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](#_Toc130833287)

[1 Понятие и термины 4](#_Toc130833288)

[2 Общие сведения 7](#_Toc130833289)

[2.1 Полное наименование системы и название веб-приложения 7](#_Toc130833290)

[2.2 Наименование исполнителя и заказчика приложения 7](#_Toc130833291)

[2.2.1 Наименование исполнителя 7](#_Toc130833292)

[2.2.2 Наименование заказчика 7](#_Toc130833293)

[2.3 Перечень документов, на основании которых создается веб-приложение 7](#_Toc130833294)

[2.4 Плановые сроки начала и окончания работы 7](#_Toc130833295)

[2.5 Порядок оформления и предъявления заказчику результатов работ по созданию веб-приложения 7](#_Toc130833296)

[3 Назначение и цели создания веб-приложения 9](#_Toc130833297)

[3.1 Назначение веб-приложения 9](#_Toc130833298)

[3.2 Цели создания веб-приложения 9](#_Toc130833299)

[3.3 Характеристика объектов автоматизации 9](#_Toc130833300)

[4 Требования к веб-приложению и программному обеспечению 10](#_Toc130833301)

[4.1 Требования к веб-приложению в целом 10](#_Toc130833302)

[4.1.1 Требования к структуре и функционированию веб-приложения 10](#_Toc130833303)

[4.1.2 Требования к защите информации 11](#_Toc130833304)

[4.1.3 Требования к оформлению и верстке страниц 11](#_Toc130833305)

[4.1.4 Требования к архитектуре 12](#_Toc130833306)

[4.1.5 Требования к используемым технологиям 12](#_Toc130833307)

[4.1.6 Требования к численности и квалификации персонала, обслуживающего приложение 14](#_Toc130833308)

[4.2 Требования к функциям, выполняемым веб-приложением 15](#_Toc130833309)

[5 Языковые версии веб-приложения 16](#_Toc130833310)

[6 Группы пользователей 17](#_Toc130833311)

[7 Дизайн сайта 20](#_Toc130833312)

[8 Навигация по веб-приложению 21](#_Toc130833313)

[9 Описание страниц сайта 22](#_Toc130833314)

[9.1 Запуск приложения 22](#_Toc130833315)

[9.2 Лэндинг 22](#_Toc130833316)

[9.3 Главная страница 22](#_Toc130833317)

[9.3.1 Окно добавления трат и поступлений 22](#_Toc130833318)

[9.3.2 Окно просмотра статистики 22](#_Toc130833319)

[9.4 Профиль 23](#_Toc130833320)

[9.5 Боковое меню 23](#_Toc130833321)

[10 Перспективы развития, модернизации веб-приложения 24](#_Toc130833322)

[11 Требования к документированию 25](#_Toc130833323)

[12 Порядок контроля и приемки работы 26](#_Toc130833324)

[Реквизиты и подписи сторон 27](#_Toc130833325)

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.
* Гость – это неавторизованный на сервисе человек.
* Демо-вход – это вход в веб-приложение под предварительно авторизованным пользователем.
* Заглушка –это временное решение отображения информации на веб-сайте, состоящее только из картинки или одной веб-страницы.
* Клиент – это объект, запрашивающий информацию по сети.
* 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.
* 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 загружается на одну HTML-страницу.
* 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 самых доступных способов взлома сайта.

1. Общие сведения
   1. Полное наименование системы и название веб-приложения

Полное наименование: приложение для мониторинга бюджета.

Название веб-приложения: MoneyPie.

* 1. Наименование исполнителя и заказчика приложения
     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.

* + 1. Наименование заказчика

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

* 1. Перечень документов, на основании которых создается веб-приложение

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 N 149-ФЗ создается данное веб-приложение для мониторинга бюджета.

* 1. Плановые сроки начала и окончания работы

Плановый срок начала работ – март 2023 г.

Плановый срок окончания работ – июнь 2023 г.

* 1. 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.

1. Назначение и цели создания веб-приложения
   1. Назначение веб-приложения

Назначением приложения является автоматизация учёта и контроля финансов пользователя.

* 1. Цели создания веб-приложения

Онлайн-сервис MoneyPie создается для выполнения следующих целей:

* Систематизированное ведение учета расходов и доходов, как личных, так и общих;
* Информирование пользователя о практиках финансовой грамотности;
* Отображение статистики расходов и доходов;
* Формирование рекомендаций по сокращению расходов;
* Возможность приглашать пользователей для ведения совместного бюджета.
  1. 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.

1. Требования к веб-приложению и программному обеспечению
   1. Требования к веб-приложению в целом

Приложение должно удовлетворять следующим требованиям:

* Сайт должен работать в следующих браузерах: Google Chrome (начиная с версии 90), Firefox (начиная с версии 90) и Microsoft Edge (начиная с версии 90);
* Предоставляет возможность регистрации и авторизации;
* Позволяет уже авторизованному пользователю пригласить друга или члена семьи для ведения совместного бюджета;
* Предоставляет статистику для анализа расходов и доходов по категориям. Авторизованный пользователь может посмотреть, 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.
  + 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).

Должна быть реализована стабильная работа приложения в следующих браузеров:

* Google Chrome (версия 90.0.4430.93 и новее);
* Firefox (версия 90.0.2 и новее);
* Microsoft Edge (версия 90.0.818.39 и новее).

Для связи клиента и сервера следует использовать RESTful 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 компилируется в последний.

После компиляции программу на TypeScript можно выполнять в любом современном браузере как JavaScript код или использовать совместно с серверной платформой Node.js. Код компилятора, транслирующего TypeScript в JavaScript, распространяется под лицензией Apache. Его разработка ведётся в публичном репозитории и представлена на сервисе GitHub.

В качестве СУБД будет использоваться реляционная open-source СУБД PostgreSQL.

* + 1. Требования к защите информации

Разрабатываемое веб-приложение должно быть защищено от самых распространённых способов взлома и несанкционированного доступа, к примеру от SQL и XSS инъекций. Пароли пользователей в базе данных должны хешироваться.

* + 1. Требования к оформлению и верстке страниц

Для удобства пользователя страницы должны быть сверстаны с использованием принципов резиновой верстки, которые позволяют перестроить и адаптировать интерфейс к различным размерам экрана конечного потребителя услуг.

Также должна быть реализована кроссбраузерная поддержка веб-приложения, которая заключается в том, что разрабатываемый веб-сайт должен отображаться и функционировать во всех часто используемых 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 и новее);
* Firefox (версия 90.0.2 и новее);
* Microsoft Edge (версия 90.0.818.39 и новее).
  + 1. 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 достаёт из базы данных информацию, нужную для выполнения конкретных запросов. Представление определяет результат запроса, который получает пользователь. Для связи клиент-сервер следует реализовать REST API.

* + 1. Требования к используемым технологиям

При разработке Frontend части приложения будут использованы следующие технологии:

* JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript. Именно он исполняется браузерными движками и позволяет внедрять в сайт различные функции на стороне клиента;
* TypeScript – язык программирования, представленный Microsoft и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript, самый популярный сценарий использования - введение в код на этапе разработки четкую систему типов, которая позволяет избежать различных ошибок в типизации при компиляции в JS. Является обратно совместимым с JS и компилируется в него же. Упрощает поддержку и масштабирование JS приложений со стороны разработчика;
* React – библиотека с открытым исходным кодом, позволяющая упростить создание одностраничных приложений, работает как с JS, так и с TS. Поддерживает декларативный подход, который помогает сделать код более предсказуемым и упрощает дальнейшую отладку приложения.

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 эффективных, масштабируемых серверных приложений на Node.js. Сам фреймворк построен с использованием TypeScript и полностью поддерживает его (но при этом позволяет разработчикам использовать чистый JavaScript, что добавляет ему гибкости). Сочетает в себе элементы объектно-ориентированного программирования, функционального программирования и функционально-реактивного программирования. Позволяет реализовать MVC архитектуру;
* Приложение будет оперировать реляционной БД, в качестве СУБД будет использоваться PostgreSQL в силу своей бесплатной модели распространения, а также широкой поддержки как со стороны серверов, так и со стороны сообщества;
* В качестве инструмента развертки приложения будет использоваться Docker, который позволяет автоматизировать процесс развертывания и управления приложениями в средах с поддержкой системы контейнеризации;
* Для документации и упрощения понимания особенностей 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 удобно разрабатывать приложение, а также 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. Имеет широкий спектр возможности, от предоставления функционала удаленного репозитория до настройки процесса CI/CD при помощи GitHub Actions;
* Miro – платформа для визуализации гипотез и моделей, предоставляющая возможности для совместной работы распределенных команд;
* Trello – облачная программа для управления проектами небольших групп, поддерживает методологию управления канбан, позволяет наглядно отслеживать задачи каждого участника разработки;
* Figma – онлайн-сервис для разработки и прототипирования интерфейсов с возможностью организации совместной работы в режиме реального времени.
  + 1. Требования к численности и квалификации персонала, обслуживающего приложение

В случае необходимости поддержки (например, исправления ошибок или актуализации информации по финансовой грамотности в базе данных) или развития приложения (внедрения новых функций):

* Один или более разработчиков со знаниями PostgreSQL, TypeScript и Nest.JS для работы с базой данных и серверной части приложения;
* Один или более разработчиков со знаниями TypeScript и React.
  1. Требования к функциям, выполняемым веб-приложением

Разрабатываемое веб-приложение должно обладать следующими функциональными возможностями:

* Возможность регистрации и авторизации для пользователей;
* Добавление и редактирование записей расходов и доходов по каждой категории;
* Просмотр статистики расходов и доходов по каждой категории за определенный период;
* Создание и редактирование бюджетных авторских категорий трат и доходов (для премиум-пользователя);
* Реализация возможности приглашения друга для ведения совместного бюджета;
* Предоставление пользователю информации о повышении финансовой грамотности в виде всплывающих подсказок.

1. Языковые версии веб-приложения

На данном этапе разработки предусмотрена только русскоязычная версия веб-приложения.

1. Группы пользователей

В системе существуют такие группы пользователей как: неавторизованный и авторизованный, приглашенный, премиум-пользователь, хозяин совместного бюджета (группы) и участник группы.

Неавторизованный пользователь (гость) – посетитель веб-сайта, узнавший о сервисе из поисковой выдачи или любым другим способом.

Для неавторизованного пользователя должна быть реализована следующая функциональность:

* Возможность регистрации в веб-приложении;
* Возможность входа в демо-режим, в котором доступна ограниченная функциональность веб-приложения, а именно: просмотр трат, внесенных заранее (данные траты фактически являются предписанными неизменяемыми данными), возможность фильтрации таких трат.

Авторизованный пользователь – пользователь, прошедший процесс регистрации или авторизации.

Для авторизованного пользователя должна быть реализована следующая функциональность:

* Возможность приглашения одного друга для ведения совместного бюджета;
* Возможность редактирования и добавления трат;
* Возможность оформления премиум подписки;
* Возможность просмотра статистики трат;
* Возможность фильтрации трат по категориям;
* Возможность отключения подсказок по финансовой грамотности;
* Возможность выхода из аккаунта.

Приглашенный пользователь – пользователь, который перешел по ссылке-приглашению и после регистрации/авторизации на сервисе был автоматически добавлен в совместный бюджет к пригласившему его пользователю.

Для приглашенного пользователя должна быть реализована следующая функциональность:

* Возможность регистрации или авторизации;
* Возможность редактирования и добавления трат;
* Возможность оформления премиум подписки;
* Возможность просмотра статистики трат;
* Возможность фильтрации трат по категориям.

Премиум-пользователь – одна из возможных ролей, характерная для авторизированного пользователя, который приобрел подписку на сервис, открывающую дополнительные функциональные возможности.

Для премиум-пользователя должна быть реализована следующая функциональность:

* Возможность приглашения до четырех пользователей для ведения совместного бюджета;
* Возможность добавления авторской категории трат;
* Возможность редактирования и добавления трат;
* Возможность оформления премиум подписки;
* Возможность просмотра статистики трат;
* Возможность фильтрации трат по категориям.
* Возможность отключения подсказок по финансовой грамотности.

Пользователь хозяин совместного бюджета (группы) – одна из возможных ролей, которая может быть у авторизированного пользователя после того, как по его ссылке-приглашению перешли другие участники группы ведения совместного бюджета.

Такой пользователь обладает следующей функциональностью:

* Возможность приглашения новых пользователей в группу (при условии наличия премиум подписки);
* Возможность удаления пользователя из группы;
* Возможность добавления трат в общий бюджет;
* Возможность добавления трат в личный бюджет. Траты в личном бюджете являются приватными, то есть они не видны другим пользователям группы;
* Возможность редактирования трат;
* Возможность оформления премиум подписки;
* Возможность просмотра статистики трат;
* Возможность фильтрации трат по категориям;
* Возможность отключения подсказок по финансовой грамотности.

Пользователь участник группы – одна из возможных ролей, которая может быть у авторизированного пользователя после того, как его пригласили в группу для ведения бюджета. Стоит отметить, что пользователь может быть участником только одной группы, а также он не обладает правами добавления или удаления других участников группы.

Данному пользователю доступна следующая функциональность:

* Возможность добавления трат в общий бюджет, но не имеет возможности приглашения друзей в общий бюджет;
* Возможность добавления трат в личный бюджет. Траты в личном бюджете являются приватными, то есть они не видны другим пользователям группы;
* Возможность просмотра статистики трат;
* Возможность фильтрации трат по категориям;
* Возможность отключения подсказок по финансовой грамотности;
* Возможность выхода из группы.

1. 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), допольнительные акцентные (#59345C, #A17BA4). Допускается использование градиентов и теней, выдержанных в общей палитре.

1. Навигация по веб-приложению

Данное 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».

1. Описание страниц сайта
   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.

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.

* 1. Лэндинг

Приветственная страница сервиса, на которой присутствуют кнопки «Войти» либо «Зарегистрироваться», а также кнопка, позволяющая войти в демо-режим.

* 1. 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.

* + 1. Окно добавления трат и поступлений

В данном окне можно добавить трату или поступления, выбрать категорию, дату и время, сумму траты и описание.

* + 1. Окно просмотра статистики

В верхней части окна находится header с возможностью перейти в меню или вернуться на домашнюю страницу. Также представлена статистика трат по категориям в текстовом виде и график трат и доходов.

* 1. Профиль

На данной странице содержатся:

* Настройка финансовых подсказок;
* Элемент управления общим бюджетом, а именно добавления новых участников группы или исключение текущих.
  1. Боковое меню

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

1. Перспективы развития, модернизации веб-приложения

Развитие демо-режима путем внедрения возможности добавлять пользовательские даты, а не только просматривать предзаписанные.

Добавление функции установки ограничения для конкретной категории трат, выбранной пользователем. Если лимит превышен система сигнализирует об этом всплывающим сообщением.

Добавление другой цветовой темы в приложение.

Переход от резиновой верстки к адаптивной, локализация на иные языки (прежде всего английский), добавление поддержки мультивалютности, интеграция с банковскими API.

1. Требования к документированию

Документирование системы ведется в рамках Технического Задания в соответствие 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 серверной части приложения обеспечивается с помощью инструмента Swagger.

1. 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;
* Демонстративную версию проекта со всеми ключевыми сценариями;
* Курсовой проект;
* Видео-презентацию работы приложения;
* Исходный код системы.

Приложение

Диаграмма прецедентов:

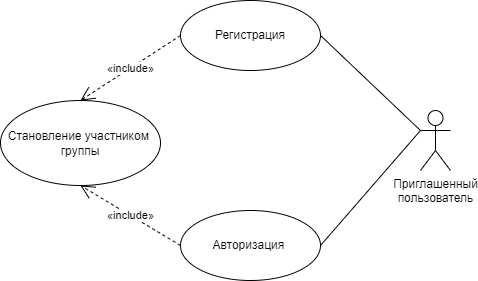
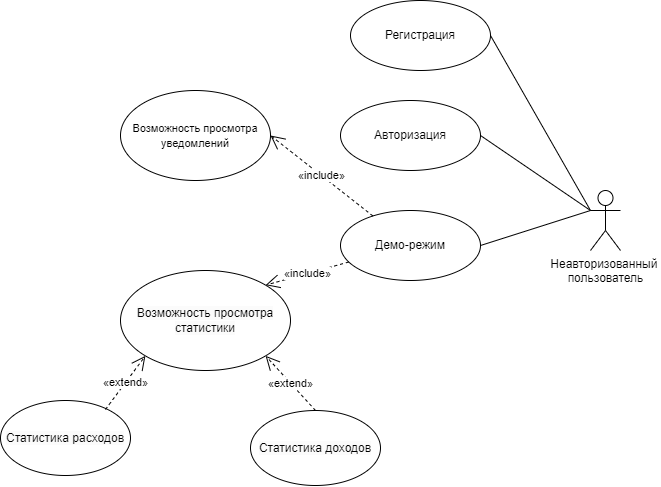
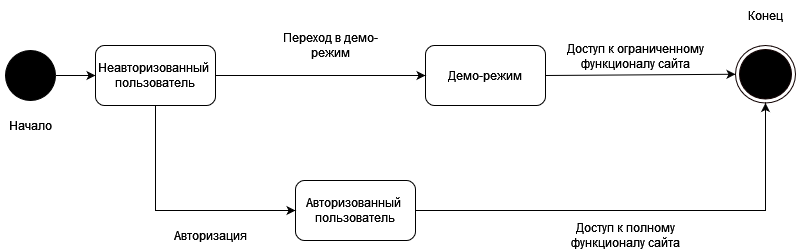


Диаграмма последовательностей:

Диаграмма состояний:



Контекстная диаграмма (IDEF0):

Диаграмма сотрудничества:

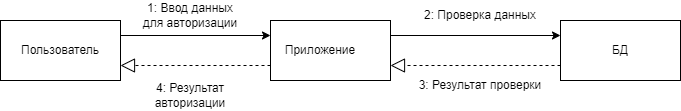
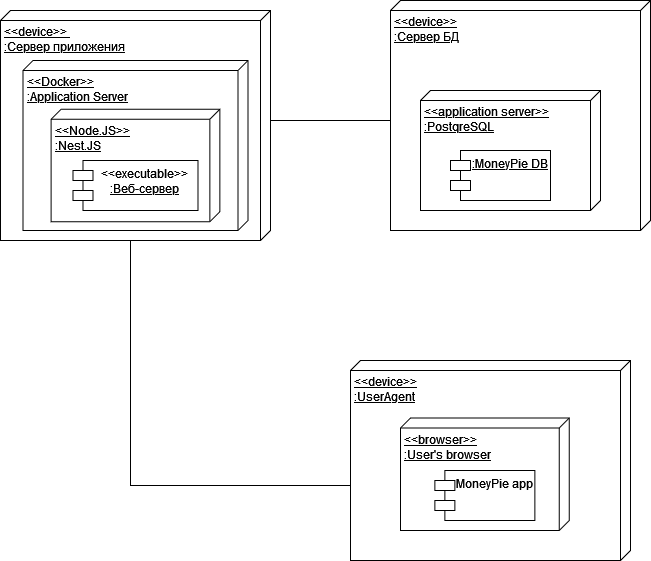


Диаграмма развертывания:



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 |