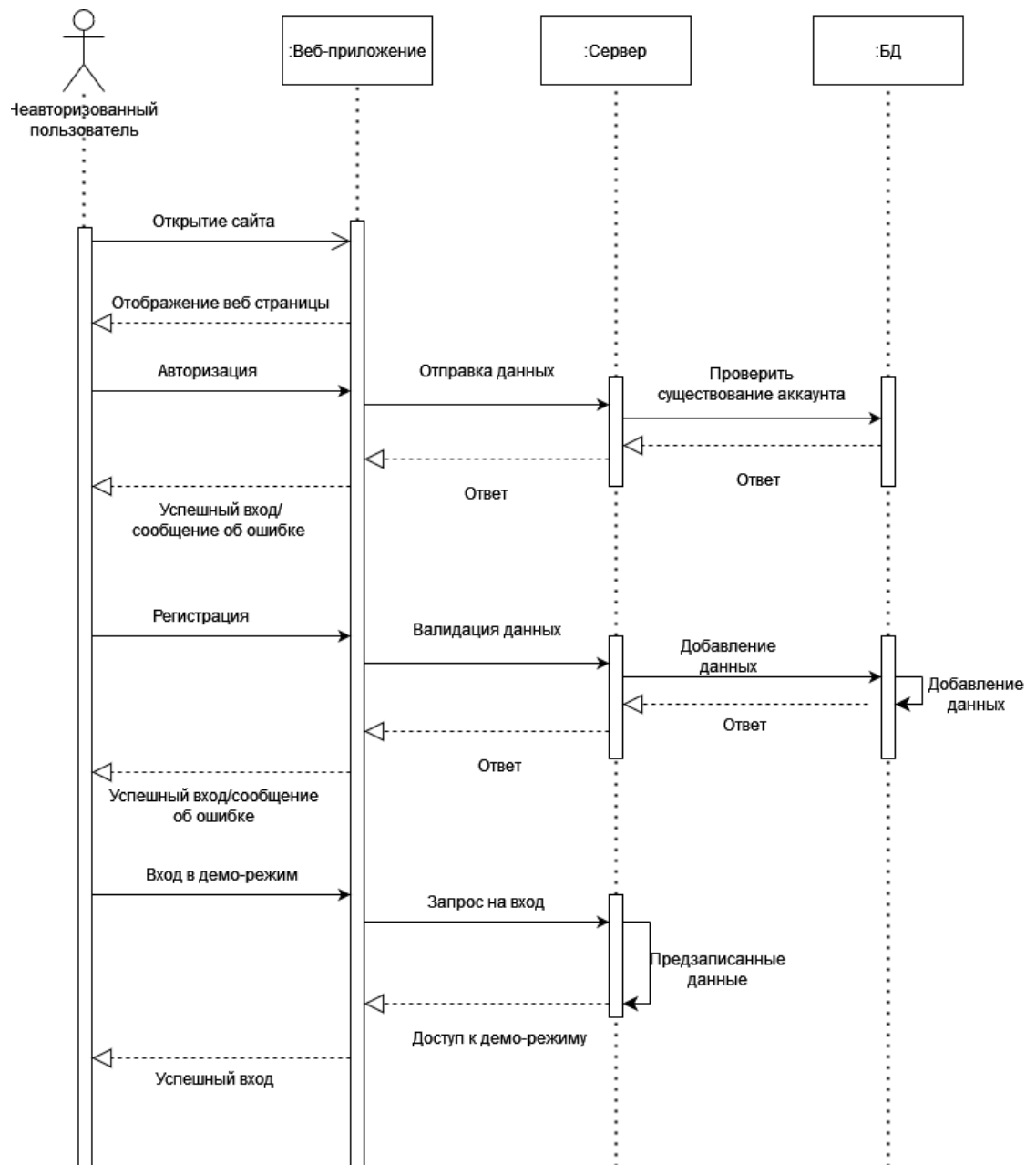
- Техническое задание;
- Демонстрационную версию проекта со всеми ключевыми сценариями;
- Курсовой проект;
- Видео-презентацию работы приложения;
- Исходный код системы.

Приложение

Диаграмма прецедентов:



Диаграмма последовательностей:



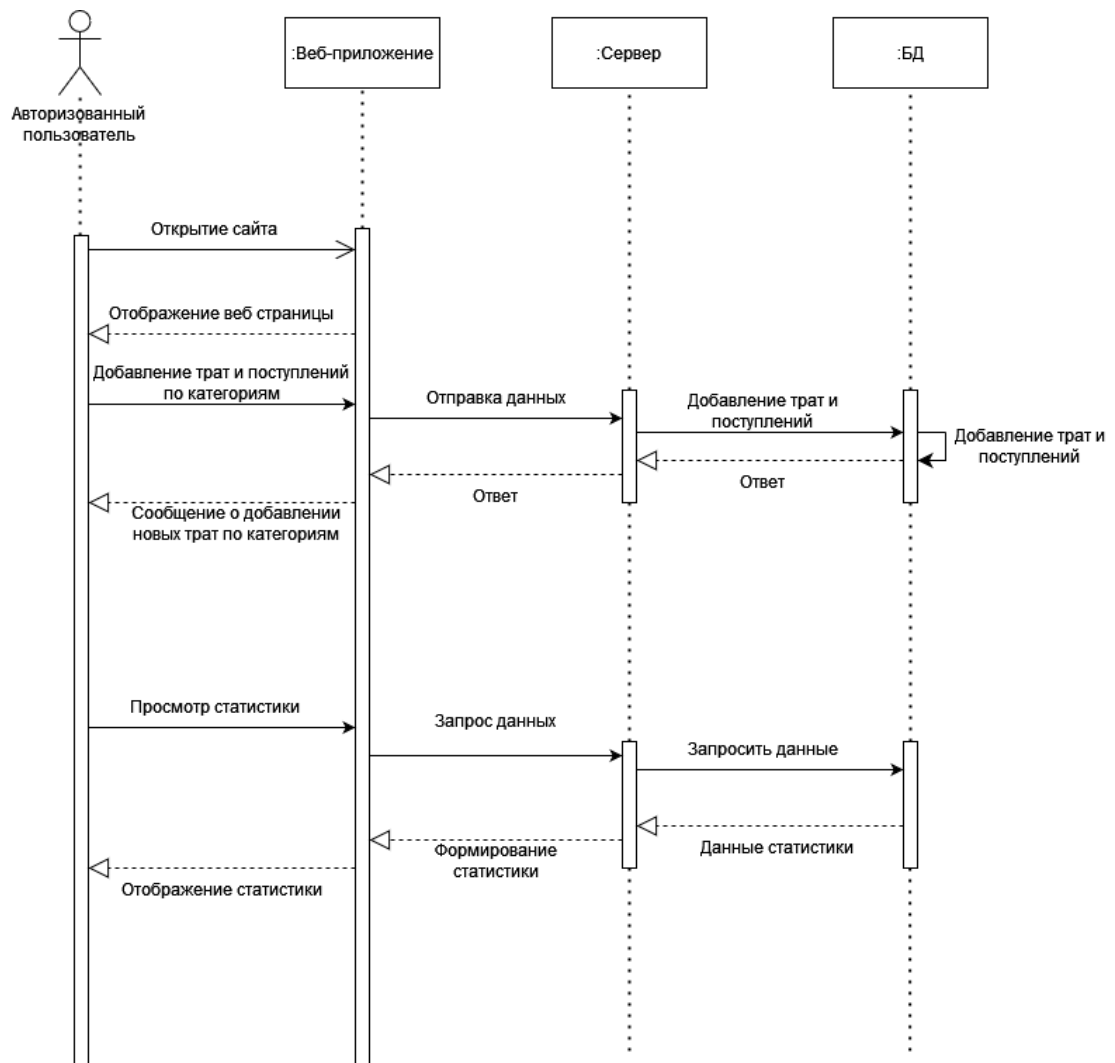
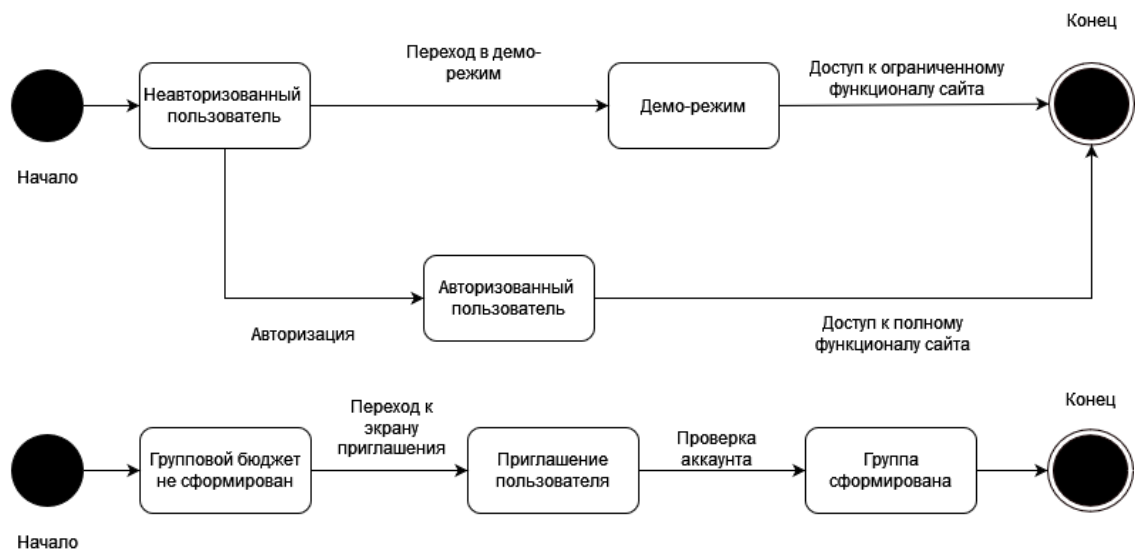


Диаграмма состояний:



Контекстная диаграмма (IDEF0):

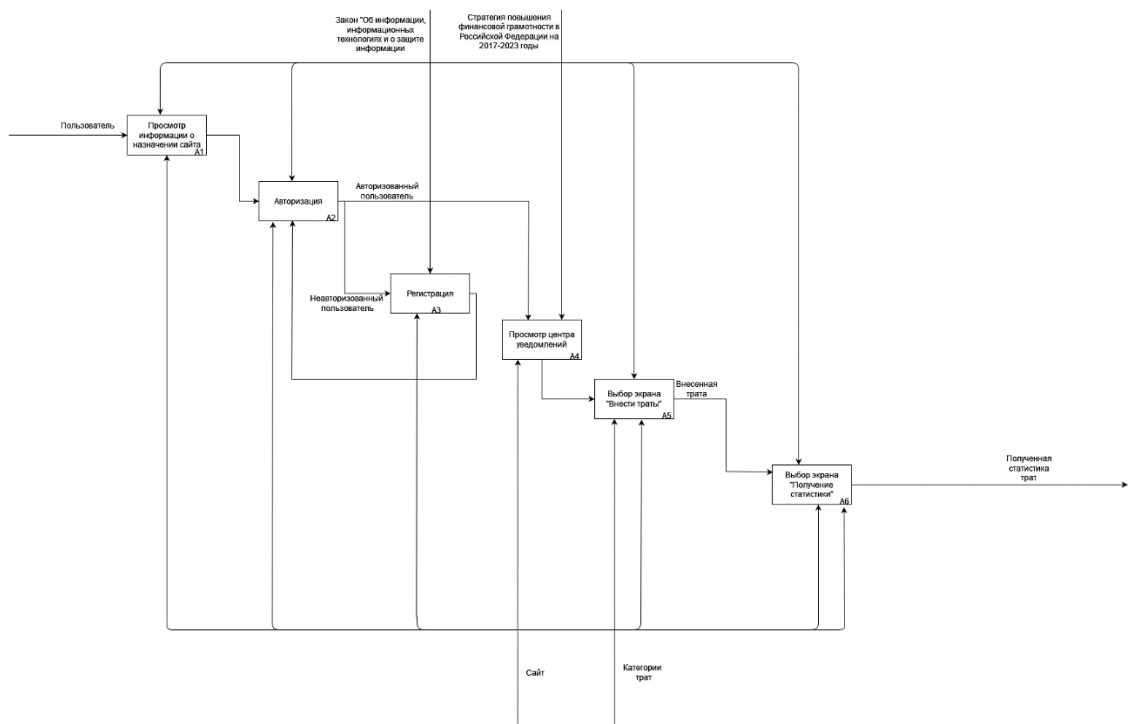
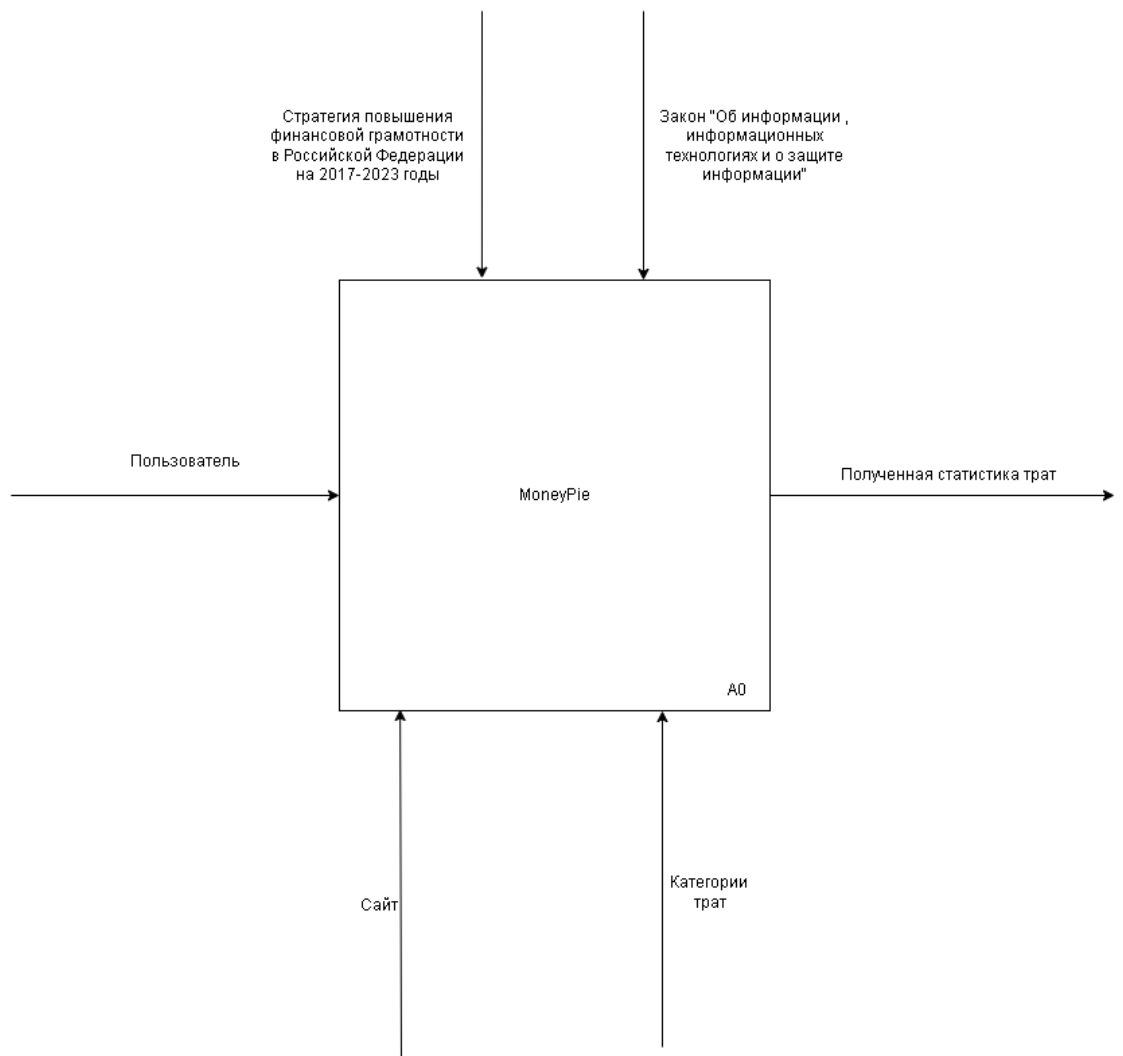


Диаграмма сотрудничества:



Диаграмма активностей:

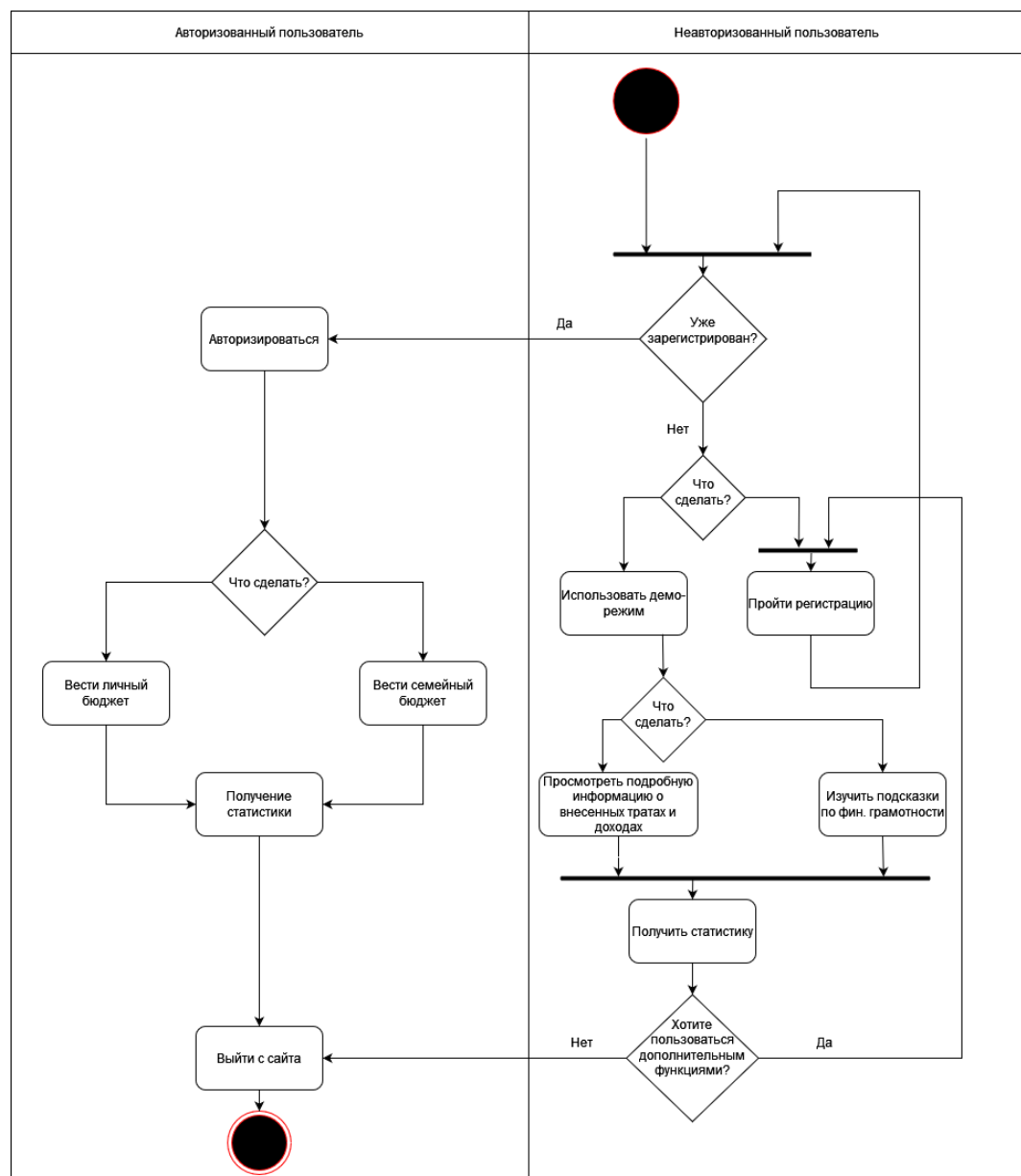
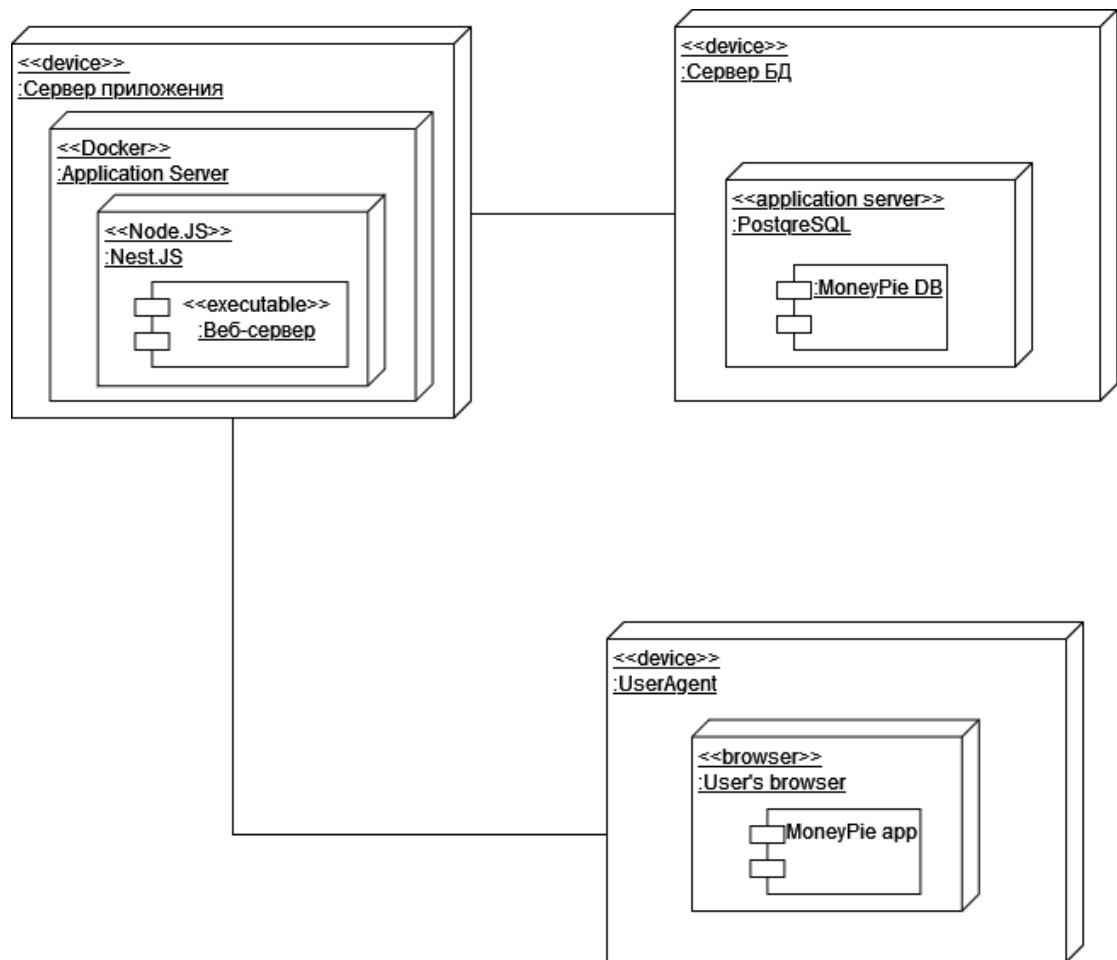


Диаграмма развертывания:



Реквизиты и подписи сторон

СОСТАВИЛИ

Наименование организации, предприятия	Должность исполнителя	Фамилия, имя, отчество	Подпись	Дата
Воронежский Государственный Университет	Студент	Пустовалов Артем Алексеевич		24.03.2023
Воронежский Государственный Университет	Студент	Новиков Виктор Григорьевич		24.03.2023
Воронежский Государственный Университет	Студент	Бордюжа Екатерина Олеговна		24.03.2023

СОГЛАСОВАНО

Наименование организации, предприятия	Должность исполнителя	Фамилия, имя, отчество	Подпись	Дата
Воронежский Государственный Университет	Старший преподаватель	Тарасов Вячеслав Сергеевич		24.03.2023
Воронежский Государственный Университет	Ассистент	Зенин Кирилл Вячеславович		24.03.2023