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

Факультет компьютерных наук

Кафедра информационных технологий управления

Курсовая работа

Веб-приложение для ведения домашнего бюджета

Направление 09.03.02 «Информационные системы и технологии»

Профиль «Информационные технологии управления»

Преподаватель _____ В.С. Тарасов, ст. преподаватель

Обучающийся _____А.А. Пустовалов, 3 курс, д/о

Обучающийся _____В.Г. Новиков, 3 курс, д/о

Обучающийся _____Е.О. Бордюжа, 3 курс, д/о

Воронеж 2023

Содержание

Введение

Ведение личного и совместного бюджета является важной повседневной задачей для многих людей, которые хотят контролировать свои расходы и вести учет своих доходов. В настоящее время существует множество приложений и сервисов для управления бюджетом, но часто они либо не удовлетворяют всем потребностям пользователей, либо не позволяют контролировать совместный бюджет нескольким пользователям.

В данной курсовой работе рассмотрен процесс разработки веб-приложения для мониторинга личного и совместного бюджета, которое будет условно бесплатным и удобным в использовании. Для реализации данного приложения будут использованы современные технологии веб-разработки, такие как React, Node.js, PostgreSQL.

Целью данной курсовой работы является практическое применение знаний и навыков веб-разработки для создания полноценного коммерческого приложения для ведения бюджета, которое будет полезно для широкого круга пользователей. Работа будет содержать детальное описание всех этапов процесса разработки, включая анализ требований, проектирование приложения, реализацию, тестирование.

1 Постановка задачи

Задачей данной курсовой работы является разработка веб-приложения для ведения домашнего бюджета. Онлайн-сервис позволит систематизировано вести учет расходов и доходов, отслеживать статистику по каждой категории трат.

Следует реализовать следующие пункты:

- Регистрация нового пользователя;
- Авторизация;
- Формирование и просмотр статистики по расходам;
- Возможность добавления пользователя для ведения совместного бюджета;
- Запись трат и входящих денежных средств по категориям.

2 Анализ предметной области

2.1 Глоссарий

- 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;
- 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;
- СУБД – это система управления базы данных;

- Фреймворк – это программная среда, облегчающая разработку и объединение разных компонентов большого программного проекта;
- 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;
- SQL-запросы – это наборы команд для работы с реляционными (табличными) базами данных;
- SQL-инъекция – это один из способов взлома сайта.

2.2 Сценарии пользователей

1. Пользователь: Александр, 32 года.

   Описание: имеет постоянный доход, занимается бизнесом и желает отслеживать свои расходы.

   Пользовательская история: Александр хочет вести учет своих расходов на личные нужды и бизнес, чтобы определить, какие

категории занимают большую часть его бюджета и где можно сэкономить.

2. Пользователь: Анастасия, 26 лет.

   Описание: студентка, живет в общежитии, получает стипендию, не имеет постоянного дохода.

   Пользовательская история: Анастасия хочет контролировать траты, чтобы не истощить свою стипендию слишком быстро и иметь возможность купить то, что ей действительно нужно.

3. Пользователь: Дмитрий, 40 лет.

   Описание: имеет постоянный доход, семейный человек.

   Пользовательская история: Дмитрий и его жена хотят вести совместный бюджет, отслеживать свои расходы и доходы, чтобы узнать, сколько они тратят на различные категории и насколько могут сэкономить. Они хотят иметь возможность просмотреть свою статистику расходов за месяц и планировать свой бюджет на будущее.

2.3 Цели создания веб-приложения

Цель приложения – предоставить пользователю функциональность, которая позволит отслеживать движение личных денежных средств, просматривать статистику по расходам и доходам.

Для достижения данной цели были выделены следующие задачи:

- Проектирование и развертывание базы данных;
- Разработка Frontend части веб-приложения;
- Реализация бизнес-логики приложения на сервере;
- Реализация связи между клиентом и сервером с применением подхода REST API;
- Развертывание приложения на серверной части.

2.4 Сфера применения

Приложения для ведения бюджета имеют большое значение для многих людей, поскольку позволяют им контролировать свои расходы и доходы, следить за своим финансовым положением и планировать свои расходы. Это может быть особенно полезно для тех, кто имеет ограниченный бюджет или хочет экономить на своих расходах.

Использование приложений для учета бюджета может также помочь людям научиться лучше управлять своими деньгами и развивать финансовую грамотность. Это может включать в себя понимание, как управлять своими расходами, как правильно планировать свои финансы и как принимать обоснованные финансовые решения.

Некоторые из наиболее распространенных сфер применения включают в себя:

- Личные финансы: приложения для учета бюджета могут помочь людям контролировать свои личные финансы, планировать свои траты, отслеживать свои доходы и расходы, а также устанавливать финансовые цели и следить за их выполнением;
- Семейный бюджет: приложения для учета бюджета могут помочь семьям планировать свой бюджет, учитывая расходы на питание, жилье, транспорт и другие категории трат;
- Финансовое планирование: приложения для учета бюджета могут помочь людям планировать свои финансовые цели и следить за их выполнением, а также помочь им принимать взвешенные финансовые решения;
- Обучение финансовой грамотности: приложения для учета бюджета могут быть полезны для обучения финансовой грамотности, а также помочь людям научиться планировать свои финансы и контролировать расходы.

2.5 Обзор аналогов

2.5.1 Мобильное приложение Monefy

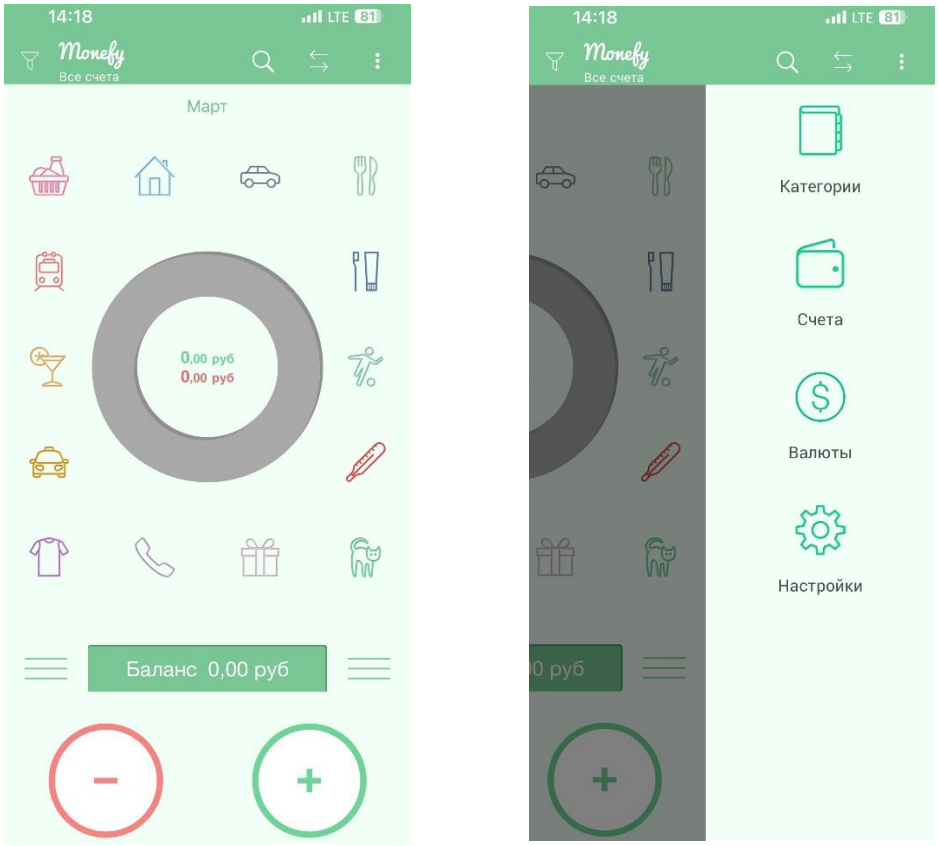

Рисунок 1 - Приложение Monefy

Сильные стороны приложения:

- Есть функция «совместного доступа», благодаря которой можно вести общий бюджет с разных устройств;
- Есть авторизация с паролем;
- Есть встроенный калькулятор для автоматического подсчета бюджета;
- Гибкие настройки: категории доходов и расходов, поддержка различных валют;
- Реализован выбор периода для анализа от дня до года;
- Есть возможность пользоваться приложением без авторизации.

Слабые стороны приложения:

- Приложение работает только на мобильных устройствах, нет веб-версии;
- Нет интеграции с приложениями банков;

– При отсутствии подписки: реклама, невозможность создавать новые пользовательские категории трат, нет темной темы;
– Нет системы уведомлений.

2.5.2 Приложение Toshl finance

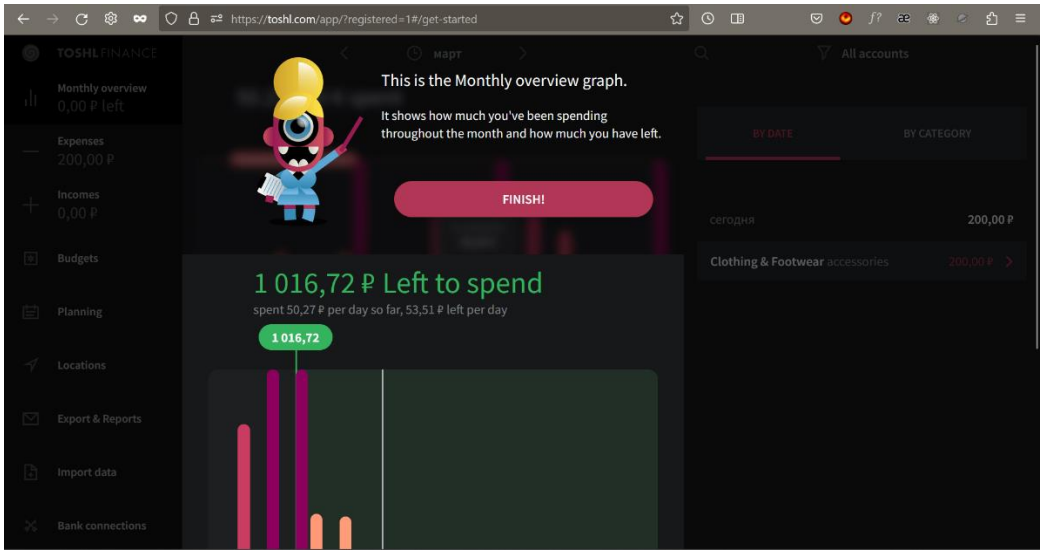

Рисунок 2 - Приложение Toshl finance

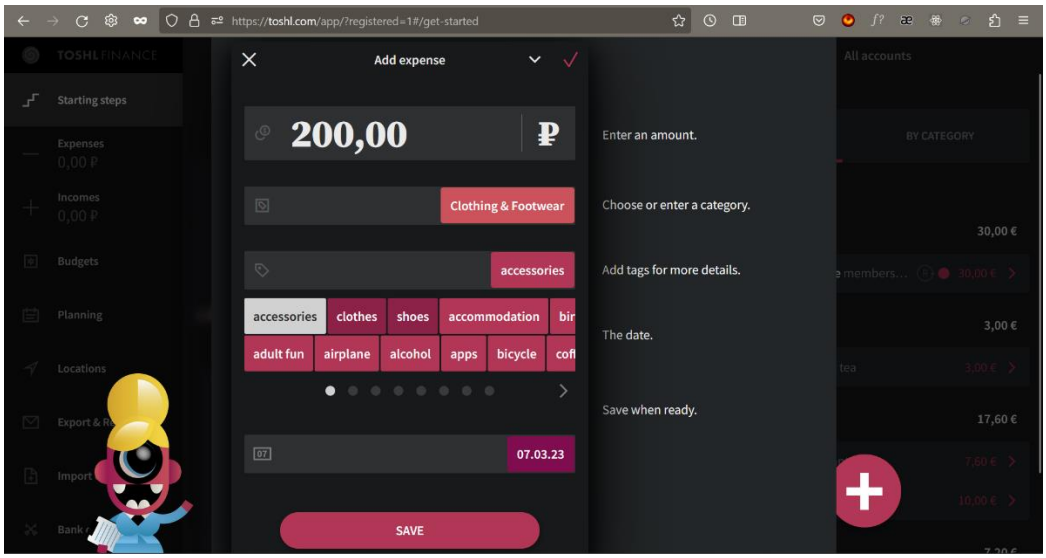

Рисунок 3 - Приложение Toshl finance

Сильные стороны приложения:

– Есть мобильная и веб-версия приложения;
– Поддержка множества видов валют;

– Реализованы информативные графики с историями доходов и расходов;
– Существует клиентская служба поддержи, а также есть возможность связаться с разработчиками и сообщить о возникших ошибках в работе приложения.

Слабые стороны приложения:

– Постоянно меняются языки с русского на английский и обратно в категориях и пометках, причем как в приложении, так и в веб версии;
– Приложение выдает сбои в обновлениях состояния карт и данных о тратах;
– При обновлениях приложения сбрасываются все настройки пользователя;
– Регулярная реклама о подписке;
– Нет общего бюджета;
– Нет возможности пользоваться приложением без авторизации.

2.5.3 Приложение Дзен-мани

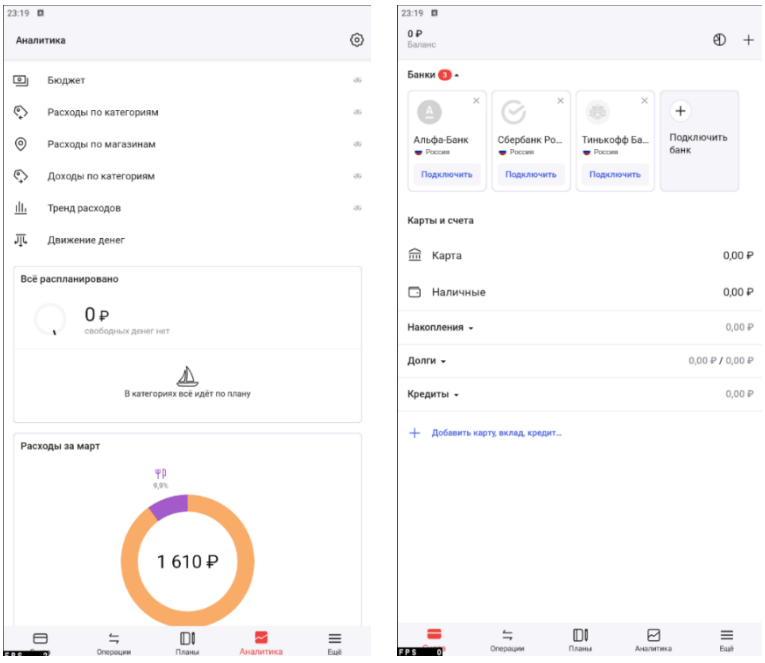

Рисунок 4 - Приложение Дзен-мани

Сильные стороны приложения:

– Есть мобильная и веб-версия приложения;
– Реализована система напоминаний;
– Есть общий бюджет;
– Есть возможность демо-входа.

Слабые стороны приложения:

– Веб-версия приложения работает не во всех браузерах корректно;
– Нет возможности пользоваться приложением без авторизации;
– Основные функции платные (например, функция аналитики за прошлый месяц);
– Нет возможности сгруппировать покупки по категориям.

2.5.4 Приложение Coinkeeper 3

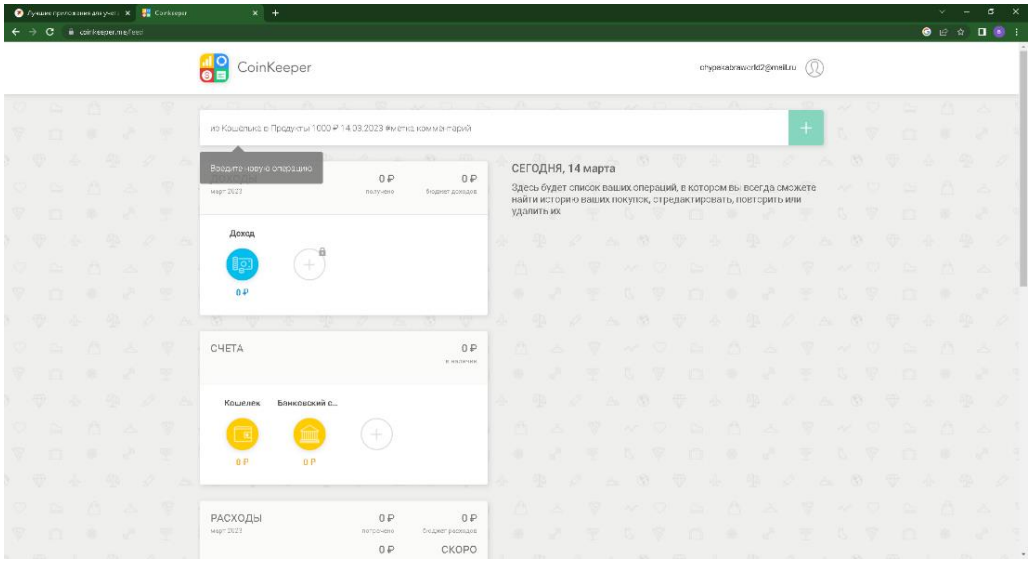

Рисунок 5 - Приложение Coinkeeper 3

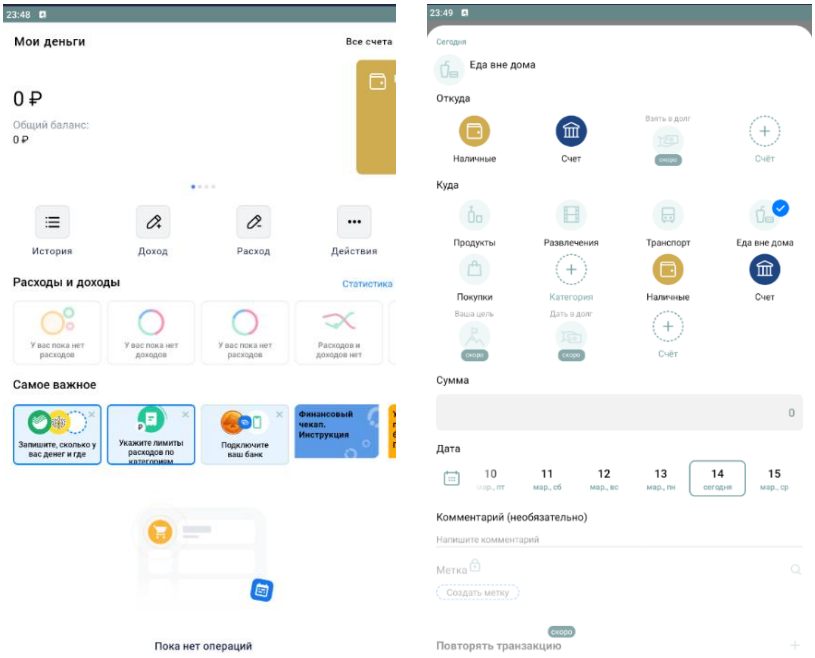

Рисунок 6 - Приложение Coinkeeper 3

Сильные стороны приложения:

- Есть мобильная и веб-версия приложения;
- Есть возможность пользоваться приложением без авторизации;
- Присутствуют курсы и инструменты финансовой грамотности.

Слабые стороны приложения:

- Дорогая платная подписка (по сравнению с другими рассматриваемыми сервисами);
- Ограниченная функциональность без подписки (нельзя добавить новые категории трат);
- Нет общего бюджета;
- Плохая техподдержка (многие пользователи жалуются, что происходит полное игнорирование со стороны разработчиков).

2.6 Требования к функциональности

Разрабатываемое веб-приложение должно обладать следующей функциональностью:

- 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.

2.7 Пользователи системы

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 – авторизированный 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-пользователь, 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.

3 Анализ задачи

3.1 Диаграмма прецедентов

Диаграмма прецедентов помогает описать, как пользователи будут взаимодействовать с системой. На Рисунке 7 представлена диаграмма прецедентов для авторизованного пользователя, а на Рисунке 8 – для неавторизованного пользователя.

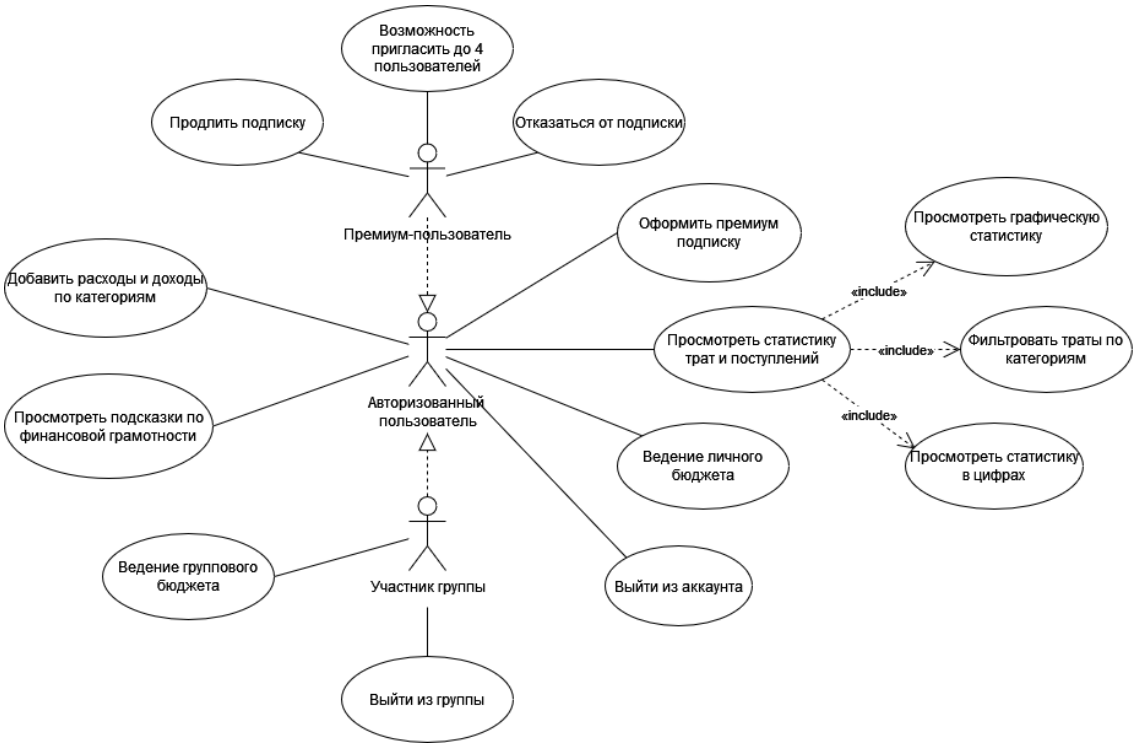

Рисунок 7 - Диаграмма прецедентов

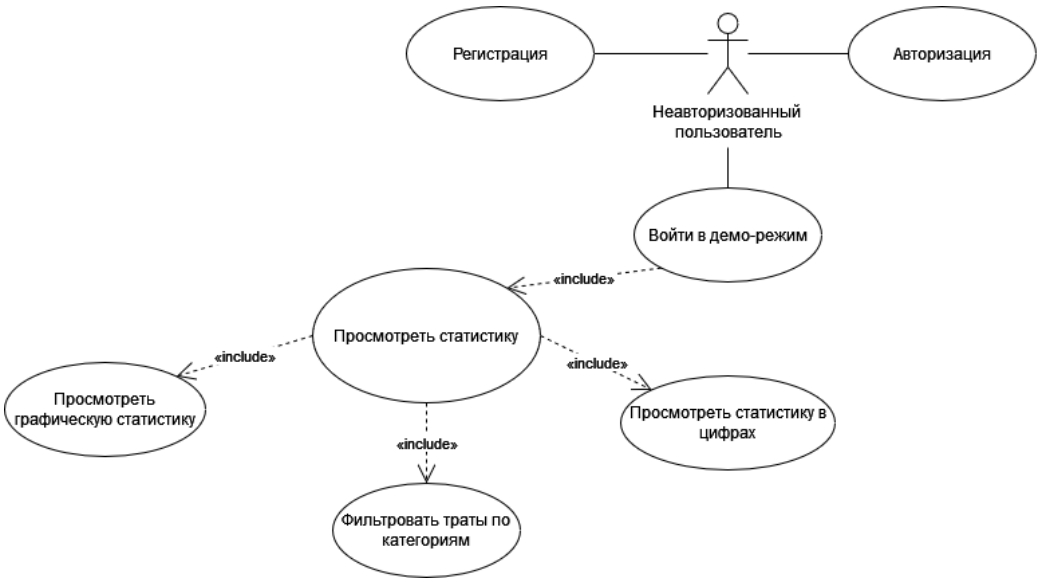

Рисунок 8 - Диаграмма прецедентов

3.2 Диаграмма последовательностей

Диаграмма последовательностей используется для уточнения диаграмм прецедентов, более детального описания логики сценариев использования. Диаграмма последовательности для неавторизованного пользователя представлена на Рисунке 9, а для авторизованного – на Рисунке 10.

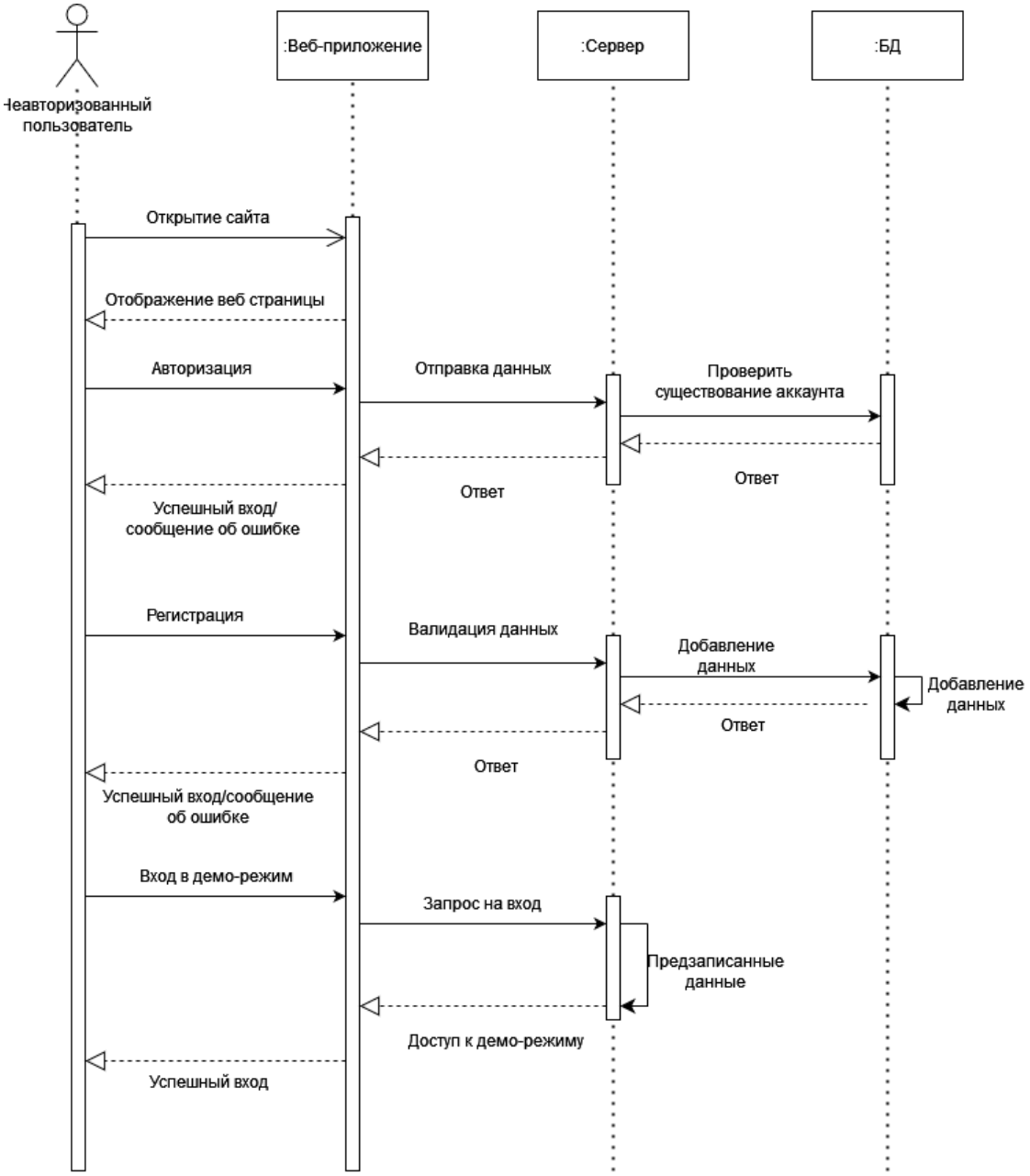

Рисунок 9 - Диаграмма последовательности

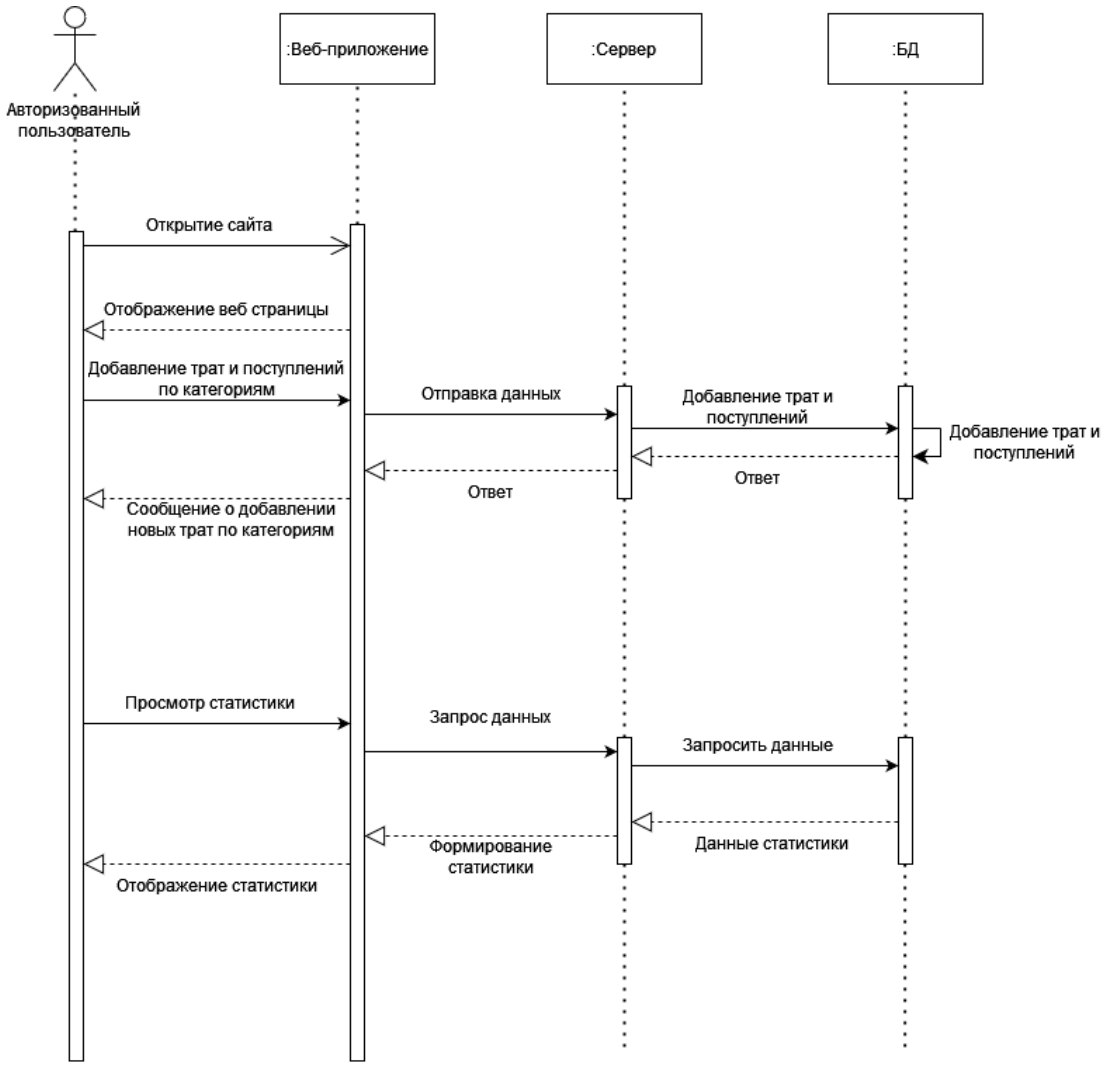

Рисунок 10 - Диаграмма последовательности

3.3 Диаграмма состояний

Диаграмма состояний показывает все возможные изменения в состоянии конкретного объекта на протяжении всего его жизненного цикла (Рисунок 11-12).

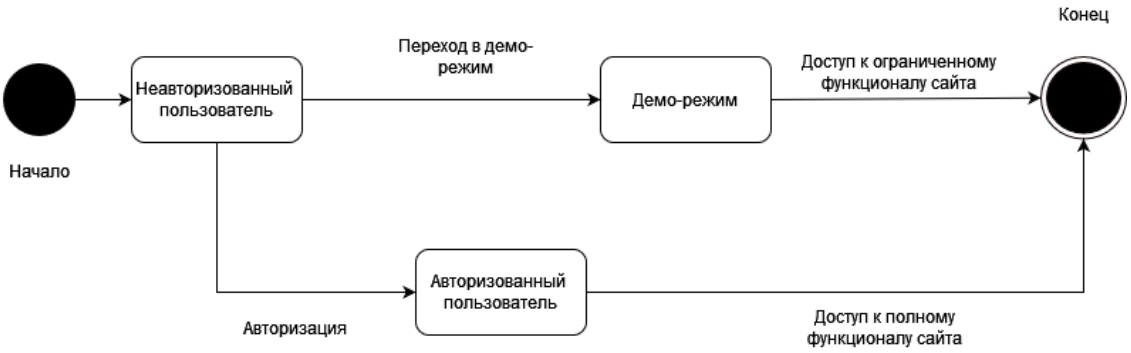

Рисунок 11 - Диаграмма состояния пользователя

Рисунок 12 - Диаграмма состояния группы

3.4 Контекстная диаграмма (IDEF0)

Диаграммы по методологии IDEF0 (Рисунок 13-14). Эта методология функционального моделирования и графическая нотация, которая предоставляет возможность описать бизнес-процессы и определить их структуру, что позволяет более детально показать функциональность проекта и разработать наиболее оптимальный план действий при его реализации.

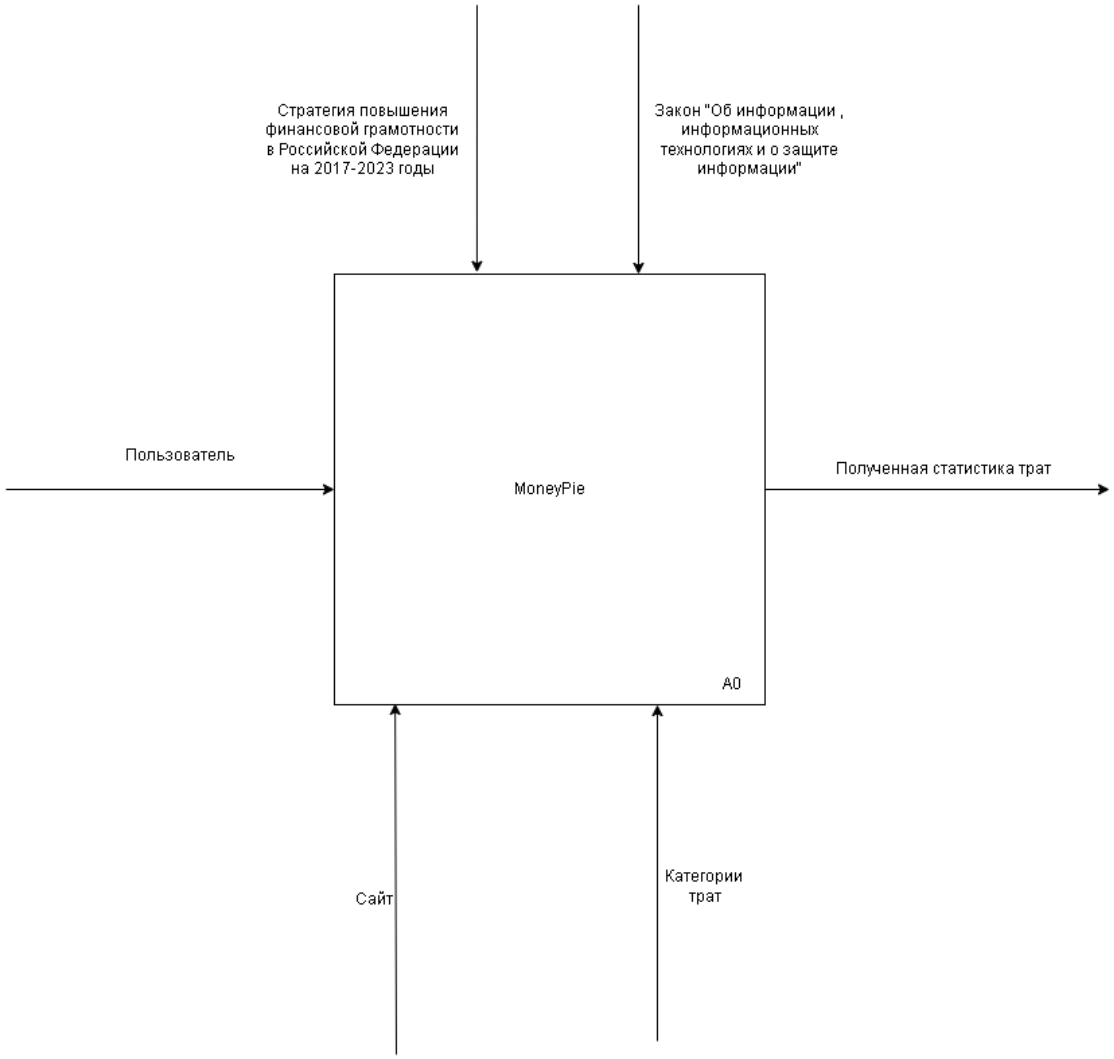

Рисунок 13 - IDEF0 уровень 0

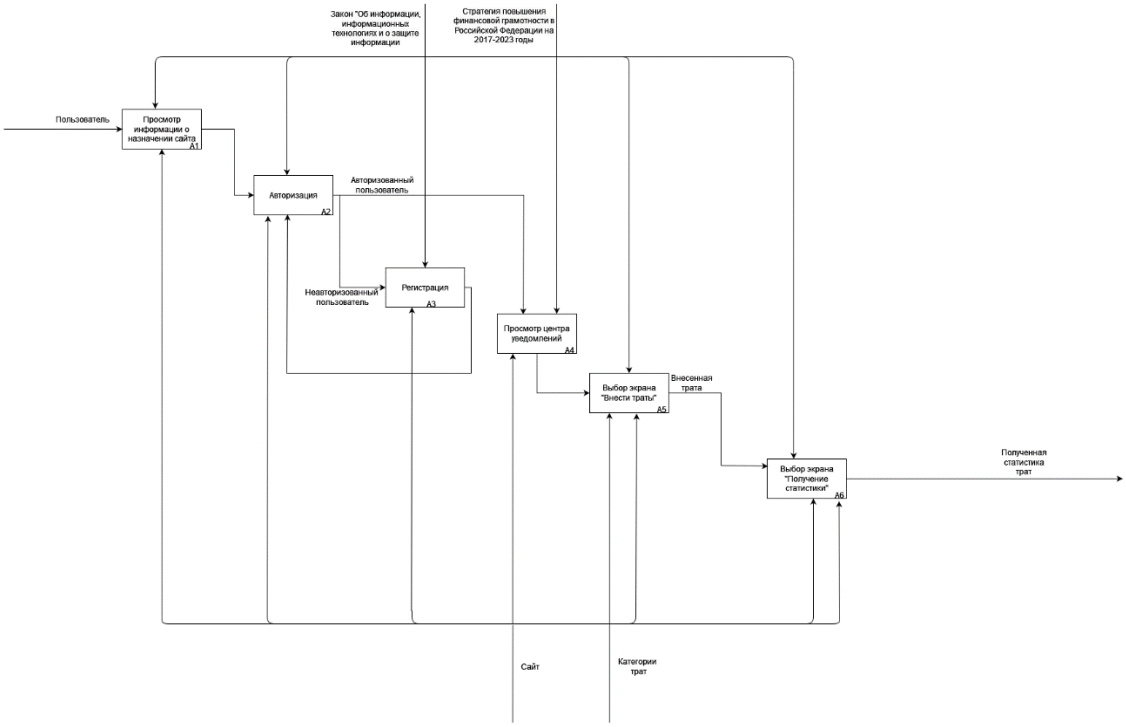

Рисунок 14 - IDEF0 уровень 1

3.5 Диаграмма сотрудничества

Диаграмма сотрудничества определяет набор взаимодействующих ролей, используемых вместе, чтобы показать функциональность приложения (Рисунок 15-17).

Рисунок 15 - Диаграмма сотрудничества

Рисунок 16 - Диаграмма сотрудничества

Рисунок 17 - Диаграмма сотрудничества

3.6 Диаграмма активностей

Диаграмма активностей отражает динамические аспекты поведения системы и наглядно показывает, как поток управления переходит от одной деятельности к другой (Рисунок 18).

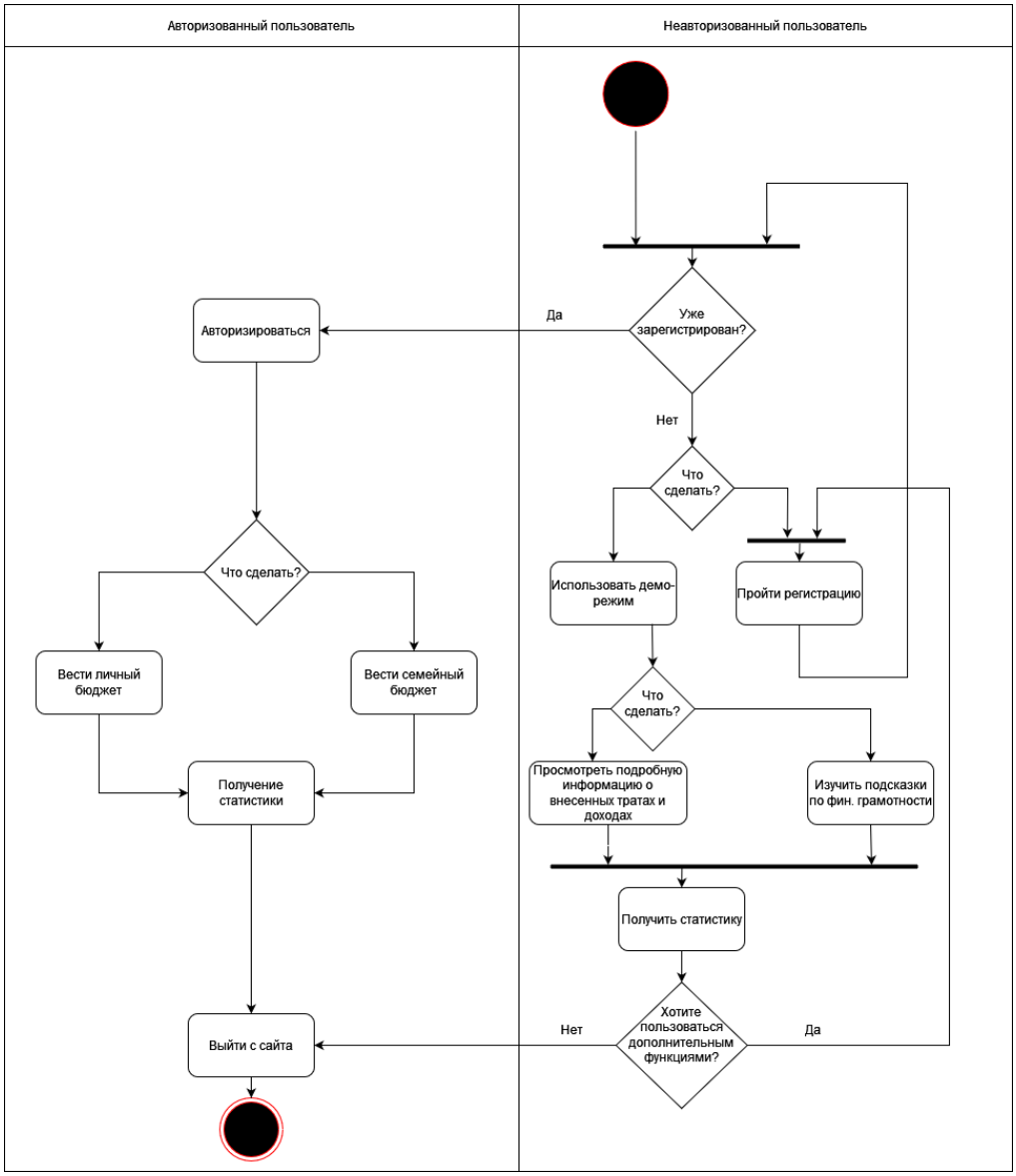

Рисунок 18 - Диаграмма активностей

3.7 Диаграмма развертывания

Диаграмма развертывания показывает архитектуру исполнения системы, включая такие узлы, как программные среды исполнения, а также промежуточное программное обеспечение, соединяющее их.

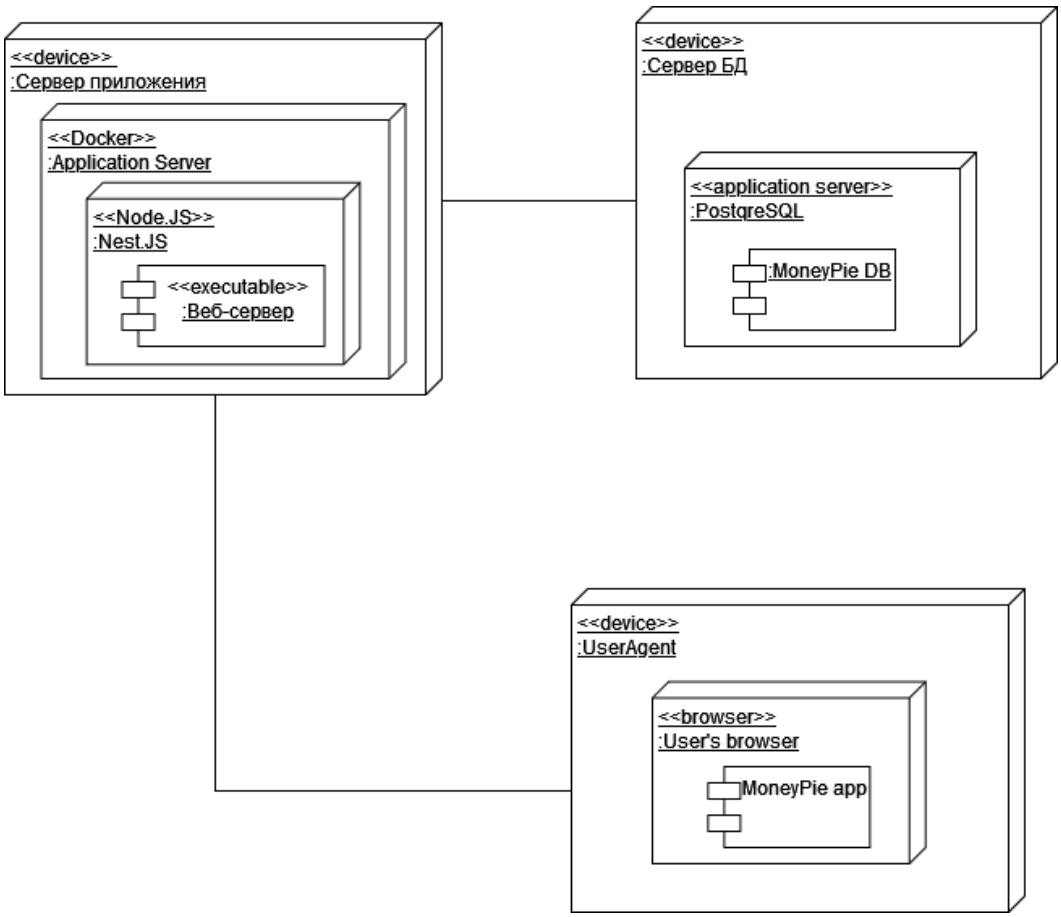

Рисунок 19 - Диаграмма развертывания

4 Реализация

4.1 Разработка frontend

При разработке клиентской 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.

4.2 Разработка backend

При разработке серверной 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.

5 Тестирование

Заключение

Список использованных источников