



**Министерство науки и высшего образования  
Российской Федерации Федеральное государственное  
бюджетное образовательное учреждение высшего  
образования «Московский государственный технический  
университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»**

**Дисциплина «Разработка интернет-приложений»**

**Расчетно-пояснительная записка**

**Тема: «Заявки на установку серверного ПО»**

**Студент: Ювенский Л.А.**

**Группа ИУ5-53Б**

**Преподаватель: Канев А.И.**

**Москва, 2024 г.**

## ВВЕДЕНИЕ

Серверы играют ключевую роль в современном мире, обеспечивая инфраструктуру для хранения данных, обработки информации и поддержания множества онлайн-сервисов. По данным Tadvise по итогам 2023 года объём рынка мощных серверов достиг \$20,4 млрд. Что на 11% больше по сравнению с результатом за 2022-й. В условиях растущего объема данных, облачных технологий и необходимости быстрого обмена информацией серверы становятся основой функционирования как крупных корпораций, так и малых предприятий. Особенно важным в этих обстоятельствах становится использование специализированных сервисов, помогающих настроить программное обеспечение на серверах. Эти сервисы играют ключевую роль в оптимизации процессов установки, конфигурации и мониторинга серверного ПО.

Целью данного проекта является разработка системы, предоставляющей интерфейс для учета заявок на установку серверного ПО, и включающей в себя веб-сервис, веб-приложение и десктопное приложение Tauri.

Данная система предназначена для клиентов, желающих разместить заказы на установку ПО, а также для сотрудников компании, занимающихся обработкой этих заказов. Клиентам предлагается возможность создания заявок, в которые они могут добавлять необходимое им ПО и управлять процессом заказа. Система предоставляет автоматизированный способ создания, отслеживания и управления заявками на установку ПО. Также она предоставляет сотрудникам компании возможность принимать или отклонять заявки на установку ПО.

Нефункциональные требования к разрабатываемой системе:

1. Должна поддерживаться кроссплатформенность.
2. Интерфейс системы и текст ошибок должны быть русифицированы.

В ходе работы необходимо выполнить следующие задачи:

1. Создать MVP и базового дизайна на основе [firstbyte.ru](http://firstbyte.ru);

2. Создать базу данных для хранения информации о ПО и заявках на установку;
3. Создать веб-сервис в бэкенде на Django (DRF);
4. Реализовать авторизацию и хранение сессий в Redis;
5. Разработать базовый SPA на React для гостя;
6. Внедрить адаптивность, менеджер состояний Redux Toolkit, PWA, разработать Tauri приложение;
7. Завершить разработку интерфейса пользователя в React, использовать для обращений к методам веб-сервиса Axios;
8. Реализовать интерфейс сотрудника поддержки в React;
9. Разработать десктопное приложение Tauri
10. Развернуть приложение при помощи GitHub Pages
11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.
12. Оформить git-репозиторий на сервисе GitHub, содержащий исходный код проекта.

# 1 ПРЕДМЕТНАЯ ОБЛАСТЬ

Серверы занимают важное место в современном мире, обеспечивая инфраструктуру для хранения данных и обработки информации. Не менее важен вопрос контроля процесса обновления ПО, запускаемого на серверах. Одним из наиболее популярных решений в области управления ПО, запускаемого на серверах, является система оркестровки контейнеров (container orchestration system) Kubernetes [3].

В своей работе «Распределенная трассировка для каскадных изменений объектов в плоскости управления Kubernetes» («Distributed Tracing for Cascading Changes of Objects in the Kubernetes Control Plane») Томоюки Эхира (Tomoyuki Ehira), Дайсуке Котани (Daisuke Kotani) и Ясую Окабе (Yasuo Okabe) рассматривают вопрос мониторинга каскадных изменений (cascading change) объектов Kubernetes [3]. Не существует практического способа наблюдать за каскадным изменением, в том числе, иметь разбивку по времени, затраченном на каждое изменение.

Методы распределённой трассировки (Distributed tracing techniques) обычно применяются для мониторинга микросервисов (microservices), однако неприменимы напрямую в Kubernetes.

В статье [3] авторы описывают разработанную ими систему, которая позволяет автоматически отслеживать изменения объектов в плоскости управления (control plane). Описанный метод решения задачи включает добавление идентификатора распространения изменений (Change Propagation ID, CPID) к метаданным объекта и контроллер, который наблюдает за изменениями объекта.

Разработанная система позволяет визуализировать изменения и отображать их операторам кластера (cluster operators). Накладные расходы системы незначительны.

Компания, предоставляющая услуги установки ПО имеет определенный перечень программ, доступных для установки и настройки [2]. ПО доступно

для установки средствами ОС сервера или для развертывания в системе оркестровки контейнеров (container orchestration system) Kubernetes [3]. Во втором случае установкой и настройкой ПО занимается оператор кластера (cluster operator) [3], который развернёт (deploy) на сервере клиента всё необходимое ПО и проконтролирует ключевые показатели (key metrics).

Пользователь (user), являющийся клиентом (client) компании, может выбрать ПО (software) для установки или развертывания (deploy), указать версию каждой программы и отправить заявку (request) на рассмотрение оператора кластера (cluster operator). Каждая заявка (request) в начале создается как черновик, в который можно добавлять и из которого можно удалять ПО. Одна заявка может содержать несколько программ; тогда, в случае согласования заявки, будет оформлен заказ на установку всего ПО из списка. Для каждого ПО в заявке необходимо указать требуемую для установки версию.

Клиент может посмотреть список всех созданных им заявок. В случае, если заявку составлена корректно, менеджер утверждает её. В противном случае – отклоняет. Менеджер может только либо согласовать, либо отклонить заявку.

Менеджеры компании, предоставляющей услуги установки ПО, могут редактировать список программ, доступных для установки: добавлять новое ПО, удалять устаревшее, а также редактировать информацию о каждой отдельно взятой программе. Менеджеры имеют возможность просматривать все заявки, созданные пользователями, а также фильтровать их по имени клиента, дате формирования и статусу. После того, как заявка была согласована или отклонена, менеджер не может изменять её статус.

Функции пользователей с различными ролями описаны на диаграмме прецедентов (рисунок 1).

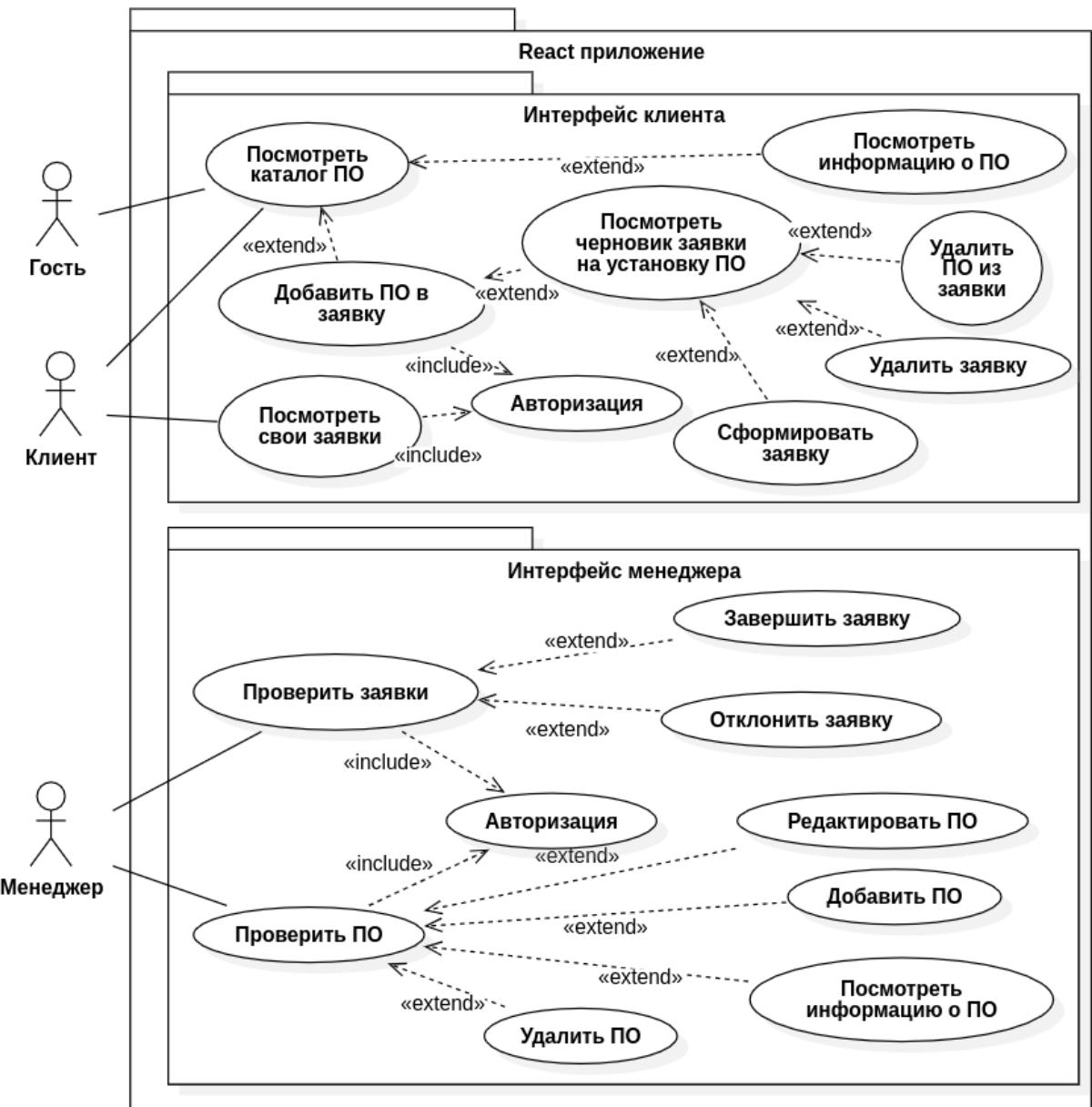


Рисунок 1 – Диаграмма прецедентов

Гостям доступен просмотр списка доступного для установки ПО. Гости, прошедшие этап регистрации, являются клиентами. Клиенты могут добавлять ПО в заявку на установку, формировать заявку из черновика, а также просматривать список своих заявок.

Клиент выбирает ПО для установки, затем на основе выбранного списка ПО формирует заявку. После чего менеджер может просмотреть её, а также согласовать или отклонить. В случае некорректного составления заявки клиент имеет возможность удалить заявку. Возможные состояния статуса заявки отображены на диаграмме состояний (рисунок 2)

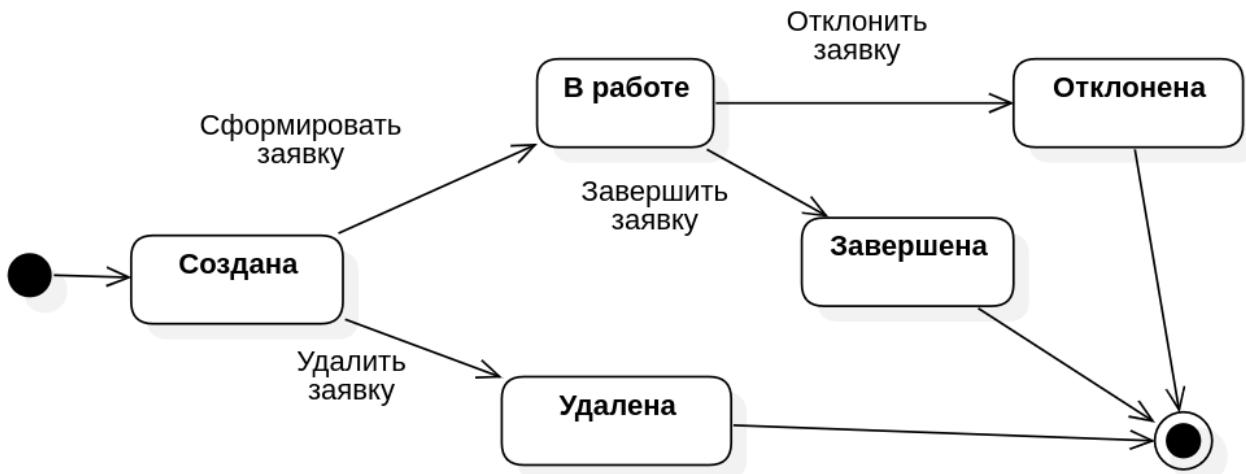


Рисунок 2 – Диаграмма состояний заявок

Заявки обрабатываются менеджерами компании. В результате обработки заявки её либо одобряют, либо отклоняют. Менеджеру доступны операции для работы с ПО: просмотр всех программ, редактирование, создание и удаление программ, а также просмотр списка всех программ табличном виде.

Процесс оформления заявки отражен на диаграмме деятельности (рисунок 3).

В начале взаимодействия с системой оформления заявок на установку ПО клиент запрашивает список доступного для установки ПО. При необходимости клиент может добавить конкретное ПО в текущую черновую заявку. После чего может продолжить выбор ПО из каталога или перейти к оформлению заявки. В случае, если клиент выбрал оформление заявки на установку ПО, ему необходимо указать версии для всех программ в списке. После чего указать адрес хоста (сервера), на который необходимо произвести установку ПО из списка. По нажатии кнопки «Сформировать заявку» заявка формируется и становится доступна в интерфейсе менеджера.

Менеджер имеет возможность просматривать заявки пользователей, а также принимать решение об одобрении заявки, если она может быть выполнена компанией или отклонении заявки, если она не может быть выполнена компанией.

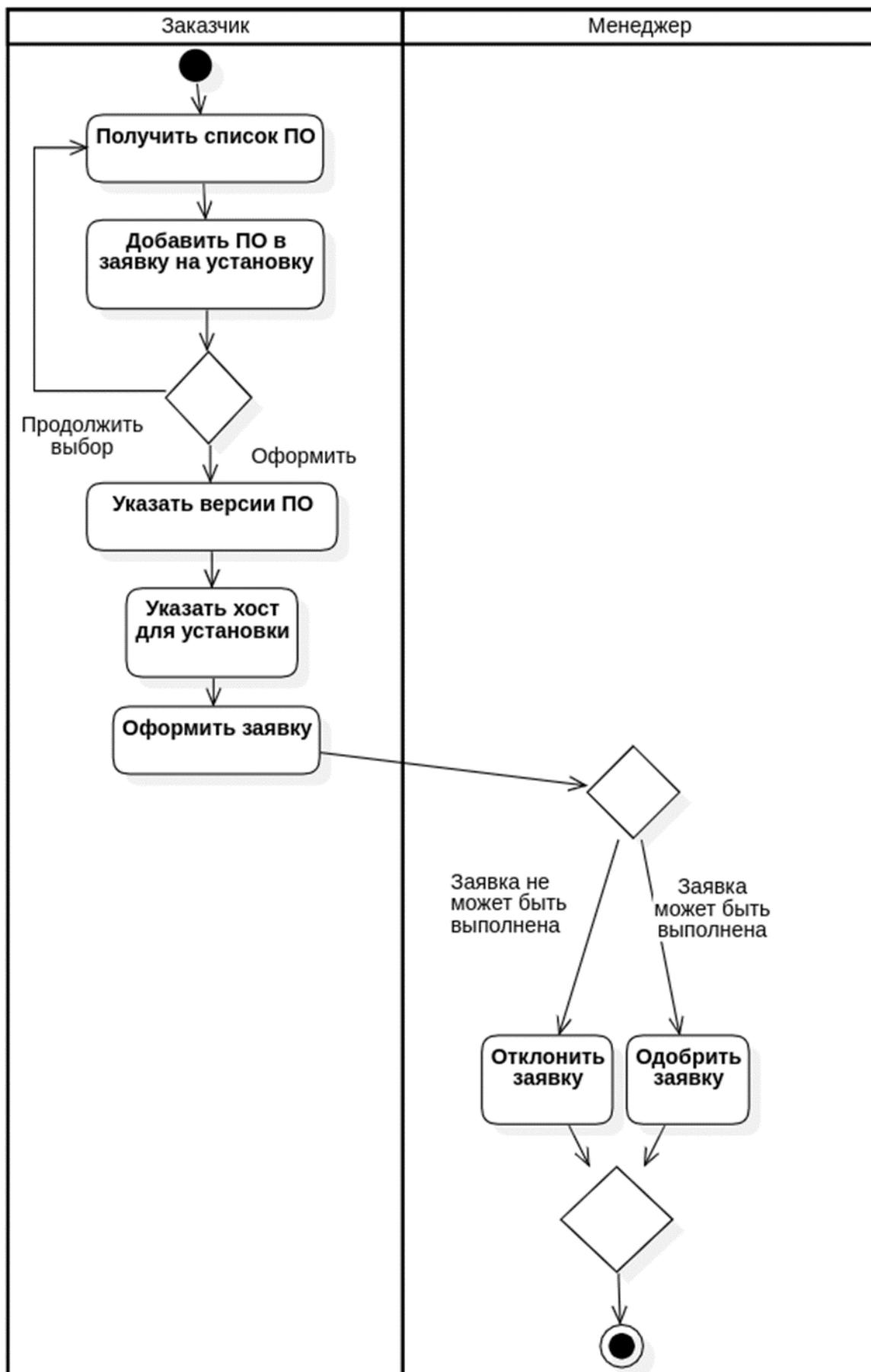


Рисунок 3 – Диаграмма деятельности

## 2 АРХИТЕКТУРА

Архитектура системы отображена на диаграмме развертывания (рисунок 4).

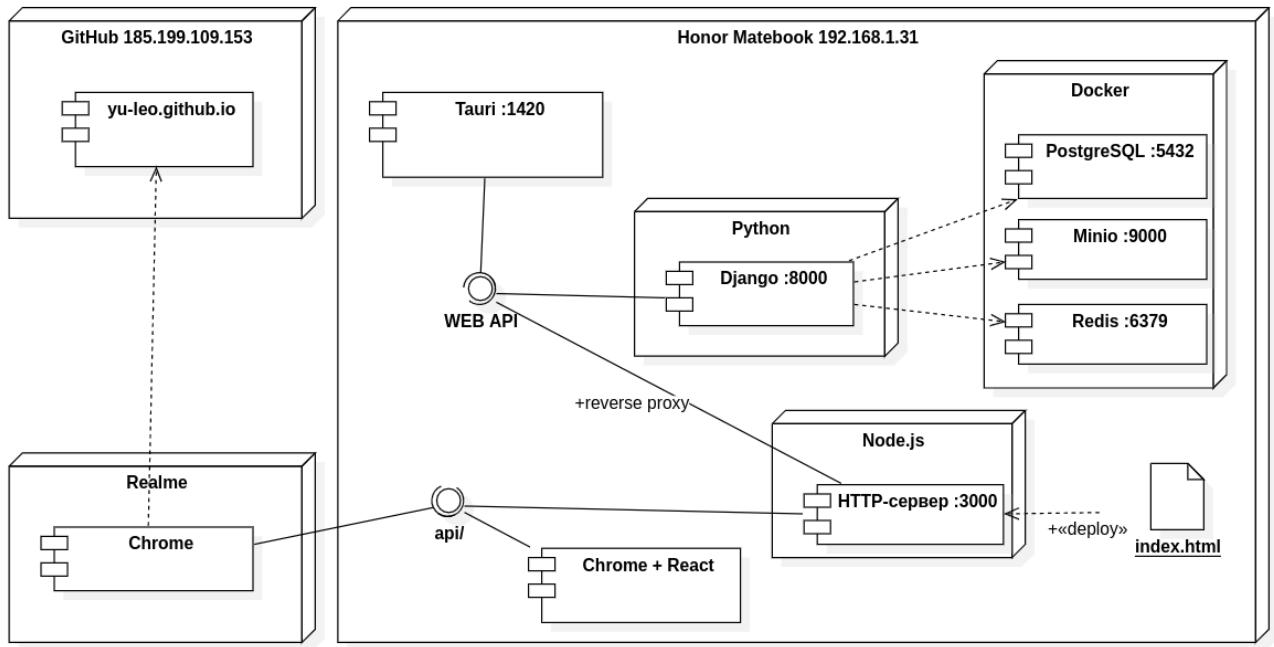


Рисунок 4 – Диаграмма развертывания

Веб-сервис, реализованный на фреймворке Django [4] с использованием DRF [5] связан с серверами Minio [6], Redis [7] и СУБД PostgreSQL [8]. В Redis хранятся активные сессии пользователей; при выходе пользователя из системы – удаляются.

Язык программирования Python [9] был выбран в силу его простоты, а также популярности. Использование Django обусловлено тем, что этот фреймворк имеет достаточно широкий возможности и является стандартом индустрии. Веб-сервис на Django является общим для веб-сервера и десктопного приложения Tauri [10].

Данные хранятся в СУБД PostgreSQL. Она была выбрана, поскольку, наравне с Python и Django она является стандартом современной индустрии разработки.

Структура данных отражена на ER диаграмме (рисунок 5). Модель программ (услуг) представляет собой набор полей, отражающих свойства ПО.

Данные о ПО хранятся в таблице software. Для хранения в одной заявке на установку ПО нескольких программ используется промежуточная таблица связи software\_in\_request, которая реализует связь М-М. Таблица install\_software\_requests представляет собой список заявок на установку ПО. Данные о пользователях системы – клиентах и менеджерах – хранятся в таблице auth\_user.

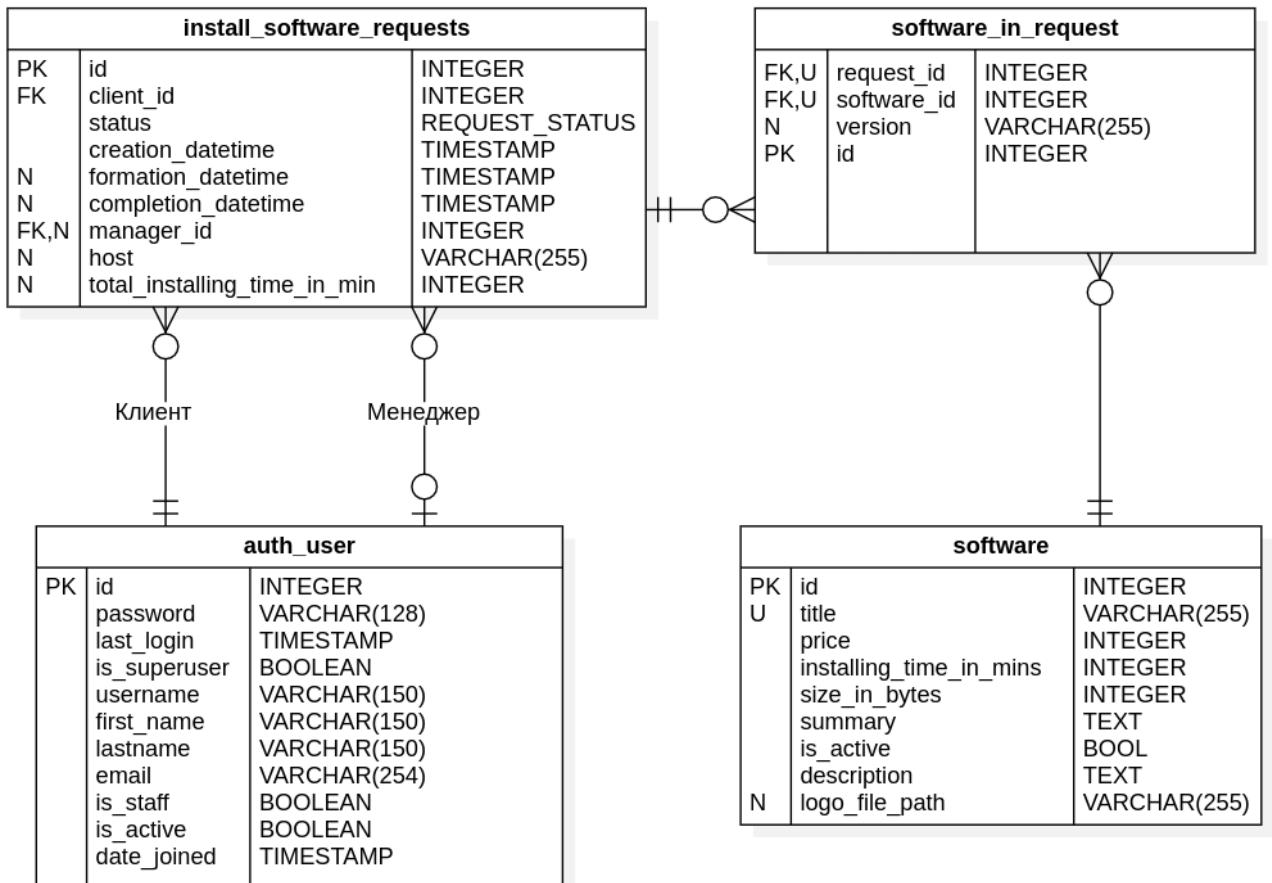


Рисунок 5 – ER-диаграмма

Веб-сервер реализован с использованием языка TypeScript и фреймворка React [11]. Устройство бэкенда приложения разработанной системы приведено на диаграмме классов бэкенда (рисунок 6). Пользователи взаимодействуют с доменами. Домены связаны с моделями. Модели имеют связи с таблицами в базе данных.

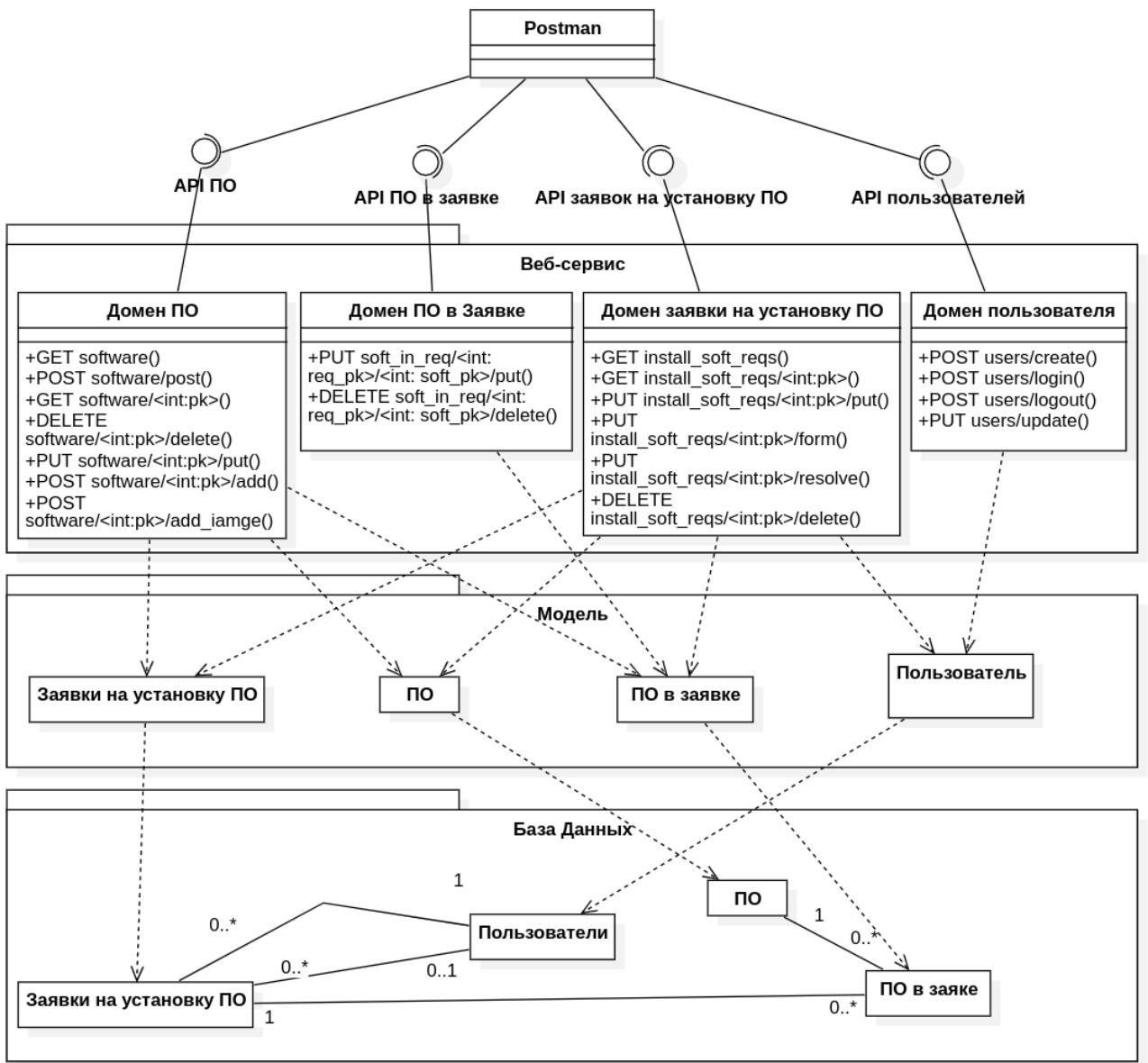


Рисунок 6 – Диаграмма классов бэкенда

Связь фронтенда и бэкенда отражена на диаграмме классов фронтенда (рисунок 7). Каждая страница связана с API, которое используется для взаимодействия с данными на соответствующей странице.

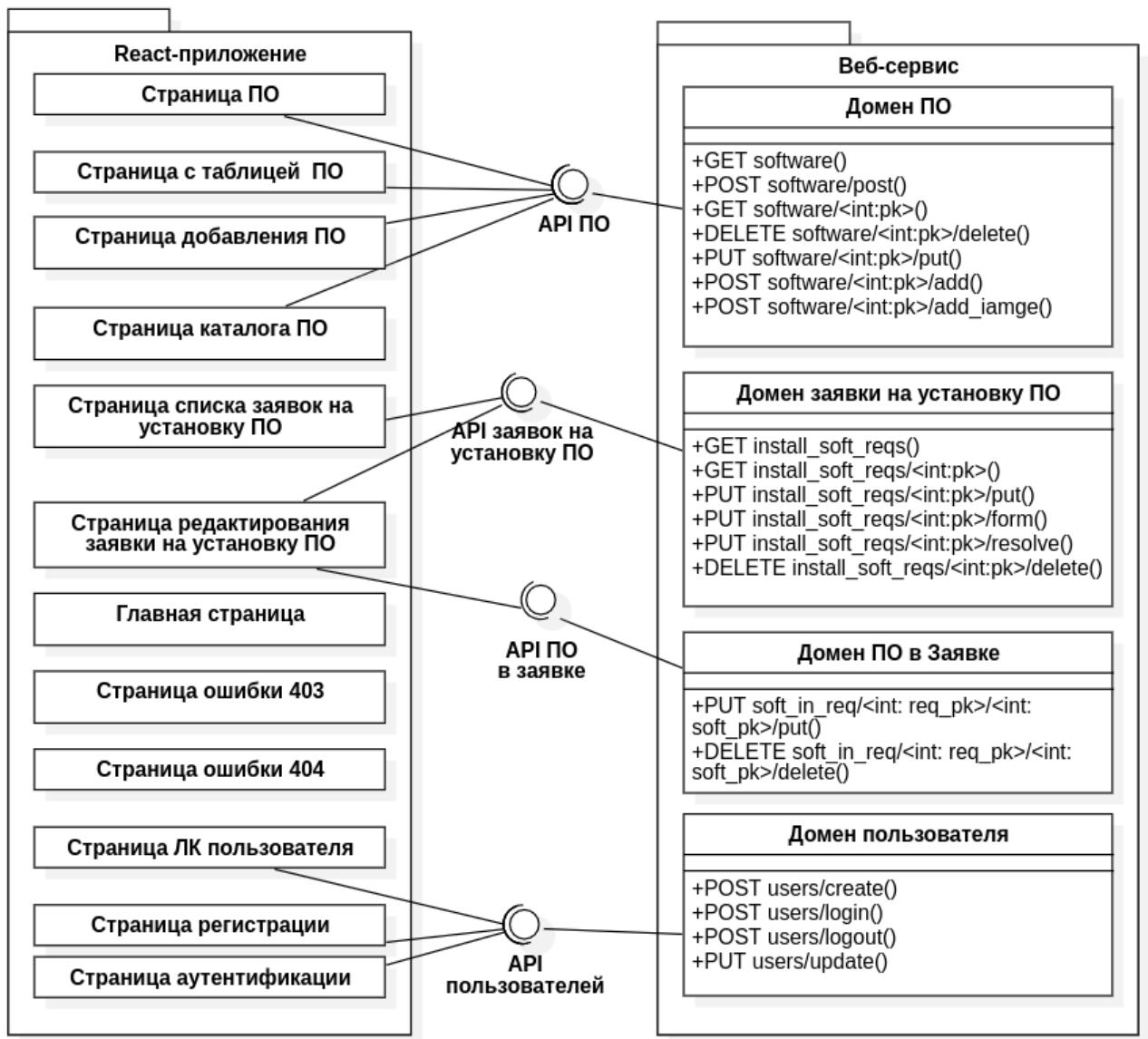


Рисунок 7 – Диаграмма классов фронтенда

### 3 АЛГОРИТМЫ

Алгоритм работы разработанной системы отображен на диаграмме последовательности (рисунок 8). В основе системы лежит веб-сервис, реализующий внутри себя всю бизнес-логику. Он предоставляет доступ к методам из следующих доменов: ПО, заявки на установку ПО, ПО в заявке, пользователи. Методы следуют правилам REST API.

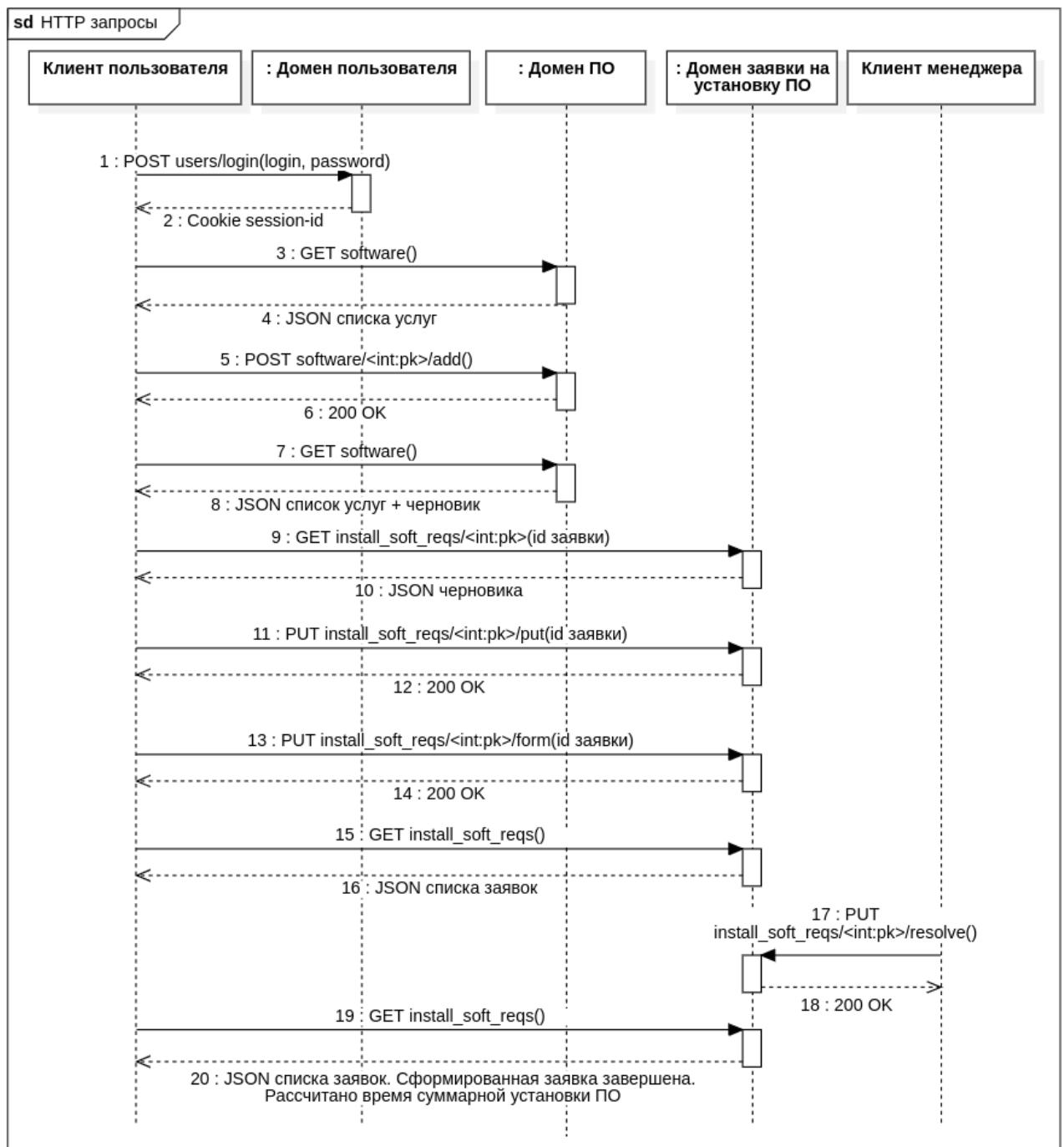


Рисунок 8 – Диаграмма последовательности

В начале бизнес-процесса происходит аутентификация пользователя. При помощи графического интерфейса пользователь отправляет запрос, передавая в нем логин и пароль. Если аккаунт с указанными данными существует в базе, на клиент возвращается информация о пользователе и устанавливаются куки с полем session\_id (идентификатор текущей сессии) в ответном запросе. Если же аккаунта с введённым логином не существует, или пароль введен неверно, на клиент возвращается ошибка. В таком случае пользователю надо пройти регистрацию, либо ввести верный пароль. На этом же этапе происходит проверка: является пользователь клиентом или менеджером. Графический интерфейс пользователя запрашивает у веб-сервиса список услуг (программ), которые возвращаются в JSON формате. Клиент выбирает программу, которую хочет добавить в заявку на установку, и, нажимая на кнопку «добавить» в графическом интерфейсе, отправляет запрос на добавление программы в свою черновую заявку. Этот процесс может продолжаться несколько раз.

Когда пользователь определится со списком необходимого для установки ПО, а также указал версии ПО и адрес хоста для установки, он нажимает кнопку «сформировать» в графическом интерфейсе. После чего приложение отправляет на веб-сервис запрос на формирование заявки. Клиент может отслеживать статус сформированных заявок на соответствующей странице в графическом интерфейсе.

Процесс рассмотрения заявок менеджерам так же происходит при помощи графического интерфейса. Менеджеры могут просматривать список всех созданных пользователями заявок; согласовывать или отклонять их при помощи соответствующих кнопок. Доступна фильтрация по логинам пользователей, а также статусам и дате формирования заявок. Менеджеры имеют возможность редактировать список ПО, доступного для установки, а также изменять данные о каждой программе. Для каждой из этих возможностей существует соответствующий метод, отправляемый на веб-сервис.

## 4 ОПИСАНИЕ ИНТЕРФЕЙСА

Главное меню приложения включает пункты, которые доступны в зависимости от роли пользователя (рисунки 9, 10, 11).



Рисунок 9 – Меню приложения (для гостя)



Рисунок 10 – Меню приложения (для клиента)



Рисунок 11 – Меню приложения (для менеджера)

На странице с формой регистрации (рисунок 12) отображается форма, при помощи которой гость может создать новый аккаунт. После успешной регистрации открывается форма аутентификации.

Регистрация

Логин

Введите логин

Пароль

Введите пароль

Зарегистрироваться

Рисунок 12 – Страница регистрации

На странице с формой входа (рисунок 13) отображается форма, через которую пользователь может войти в свой аккаунт. При успешном вводе данных аккаунта на клиент приходят куки с идентификатором текущей сессии.

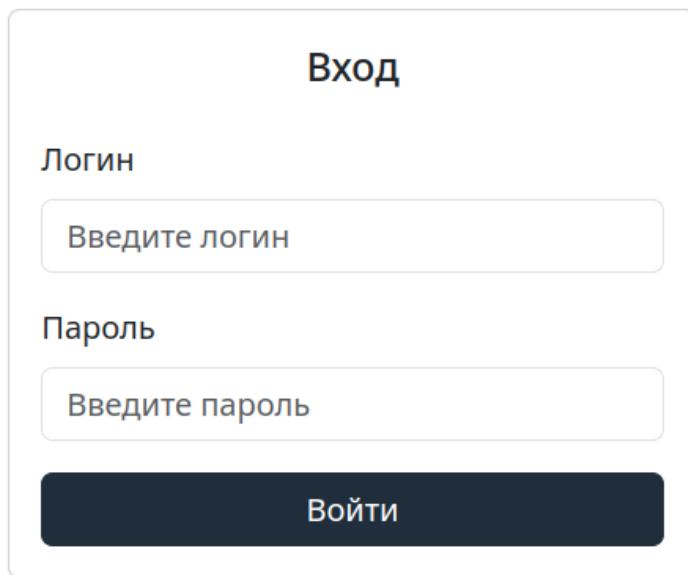


Рисунок 13 – Страница входа

После входа в аккаунт пользователь попадает на главную страницу (рисунок 14). На этой странице располагается описание разработанной системы.



### Установка ПО на ваши серверы



Заказали виртуальный сервер, но не можете поставить необходимое ПО? Этот сервис поможет вам! Оформите заявку и специалисты настроят всё в кратчайшие сроки.

Рисунок 14 – Главная страница

Страница каталога ПО (рисунок 15) содержит список программ в виде карточек. При нажатии на кнопку «Подробнее» на любой карточке, открывается страница с подробным описанием выбранного ПО. Также с помощью нажатия на кнопку «Добавить» клиент может добавлять программы в

черновик заявки. Вверху страницы находится строка для поиска ПО по названию.

The screenshot shows a software catalog page with a dark header bar containing 'Установка ПО' and 'Каталог' on the left, and 'Регистрация' and 'Вход' on the right. Below the header is a navigation bar with 'Главная / Каталог'. A search bar with 'Поиск' and a shopping cart icon with '0' are also present. The main content area displays four software items in a grid:

- NodeJS**: Programmatic platform based on V8 engine, transforming JavaScript into a general-purpose language. Price: 205 rubles. Button: Узнать подробнее.
- Git**: Distributed system for managing version control. Price: 150 rubles. Button: Узнать подробнее.
- JS**: Programming language primarily used in web development. Price: 400 rubles. Button: Узнать подробнее.
- Docker**: Software for automating application deployment. Price: 100 rubles. Button: Узнать подробнее.

Below the grid are three smaller placeholder boxes for other software items.

Рисунок 15 – Страница каталога ПО

На странице с подробным описанием программы выводится основная информация о ПО (рисунок 16).

The screenshot shows a software details page for Docker. The top navigation bar includes 'Установка ПО' and 'Каталог' on the left, and 'Регистрация' and 'Вход' on the right. The breadcrumb navigation shows 'Главная / Каталог / Docker'. The main content area features the Docker logo (a blue whale carrying shipping containers) and the word 'docker' below it. The product information is as follows:

- Docker**
- Цена установки:** 100 руб.
- Время установки:** 10 мин.
- Размер:** 1000 байт

**Описание**

Docker — программное обеспечение для автоматизации развертывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений. Позволяет «упаковать» приложение со всем своим окружением[англ.] и зависимостями в контейнер, который может быть развернут на любой Linux-системе с поддержкой контрольных групп в ядре, а также предоставляет набор команд для управления этими контейнерами. Изначально использовал возможности LXC, с 2015 года начал использовать собственную библиотеку, абстрагирующую виртуализационные возможности ядра Linux — libcontainer. С появлением Open Container Initiative начался переход от монолитной к модульной архитектуре.

Рисунок 16 – Страница с описанием ПО

На странице одной заявки (рисунок 17) отображается текущая заявка пользователя в виде таблицы. У пользователя есть возможность удалить

программу из заявки, отредактировать версию ПО, а также отправить заявку на проверку или удалить ее. Редактирование оформленных заявок недоступно

The screenshot shows a user interface for ordering software installations. At the top, there are navigation links: Установка ПО, Каталог, Заявки, and a sign-in status (1 Выход). Below the header, the URL is shown as Главная / Каталог / Заказ на установку ПО № 78. The main content area displays a list of items:

| Сервер пользователя | Приложение | Версия | Цена     | Удалить |
|---------------------|------------|--------|----------|---------|
|                     | NodeJS     |        | 205 руб. | X       |
|                     | Git        |        | 150 руб. | X       |
| Итого:              |            |        | 355 руб. |         |

At the bottom right are two buttons: 'Очистить' (Clear) and 'Оформить' (Place Order).

Рисунок 17 – Страница заявки на установку ПО

На странице списка заявок (рисунок 18) клиенты могут просматривать созданные ими заявки в виде таблицы. На этой странице можно посмотреть подробную информацию о заявке, нажав на её номер.

The screenshot shows a list of submitted software installation requests. At the top, there are navigation links: Установка ПО, Каталог, Заявки, and a sign-in status (1 Выход). The title of the page is 'Заявки'. Below the title, there are filters for 'Статус' (Any status), 'Дата оформления с' (From date), and 'Дата оформления по' (To date), each with a date input field and a clear button. A 'Показать' (Show) button is located to the right of the filters.

| №  | Статус    | Дата создания | Дата оформления | Дата завершения |
|----|-----------|---------------|-----------------|-----------------|
| 78 | В работе  | 19.11.2024    | 25.11.2024      |                 |
| 77 | Завершена | 17.11.2024    | 19.11.2024      | 19.11.2024      |
| 76 | Завершена | 17.11.2024    | 17.11.2024      | 18.11.2024      |
| 75 | Завершена | 17.11.2024    | 17.11.2024      | 18.11.2024      |
| 73 | Отклонена | 13.11.2024    | 13.11.2024      | 18.11.2024      |
| 72 | Завершена | 09.11.2024    | 09.11.2024      | 18.11.2024      |
| 70 | Отклонена | 09.11.2024    | 09.11.2024      | 18.11.2024      |
| 68 | Завершена | 09.11.2024    | 09.11.2024      | 18.11.2024      |
| 67 | Завершена | 09.11.2024    | 09.11.2024      | 18.11.2024      |
| 66 | Завершена | 09.11.2024    | 09.11.2024      | 18.11.2024      |

Рисунок 18 – Страница списка заявок на установку ПО

Пользователь может изменить свои данные (логин и пароль) на странице редактирования данных пользователя (рисунок 19). Она открывается при нажатии на логин пользователя в меню.

Изменить данные аккаунта

E-mail

Введите новый e-mail

Пароль

Введите новый пароль

Изменить

Рисунок 19 – Страница редактирования информации о пользователе  
Менеджер может просматривать информацию о всём ПО, занесённом в систему на странице списка ПО (рисунок 20). Менеджер имеет возможность добавить новое ПО, отредактировать или удалить существующее.

| NodeJS  | 205 руб. | Редактировать | Удалить |
|---------|----------|---------------|---------|
| Git     | 150 руб. | Редактировать | Удалить |
| JS      | 400 руб. | Редактировать | Удалить |
| Docker  | 100 руб. | Редактировать | Удалить |
| SuperJS | 0 руб.   | Редактировать | Удалить |

Рисунок 20 – Страница списка ПО

На странице списка всех заявок пользователей (рисунок 21), менеджер может согласовывать или отклонять заявки пользователей путём нажатия соответствующих кнопок в интерфейсе. Менеджер может просматривать подробную информацию о заявке путём нажатия на её номер. Доступны поля фильтрации по логину пользователя, статусу и датам формирования заявки.

| №  | Автор | Статус    | Дата создания | Дата оформления | Дата завершения | Завершить                  | Отклонить                  |
|----|-------|-----------|---------------|-----------------|-----------------|----------------------------|----------------------------|
| 78 | 1     | В работе  | 19.11.2024    | 25.11.2024      |                 | <button>Завершить</button> | <button>Отклонить</button> |
| 77 | 1     | Завершена | 17.11.2024    | 19.11.2024      | 19.11.2024      | <button>Завершить</button> | <button>Отклонить</button> |
| 76 | 1     | Завершена | 17.11.2024    | 17.11.2024      | 18.11.2024      | <button>Завершить</button> | <button>Отклонить</button> |
| 75 | 1     | Завершена | 17.11.2024    | 17.11.2024      | 18.11.2024      | <button>Завершить</button> | <button>Отклонить</button> |
| 73 | 1     | Отклонена | 13.11.2024    | 13.11.2024      | 18.11.2024      | <button>Завершить</button> | <button>Отклонить</button> |
| 72 | 1     | Завершена | 09.11.2024    | 09.11.2024      | 18.11.2024      | <button>Завершить</button> | <button>Отклонить</button> |
| 70 | 1     | Отклонена | 09.11.2024    | 09.11.2024      | 18.11.2024      | <button>Завершить</button> | <button>Отклонить</button> |

Рисунок 21 – Страница списка заявок на установку ПО

С помощью формы редактирования и создания ПО (рисунок 22) сотрудник может добавлять новые программы или изменять существующие.

**Редактирование ПО**

Название \*

Краткое описание \*

Программная платформа, основанная на движке V8, которая превращает JavaScript из узкоспециализированного языка в язык общего назначения.

Цена установки \*

Время установки (в мин.) \*

Размер (в байтах) \*

Логотип

Выберите файл | Файл не выбран

Полное описание \*

Node.js применяется преимущественно на сервере, выполняя роль веб-сервера. Однако есть возможность разрабатывать на Node.js и десктопные оконные приложения, а также программировать микроконтроллеры.

\* — обязательные поля

## Рисунок 22 – Страница редактирования и создания ПО

В случае, если пользователь запрашивает несуществующую страницу, он перенаправляется на страницу ошибки 404 (рисунок 23).

# 404

Страница не найдена

[На главную](#)

## Рисунок 23 – Страница 404

В случае, если пользователь пытается получить доступ к страницам, для просмотра которых ему не хватает прав, он перенаправляется на страницу ошибки 403 (рисунок 24).

# 403

Доступ запрещён

[На главную](#)

## Рисунок 23 – Страница 403

## **ЗАКЛЮЧЕНИЕ**

В ходе работы были достигнуты следующие результаты:

1. Создан MVP и базовый дизайн приложения на основе [firstbyte.ru](http://firstbyte.ru);
2. Создана базу данных для хранения информации о ПО и заявках на установку;
3. Создан веб-сервис на Django (DRF);
4. Реализована авторизация и хранение сессий в Redis;
5. Разработан базовый SPA на React для гостя;
6. Внедрена адаптивность, менеджер состояний Redux Toolkit, PWA, разработано Tauri приложение;
7. Завершена разработка интерфейса пользователя в React;
8. Реализован интерфейс менеджера;
9. Разработано десктопное приложение Tauri
10. Приложение развернуто при помощи сервиса GitHub Pages и доступно по ссылке: <https://yu-leo.github.io/install-server-software-frontend/>
11. Подготовлен набор документации, включающий РПЗ, ТЗ и набор диаграмм.
12. Оформлен git-репозиторий на сервисе GitHub, содержащий исходный код проекта: <https://github.com/Yu-Leo/install-server-software-backend>

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Серверы (мировой рынок) [Электронный ресурс] // Tadviser. URL: [https://www.tadviser.ru/index.php/Статья:Серверы\\_\(мировой\\_рынок\)](https://www.tadviser.ru/index.php/Статья:Серверы_(мировой_рынок)) (дата обращения: 10.11.2024);
2. Robotic It Service [Электронный ресурс] // Ritsgroup. URL: <https://moscow.ritsgroup.ru/> (дата обращения: 10.11.2024);
3. Distributed Tracing for Cascading Changes of Objects in the Kubernetes Control Plane [Электронный ресурс] // Arxiv. URL: <https://arxiv.org/abs/2411.01336> (дата обращения: 20.11.2024);
4. Документация по Django [Электронный ресурс] // djangoproject. URL: <https://www.djangoproject.com/> (дата обращения: 20.10.2024);
5. Документация по DRF [Электронный ресурс] // Django-rest-framework. URL: <https://www.django-rest-framework.org/topics/documenting-your-api/> (дата обращения: 20.10.2024);
6. Документация по Redis [Электронный ресурс] // Netlify. URL: <https://master--redis-doc.netlify.app/docs/> (дата обращения: 10.10.2024);
7. Документация Minio [Электронный ресурс] // Min. URL: <https://min.io/docs/minio/kubernetes/upstream/index.html> (дата обращения: 09.09.2024);
8. Документация PostgreSQL [Электронный ресурс] // Postgresql. URL: <https://www.postgresql.org/docs/> (дата обращения: 15.09.2024);
9. Документация по Python [Электронный ресурс] // Python. URL: [https://docs.python.org/3/index.html/](https://docs.python.org/3/index.html) (дата обращения: 20.10.2024);
10. Документация по Tauri [Электронный ресурс] // Tauri. URL: <https://v2.tauri.app/develop/> (дата обращения: 20.11.2024);
11. Документация по React [Электронный ресурс] // React. URL: <https://react.dev/learn> (дата обращения: 01.11.2024);

**ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ**  
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

---

Кафедра «Системы обработки информации и управления»

Утверждаю  
Заведующий кафедрой ИУ-5  
В.И.Терехов  
"\_\_\_" 2024 г.

Согласовано  
Научный руководитель  
И.И. Иванов  
"\_\_\_" 2024 г.

**Заявки на установку серверного ПО**

Техническое задание  
(вид документа)

писчая бумага  
(вид носителя)

17  
(количество листов)

**ИСПОЛНИТЕЛЬ:**

Ювенский Лев  
Александрович  
"\_\_\_" 2024 г.

Москва - 2024

## **1. Введение**

Цель системы, включающей в себя веб-сервис, веб-приложение и десктопное приложение Tauri, – предоставить интерфейс для учета заявок на установку серверного ПО.

## **2. Назначение разработки**

Заявочная система предназначена для обеспечения удобного и эффективного процесса создания и управления заявками на установку ПО на клиентские сервисы. Система ориентирована на использование провайдерами, предоставляющими виртуальные серверы, желающими расширить диапазон предоставляемых услуг задачами на установку ПО на серверах.

Пользователи имеют возможность создавать заявки на установку списка ПО и отслеживать статусы своих заявок.

Менеджеры имеют возможность редактировать список доступного для установки ПО, просматривать активные заявки от пользователей, отклонять или успешно завершать их.

## **3. Стадии и этапы разработки**

- 3.1. Создание MVP и базового дизайна на основе firstbyte.ru;
- 3.2. Создать базу данных для хранения информации о ПО и заявках на установку;
- 3.3. Создать веб-сервис в бэкенде на Django (DRF);
- 3.4. Реализовать авторизацию и хранение сессий в Redis;
- 3.5. Разработать базовый SPA на React для гостя;
- 3.6. Внедрение адаптивности, разработка Tauri приложения;
- 3.7. Интерфейс пользователя клиента в React с менеджером состояний Redux Toolkit;
- 3.8. Реализовать интерфейс сотрудника поддержки в React;
- 3.9. Разработать десктопное приложение Tauri
- 3.10. Развернуть приложение при помощи GitHub Pages

3.11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.

3.12. Оформить git-репозиторий на сервисе GitHub, содержащий исходный код проекта

#### **4. Требования к функциональным характеристикам**

##### 4.1. Методы HTTP

4.1.1. POST Регистрация

4.1.2. POST Вход в аккаунт

4.1.3. POST Выход из аккаунта

4.1.4. PUT Изменение данных пользователя

4.1.5. GET Получение списка заявок на установку ПО

4.1.6. GET Получение одной заявки на установку ПО

4.1.7. PUT Изменение данных о заявке на установку ПО

4.1.8. PUT Формирование заявки на установку ПО

4.1.9. PUT Разрешение заявки на установку ПО (отклонение или завершение)

4.1.10. DELETE Удаление заявки на установку ПО

4.1.11. GET Получение списка доступного ПО

4.1.12. GET Получение информации о ПО

4.1.13. POST Добавление информации о ПО

4.1.14. PUT Изменение информации о ПО

4.1.15. PUT Добавление ПО в заявку

4.1.16. POST Добавление логотипа ПО

4.1.17. DELETE Удаление ПО

4.1.18. PUT Изменение информации о ПО в заявке

4.1.19. DELETE Удаление ПО из заявки

##### 4.2. Меню

4.2.1. Главная. Ссылка на страницу 4.3. Доступно всем пользователям

4.2.2. Каталог ПО для установки. Ссылка на страницу 4.6. Доступно всем пользователям

4.2.3. Заявки. Ссылка страницу 4.9. Доступно только авторизованному пользователю

4.2.4. Таблица ПО. Ссылка на страницу 4.14. Доступно только менеджерам

4.2.5. Войти. Ссылка на страницу 4.4. Доступно только неавторизованному пользователю

4.2.6. Зарегистрироваться. Ссылка на страницу 4.5. Доступно только неавторизованному пользователю

4.2.7. Личный кабинет. Ссылка на страницу 4.11. Доступно только авторизованному пользователю

#### 4.3. Главная страница.

4.3.1. Доступна всем пользователям.

4.3.2. На этой странице находится текст с приветствием.

##### 4.3.3. Действия

4.3.3.1. Войти – переход на страницу 4.4. Только для неавторизованных пользователей

4.3.3.2. Регистрация - переход на страницу 4.5. Только для неавторизованных пользователей

4.3.3.3. Полный список ПО – переход на страницу 4.6. Доступно всем пользователям

4.3.3.4. Список ПО в текущей заявке – переход на страницу 4.12. Только для авторизованных пользователей

4.3.3.5. Список заявок – переход на страницу 4.9. Только для авторизованных пользователей.

4.3.3.6. Добавить ПО – переход на страницу 4.8. Только для менеджеров

4.3.3.7. Таблица заявок на установку ПО – переход на страницу 4.14. Только для менеджеров

4.3.3.8. Выйти – переход на страницу 4.3, выход из аккаунта (метод 4.1.3)

#### **4.4. Страница аутентификации.**

4.4.1. Доступна неавторизованным пользователям

4.4.2. Действия

4.4.2.1. Войти (метод 4.1.2)

4.4.2.2. Зарегистрироваться – переход на страницу 4.5

#### **4.5. Страница регистрации**

4.5.1. Доступна неавторизованным пользователям

4.5.1.1. Действия

4.5.1.2. Зарегистрироваться (метод 4.1.1)

4.5.1.3. Войти – переход на страницу 4.4

#### **4.6. Страница с каталогом доступного для установки ПО.**

4.6.1. Доступна всем пользователям

4.6.2. Выводится информация о ПО в виде карточек (метод 4.1.11) и  
форма для поиска ПО по названию.

4.6.3.

4.6.3.1. Логотип ПО

4.6.3.2. Название

4.6.3.3. Стоимость установки

4.6.3.4. Краткое описание

4.6.4. Действия

4.6.4.1. Отфильтровать ПО по названию (метод 4.1.11)

4.6.4.2. Добавить ПО в заявку на установку ПО (метод 4.1.16). Только  
для авторизованных пользователей

4.6.4.3. Подробнее – перенаправляет на страницу 4.7

#### **4.7. Страница с подробным описанием ПО**

4.7.1. Доступна всем пользователям

4.7.2. Выводится подробная информация о конкретном ПО (метод 4.1.12)

4.7.2.1. Логотип ПО

4.7.2.2. Название

4.7.2.3. Размер

4.7.2.4. Стоимость установки

4.7.2.5. Время установки

4.7.2.6. Описание

#### 4.7.3. Действия

4.7.3.1. Редактировать – переход на страницу 4.8. Доступно только менеджерам.

4.7.3.2. Добавить в «корзину» - заявку на установку ПО. Доступно только авторизованным пользователям

### 4.8. Страница для добавления/редактирования ПО

4.8.1. Доступна только менеджерам

4.8.2. Выводится информация о ПО в построчном виде (метод 4.1.11)

4.8.3. Действия

4.8.3.1. Сохранить – сохранение изменений в существующем ПО (методы 4.1.15 и 4.1.17)

4.8.3.2. Добавить – добавление нового ПО в каталог (методы 4.1.13 и 4.1.17)

### 4.9. Страница со списком заявок.

4.9.1. Доступна только авторизованным пользователям. Пользователям доступны только их заявки. Менеджерам доступны заявки пользователей. Присутствует фильтрация по статусам заявок и дамам оформления.

4.9.2. Выводится информация о заявках в табличном виде (метод 4.1.5)

4.9.2.1. Id заявки

4.9.2.2. Статус заявки

4.9.2.3. Дата создания

4.9.2.4. Дата формирования

4.9.2.5. Дата завершения

4.9.3. Действия

4.9.3.1. Открыть заявку – переход на страницу 4.10.

4.9.3.2. Завершить заявку (метод 4.1.9). Доступно только менеджерам

4.9.3.3. Отклонить заявку (метод 4.1.9). Доступно только менеджерам

4.10. Страница с подробным описанием заявки на установку.

4.10.1. Доступна только авторизованным пользователям.

Пользователь может изменять данные заявки. Менеджер может просматривать заявки пользователей

4.10.2. Выводится информация о ПО в виде горизонтальных карточек

4.10.3. Действия

4.10.3.1. Удалить ПО из заявки. Доступно только создателю заявки (метод 4.1.19)

4.10.3.2. Изменить данные ПО в заявке. Доступно только создателю заявки (метод 4.1.18)

4.10.3.3. Изменить данные заявки. Доступно только создателю заявки (метод 4.1.7)

4.10.3.4. Сформировать заявку. Доступно только владельцу заявки, если заявка находится в статусе черновика (метод 4.1.8)

4.10.3.5. Удалить заявку. Доступно только владельцу заявки, если заявка находится в статусе черновика (метод 4.1.10)

4.10.3.6. Открыть ПО – переход на страницу 4.10

4.11. Страница личного кабинета пользователя

4.11.1. Доступна только авторизованным пользователям

4.11.2. Действия

4.11.2.1. Изменение информации о пользователе (метод 4.1.4)

4.11.2.2. Выйти – переход на страницу 4.3, выход из аккаунта (метод 4.1.3)

4.12. Страница статуса «403 Forbidden».

4.12.1. Отображается, если у пользователя нет доступа к запрашиваемой странице

4.13. Страница статуса «404 Not found».

4.13.1. Отображается, если запрашиваемая страница не найдена

4.14. Страница с таблицей ПО.

4.14.1. Доступна только менеджерам

4.14.2. Выводится информация о ПО в табличном виде (метод 4.1.11)

4.14.2.1. Название ПО

4.14.2.2. Краткое описание

4.14.2.3. Стоимость установки

4.14.2.4. Время установки

4.14.2.5. Размер

4.14.3. Действия

4.14.3.1. Открыть ПО – переход на страницу 4.7

4.14.3.2. Добавить ПО – переход на страницу 4.8

**5. Требования к составу и параметрам технических средств**

5.1. Серверная часть

5.1.1. Процессор минимум 2-ядерный с частотой от 2 ГГц.

5.1.2. Оперативная память от 4 Гб.

5.1.3. Место на жестком диске от 2 Гб.

5.2. Клиентская часть

5.2.1. Процессор с частотой от 1ГГц.

5.2.2. Оперативная память от 512 Мб.

**6. Требования к информационной и программной совместимости**

6.1. Сервер

6.1.1. ОС: Linux

6.1.2. Интерпретатор Python 3.10

6.1.3. Redis 7.2.3

6.1.4. Minio RELEASE.2024-09-16T04-13-10Z

6.1.5. Postgres 16.0

6.2. Клиент

6.2.1. Версия веб-браузера (по поддержке React-Bootstrap 4.5, Redux, Swagger):

6.2.1.1. Chrome >= 45

6.2.1.2. Firefox >= 38

6.2.1.3. Edge >= 12

6.2.1.4. Explorer >= 10

6.2.1.5. Safari >= 9

6.2.1.6. Opera >= 30

## ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ

Таблица 1 – HTTP методы разрабатываемого веб-сервиса

| №      | Метод | URL           | Описание   | Входные данные   | Выходные данные |
|--------|-------|---------------|--|--|-----------------|
| 4.1.1. | POST  | /users/create | Регистрация  | {<br>login: string,<br>password: string<br>}             |                 |
| 4.1.2. | POST  | /users/login  | Вход в аккаунт   | {<br>login: string<br>password: string<br>}              | auth cookies    |
| 4.1.3. | POST  | /users/logout | Выход из аккаунта.<br>Доступно только<br>авторизованным<br>пользователям | auth cookies   |                 |
| 4.1.4. | POST  | /users/update | Изменение данных<br>пользователя   | auth cookies<br>{<br>username: string,<br>email: string, | auth cookies    |

| <b>№</b> | <b>Метод</b> | <b>URL</b>                     | <b>Описание</b>  | <b>Входные данные</b>  | <b>Выходные данные</b>  |
|----------|--------------|--------------------------------|--|--|---|
|          |              |                                |  | password: string<br>}  | username: string,<br>}  |
| 4.1.5.   | GET          | /install_software_requests     | Получение всех заявок на установку ПО с фильтрацией по статусу и дате создания.<br><br>Авторизованному пользователю доступны только его заявки.<br><br>Менеджеру доступны все заявки | status: string<br>start_datetime: string<br>end_datetime: string | [{<br><br>id: number<br><br>status: string<br><br>host: string<br><br>creation_datetime: string<br><br>formation_datetime: string<br><br>finish_datetime: string<br><br>user_id: number<br><br>user_login: string<br><br>manager_id: number<br><br>}] |
| 4.1.6.   | GET          | /install_software_requests /id | Получение заявки на установку ПО по id   | -  | {<br><br>id: number<br><br>status: string<br><br>host: string   |

| <b>№</b> | <b>Метод</b> | <b>URL</b> | <b>Описание</b> | <b>Входные данные</b> | <b>Выходные данные</b>  |
|----------|--------------|------------|-----------------|-----------------------|---|
|          |              |            |                 |                       | <pre> creation_datetime: string formation_datetime: string finish_datetime: string user_id: number user_login: string manager_id: number software: [     id: number     title: string     price: number     installing_time: number     size: number     img_url: string     summary: string ], items_in_cart: number, install_software_request_id: number </pre> |

| <b>№</b> | <b>Метод</b> | <b>URL</b>                             | <b>Описание</b>  | <b>Входные данные</b>    | <b>Выходные данные</b>   |
|----------|--------------|--|--|--------------------------|--|
|          |              |  |  |                          | version: string<br>]<br>}  |
| 4.1.7.   | PUT          | /install_software_requests /id/put     | Изменение данных заявки на установку ПО.<br>Доступно только авторизованному пользователю | {<br>host: string<br>}   | -  |
| 4.1.8.   | PUT          | /install_software_requests /id/form    | Формирование заявки на установку ПО. Доступно только авторизованному пользователю        | -                        | -  |
| 4.1.9.   | PUT          | /install_software_requests /id/resolve | Завершение заявки.<br>Доступно только менеджеру  | {<br>status: string<br>} | {<br>id: number<br>status: string<br>user_id: number<br>manager_id: number         } |

| <b>№</b> | <b>Метод</b> | <b>URL</b>                            | <b>Описание</b>  | <b>Входные данные</b>    | <b>Выходные данные</b>   |
|----------|--------------|---------------------------------------|--|--------------------------|--|
|          |              |                                       |  |                          | creation_datetime: string<br>formation_datetime: string<br>finish_datetime: string<br>}  |
| 4.1.10.  | DELETE       | /install_software_requests /id/delete | Удаление заявки на установку ПО.<br>Доступно только авторизованному пользователю | -                        | -  |
| 4.1.11.  | GET          | /software                             | Получение списка ПО и информации о текущей заявке-черновике.                     | title_startswith: string | {     software: [       {         id: number         title: string         subtitle: string         price: number         installing_time: number         size: number     }   ] } |

| <b>№</b> | <b>Метод</b> | <b>URL</b>   | <b>Описание</b>                 | <b>Входные данные</b> | <b>Выходные данные</b>   |
|----------|--------------|--------------|---------------------------------|-----------------------|--|
|          |              |              |                                 |                       | <pre>        img_url: string         summary: string       },       items_in_cart: number,       install_software_request_id: number     ]   }</pre> |
| 4.1.12.  | GET          | /software/id | Получение информации о ПО по id | -                     | <pre>{   id: number   title: string   subtitle: string   price: number   installing_time: number   size: number   summary: string }</pre>            |

| <b>№</b> | <b>Метод</b> | <b>URL</b>          | <b>Описание</b>  | <b>Входные данные</b>  | <b>Выходные данные</b>   |
|----------|--------------|---------------------|--|--|--|
| 4.1.13.  | POST         | /software/post      | Добавление нового ПО в каталог. Доступно только менеджеру. | {<br>id: number<br>title: string<br>subtitle: string<br>price: number<br>installing_time: number<br>size: number<br>summary: string<br>} | {<br>id: number<br>title: string<br>subtitle: string<br>price: number<br>installing_time: number<br>size: number<br>summary: string<br>} |
| 4.1.14.  | DELETE       | /software/id/delete | Удаление ПО из каталога. Доступно только менеджеру         | -  | -  |
| 4.1.15.  | PUT          | /software/id/put    | Изменение информации о ПО по id. Доступно только менеджеру | {<br>id: number<br>title: string<br>subtitle: string<br>price: number<br>}   | {<br>id: number<br>title: string<br>subtitle: string<br>price: number<br>}   |

| <b>№</b> | <b>Метод</b> | <b>URL</b>                                | <b>Описание</b>   | <b>Входные данные</b>   | <b>Выходные данные</b>  |
|----------|--------------|---|---|---|---|
|          |              |   |   | installing_time: number<br>size: number<br>summary: string<br>} | installing_time: number<br>size: number<br>summary: string<br>} |
| 4.1.16   | POST         | /software/id/add                          | Добавление ПО в заявку.<br>Доступно только авторизованному пользователю.    | -   | -   |
| 4.1.17   | POST         | /software/id/add_image                    | Добавление логотипа ПО в заявку. Доступно только менеджеру                  | file: multipart/formdata  | -   |
| 4.1.18.  | DELETE       | software_in_request/req_id/soft_id/delete | Удаление ПО из заявки.<br>Доступно только для авторизованных пользователей. | -   | -   |
| 4.1.19   | PUT          | software_in_request/req_id/soft_id/       | Указание версии ПО в заявке   | {<br>version: string  | -   |

| <b>№</b> | <b>Метод</b> | <b>URL</b> | <b>Описание</b> | <b>Входные данные</b> | <b>Выходные данные</b> |
|----------|--------------|------------|-----------------|-----------------------|------------------------|
|          |              | put        |                 | }                     |                        |