МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

«Storix» - веб приложение инвентаризации и управлением складом.

Курсовая работа

Направление: 09.03.04. Программная инженерия

Зав. Кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ д. ф.-м. н, доцент С.Д. Махортов

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ст. преподаватель В.С. Тарасов

Руководитель практики \_\_\_\_\_\_\_\_\_\_\_\_ Г.В. Прядченко

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Вернигоров, 3 курс, о/о

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.А. Неупокоев, 3 курс, о/о

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.М. Шестопалов, 3 курс, о/о

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.П. Михайлов, 3 курс, о/о

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.А. Толпыгин, 3 курс, о/о

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.О. Голубева, 3 курс, о/о

Воронеж 2024

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc199964143)

[1 Постановка задачи 7](#_Toc199964144)

[1.1 Цели создания сайта 7](#_Toc199964145)

[1.2 Задачи сайта 7](#_Toc199964146)

[1.3 Требования к сайту 7](#_Toc199964147)

[1.3.1 Требования к сайту в целом 7](#_Toc199964148)

[1.3.2 Функциональные требования для сайта 7](#_Toc199964149)

[1.3.3 Требования к оформлению и верстке страниц 8](#_Toc199964150)

[1.3.4 Требования к защите информации 8](#_Toc199964151)

[1.4 Задачи, решаемые в процессе разработки 9](#_Toc199964152)

[2 Анализ предметной области 10](#_Toc199964153)

[2.1 Обзор аналогов 10](#_Toc199964154)

[2.2 WMS24 10](#_Toc199964155)

[2.3 Первый бит 1С:WSM24 10](#_Toc199964156)

[2.4 Stels Company 11](#_Toc199964157)

[2.5 Storix 11](#_Toc199964158)

[2.6 Общее сравнение сервисов. 12](#_Toc199964159)

[3 Моделирование системы 13](#_Toc199964160)

[3.1 Диаграмма прецедентов (Use Case) 13](#_Toc199964161)

[3.2 ER-диаграмма 15](#_Toc199964162)

[3.3 Диаграмма развертывания (Deployment diagram) 15](#_Toc199964163)

[4 Реализация 17](#_Toc199964164)

[4.1 Средства реализации 17](#_Toc199964165)

[4.2 Архитектура клиентской части 17](#_Toc199964166)

[4.3 Разработка frontend (клиентской части веб-приложения) 17](#_Toc199964167)

[4.4 Архитектура серверной части 19](#_Toc199964168)

[4.5 Реализация интерфейса 19](#_Toc199964169)

[4.5.1 Страницы неавторизованного пользователя 22](#_Toc199964170)

[4.5.2 Страницы рабочего 24](#_Toc199964171)

[4.5.3 Страницы администратора (компании) 26](#_Toc199964172)

[4.5.4 Страницы системного администратора 29](#_Toc199964173)

[5 Тестирование 32](#_Toc199964174)

[ЗАКЛЮЧЕНИЕ 36](#_Toc199964175)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 37](#_Toc199964176)

Термины и определения

* **Бэкенд** - программно-аппаратная часть сервиса.
* **Django** - высокоуровневый Python фреймворк, который позволяет быстро создавать безопасные и поддерживаемые сайты.
* **Фронтенд** - сторона пользовательского интерфейса к программно-аппаратной части сервиса.
* **GitHub** - крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.
* **Python** - высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью.
* **Рабочий** - человек, имеющий доступ к базовому функционалу веб-сервиса в пределах компании.
* **Администратор** - человек, имеющий доступ к расширенному функционалу веб-сервиса в пределах компании.
* **Системный администратор** - человек, имеющий доступ к расширенному функционалу веб-сервиса за пределами компании.

ВВЕДЕНИЕ

В условиях современного бизнеса эффективное управление запасами и инвентаризацией склада становится ключевым фактором для достижения конкурентных преимуществ. Существующие методы учета и контроля запасов часто оказываются недостаточно эффективными, что приводит к ошибкам в учете, потерям и избыточным затратам. Проблема усугубляется увеличением объемов товарооборота и разнообразия ассортимента, что требует более гибких и надежных решений. Веб-приложения, способные автоматизировать процессы инвентаризации, представляют собой актуальный ответ на эти вызовы.

Целью данной курсовой работы является разработка веб-приложения, которое позволит автоматизировать инвентаризацию склада, улучшить точность учета и оптимизировать процессы управления запасами. В рамках работы будет проведен анализ существующих решений, выявлены их недостатки и определены функциональные требования к новому приложению. Это позволит создать систему, которая будет соответствовать современным требованиям бизнеса и обеспечит пользователям удобный и эффективный инструмент для управления запасами.

Ключевыми персонами, влияющими на тему автоматизации инвентаризации, являются разработчики программного обеспечения, специалисты по логистике и управления запасами, а также конечные пользователи, которые непосредственно взаимодействуют с системой. Их потребности и ожидания должны быть учтены при создании нового веб-приложения. Важно также отметить, что на рынке автоматизации наблюдаются тенденции к интеграции с ERP и CRM системами, что открывает новые возможности для повышения эффективности работы склада.

Актуальность темы автоматизации инвентаризации склада для России и мира обусловлена необходимостью повышения эффективности бизнес-процессов в условиях глобальной конкуренции. Внедрение современных технологий позволяет компаниям не только сократить затраты, но и улучшить качество обслуживания клиентов. В условиях экономической нестабильности и изменений в потребительских предпочтениях, автоматизация становится не просто желательной, а необходимой для выживания и развития бизнеса.

В рамках данной работы можно попытаться решить несколько задач: определить ключевые функциональные требования к веб-приложению, разработать архитектуру системы, выбрать технологии для разработки и протестировать прототип. Также важно рассмотреть, как внедрение такого приложения повлияет на общую эффективность работы склада и какие перспективы его дальнейшего развития.

Дополнительные вопросы, которые помогут раскрыть тему, могут включать: какие технологии являются наиболее подходящими для разработки веб-приложения? Каковы основные проблемы, с которыми сталкиваются компании при внедрении автоматизированных систем? Каковы лучшие практики интеграции с существующими ERP и CRM системами?

Таким образом, данная курсовая работа будет направлена на создание эффективного решения для автоматизации инвентаризации склада, что позволит не только улучшить процессы учета, но и повысить общую эффективность работы предприятия. В следующей части работы будет представлен более детальный анализ существующих решений и функциональных требований к новому веб-приложению.

1. Постановка задачи
   1. Цели создания сайта

Целями создания сайта являются:

* упрощение процесса проведения инвентаризаций по видео;
* минимизация вероятности ошибок при инвентаризации;
* получение прибыли с лицензии.
  1. Задачи сайта

Сайт позволяет решать следующие задачи:

* получать отчёт о проведении инвентаризации;
* просматривать историю инвентаризаций;
* создавать/удалять склад;
* создавать/удалять аккаунты рабочих;
* добавлять на склад нужного рабочего;
* осуществлять редактирование пароля аккаунта компании после авторизации.
  1. Требования к сайту
     1. Требования к сайту в целом

Данный сайт должно удовлетворять следующим основным требованиям:

* сайт должен быть доступен для использования на большинстве популярных браузеров (Яндекс, Google, Firefox);
* реализовывать все поставленные задачи.
  + 1. Функциональные требования для сайта

Разрабатываемый сайт должен соответствовать следующим функциональным требованиям:

Неавторизованный пользователь должен обладать возможностью:

* авторизоваться на сайте;
* просмотреть цену и описание сайта.

Рабочий) должен обладать возможностью:

* загружать видео и json для проведения инвентаризации;
* получать результат инвентаризации;
* просматривать историю инвентаризаций;
* выйти из профиля.

Администратора должен обладать возможностью:

* создавать/удалять склад;
* создавать/удалять аккаунт рабочего;
* просматривать все склады, привязанные к его аккаунту;
* просматривать все аккаунты рабочих, привязанные к его аккаунту;
* изменять пароль своего аккаунта;
* просматривать историю инвентаризаций;
* выйти из профиля;

Системный администратор должен обладать возможностью:

* создавать/удалять аккаунт компании;
* просматривать все аккаунты компаний;
* выйти из профиля;
  + 1. Требования к оформлению и верстке страниц

Оформление и верстка страниц должны удовлетворять следующим требованиям:

* сайт должно быть оформлено в едином стиле;
* должно быть название, присутствующее в оформление страниц;
* сайт должно быть разработано в одной цветовой палитре с использованием ограниченного набора шрифтов;
* цветовая палитра должна быть контрастной;
  + 1. Требования к защите информации

Для защиты информации должен использоваться токен. Он будет обновляться каждые пару минут, так что даже если его похитят, то у злоумышленника почти не будет времени для его использования.

* 1. Задачи, решаемые в процессе разработки

Были поставлены следующие задачи:

* анализ предметной области;
* обзор аналогов;
* постановка задачи;
* создание репозитория GitHub и таск-менеджера Jira;
* разработка требований к приложению;
* создание диаграмм: use case, ER, развертывания;
* разработка дизайна приложения;
* написание технического задания в соответствии с ГОСТ 34.602 – 2020;
* реализация интерфейса приложения;
* реализация серверной части приложения;
* развертывание приложения;
* написание курсовой работы.

1. Анализ предметной области
   1. Обзор аналогов

Был проведен анализ конкурентов в сфере инвентаризации склада, который позволяет выявить ключевые особенности сервисов. Ниже представлен обзор популярных в России сервисов инвентаризации склада, а также их преимущества и недостатки.

* 1. WMS24

Полноценная система для склада.

Плюсы:

* гибкие настройки управления складом, возможность интеграций;
* аналитика по сотрудникам;
* автоматическое размещение;
* блоки МХ;
* работа с Грузами, 3pl.

Минусы:

* высокая стоимость внедрения (от 444000₽);
* лимит по количеству товаров и строк в экспорте;
* малый функционал у базового тарифа с маленьким лимитом и отсутствием всех плюсов выше;
* для работы системы необходимо большое количество людей;
* отсутствие обработки через компьютерное зрение.

Вывод: WMS24 уступает конкурентам из-за большого количества ограничений в базовой версии и высокой цены.

* 1. Первый бит 1С:WSM24

Комплексное решение задач по автоматизации складской логистики

Плюсы:

* доработки функционала;
* хорошие аналитические инструменты для контроля работы;
* перенос убытков предыдущих периодов.

Минусы:

* высокая стоимость внедрения (320000);
* отсутствие обработки через компьютерное зрение;
* для работы системы необходимо большое количество людей.

Вывод: хотя ПервыйБит предлагает довольно хороший функционал для анализа склада, но для продукта слишком высокая цена.

* 1. Stels Company

Система управления складом с доступом к камерам.

Плюсы:

* не нужно заказывать WSM сканеры;
* контроль сотрудников по камерам;
* цена в 30000₽.

Минусы:

* тестовый период всего 3 дня, что не позволяет полноценно протестировать сервис;
* сложная структура инвентаризации;
* отсутствие обработки через компьютерное зрение.

Вывод: Stels Company хорошее приложение для управления складом, но отсутствие возможности инвентаризации делают его неудобным.

* 1. Storix

Платформа для помощи автоматизации инвентаризации склада

Плюсы:

* обработки через компьютерное зрение;
* минимальные сроки внедрения;
* минимальное время, затрачиваемое на инвентаризацию.

Минусы:

* отсутствие AI ответов.

Вывод: хотя в Storix нет возможности приёмки и продажи, но инвентаризация в данном приложении происходит быстрее, чем у конкурентов.

* 1. Общее сравнение сервисов.

В таблице 1 представлено сравнение аналогов по критериям.

1. Сравнение аналогов

|  |  |  |  |
| --- | --- | --- | --- |
| Компания | Необходимось сканера | Контроль геолокации товара | Компьютерное зрение |
| WMS24 | Да | Нет | Нет |
| Первый Бит | Да | Нет | Нет |
| Stels  company | Нет | Нет | Нет |
| Storix | Нет | Да | Да |

Выводы:

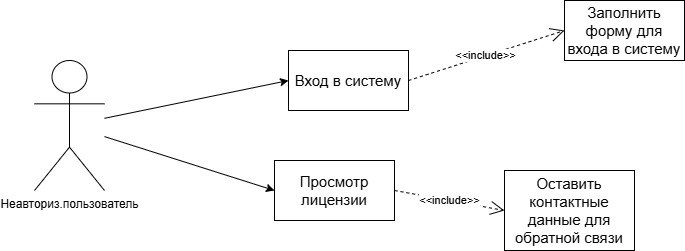
Stels Company— самый слабый сервис из-за ограниченных возможностях и отсутствия приёмки. ПервыйБит и WMS24 хорошие сервисы для работы склада, но отсутствие обработки через компьютерное зрение и контроля геолокации делают процесс инвентаризации затруднительным, а также необходимость покупки сканеров делают их менее удобными и затратными по времени и усилиям.

Storix — доступный сервис, но в нем отсутствует функционал для приёмки.

1. Моделирование системы
   1. Диаграмма прецедентов (Use Case)

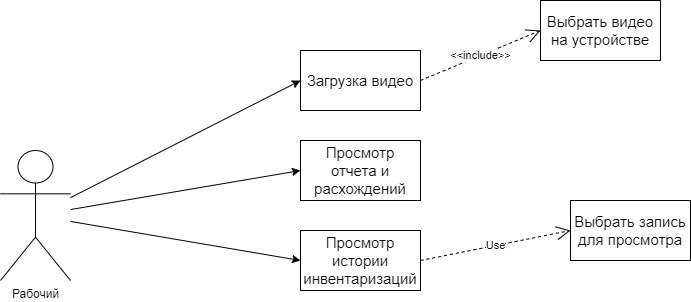
Диаграмма прецедентов – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

На рисунке 1 представлена диаграмма прецедентов для неавторизованного пользователя.



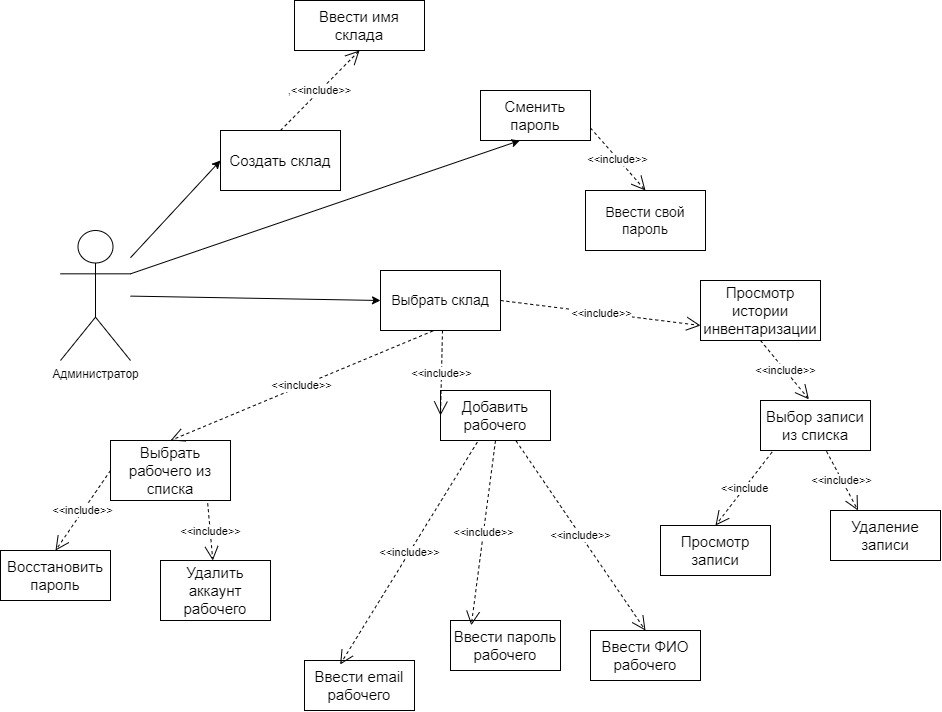
1. Use Case для неавторизованного пользователя

На рисунке 2 представлена диаграмма прецедентов для авторизованного пользователя (рабочего).



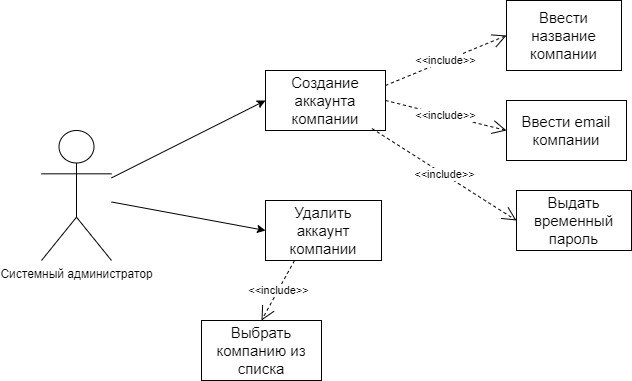
1. Use Case для авторизованного пользователя (рабочего)

На рисунке 3 представлена диаграмма прецедентов для администратора.



1. Use Case для администратора

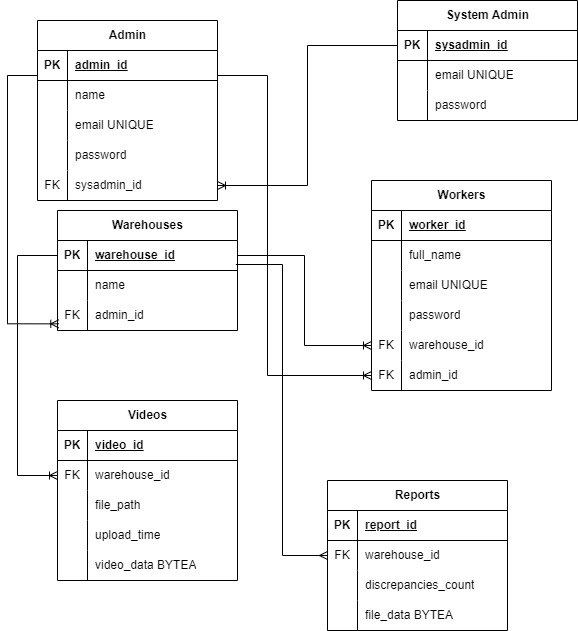
На рисунке 4 представлена диаграмма прецедентов для системного администратора.



1. Use Case для системного администратора
   1. ER-диаграмма

ER-диаграмма – это модель данных, визуализирующая связь между «сущностями» внутри системы.

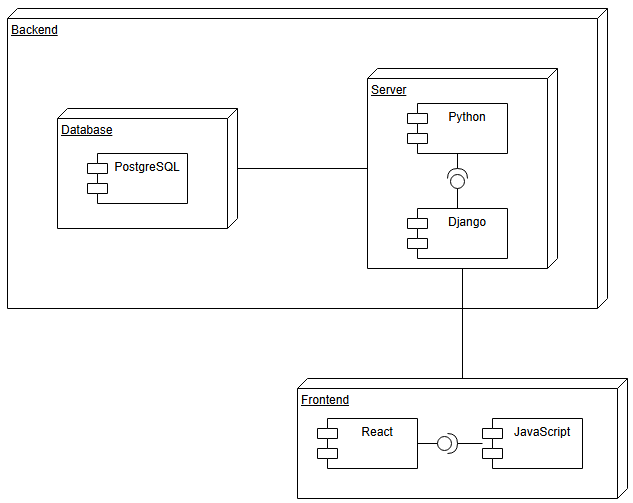
На рисунке 5 представлена ER-диаграмма для нашего сайта.



1. ER-диаграмма
   1. Диаграмма развертывания (Deployment diagram)

Диаграммы развертывания обычно используются для визуализации физического аппаратного и программного обеспечения системы. Она моделирует физическое развертывание артефактов на узлах.

На рисунке 6 представлена диаграмма развертывания для разрабатываемого приложения bookTalk.



1. Диаграмма развертывания
2. Реализация
   1. Средства реализации

Для реализации серверной части сайта используются следующие средства:

* язык программирования Python;
* фреймворк Django Framework;
* СУБД PostgreSQL;
* инструмент для создания документации API Swagger.

Для реализации клиентской части сайта используются следующие средства:

* стандартизированный язык разметки HTML;
* язык стилей CSS;
* язык программирования JavaScript;
* фреймворк React.

Для развертывания приложения будут использоваться следующие средства:

* Netlify – платформа для размещения фронтенд-части сайта;
* Docker – платформа для автоматизации развертывания бэкенд-сервиса;
* Nginx – прокси-сервер с поддержкой SSl.
  1. Архитектура клиентской части

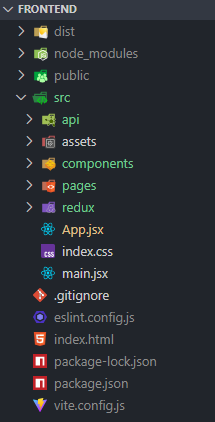
Приложение реализовано на трехуровневой архитектуре: клиент (веб-приложение) - сервер - база данных.

* 1. Разработка frontend (клиентской части веб-приложения)

Клиентская часть приложения была написана на языке JavaScript с использованием фреймворка React. Для отправки запросов с клиентской части приложения на серверную часть используется библиотека axios. Она предоставляет интерфейс для выполнения HTTP-запросов прямо из браузера.

Для создания модульной и повторно используемой архитектуры, приложение было разбито на компоненты. Каждый компонент отвечает за конкретную функциональность или отображение определенной части пользовательского интерфейса.

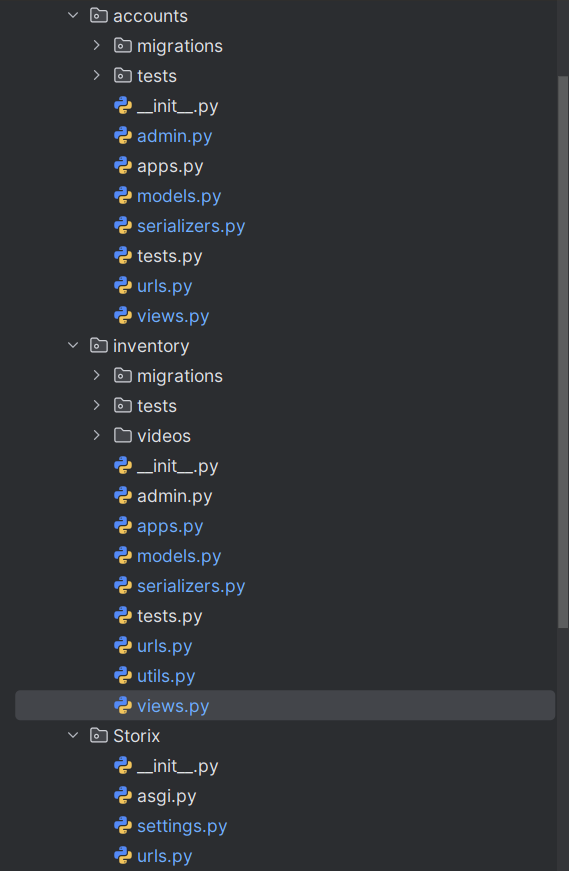
Структура клиентской части проекта показана на 7 рисунке.



1. Структура клиентской части приложения

* Модуль «api» содержит сервисы необходимые для связи с сервером, получением и отправкой данных;
* Модуль «assets» содержит шрифты и картинки, используемые на сайте.
* Модуль «components» содержит компоненты, которые используются в приложении;
* Модуль «pages» содержит компоненты, представляющие отдельные страницы приложения;
* Файл «main.jsx» является точкой входа в приложение;
* Файл «App.jsx» является основным компонентом React, в который встраиваются остальные компоненты.
  1. Архитектура серверной части

Серверная часть приложения реализована на языке Python с использованием фреймворка Django и Django REST Framework (DRF) для создания RESTful API. В качестве системы управления базами данных используется PostgreSQL. На рисунке 8 показа структура серверной части приложения.



1. Структура серверной части приложения
2. models.py — **модели базы данных.** Хранит описание всех сущностей системы:

* User: пользователь с ролями (sysadmin, admin, worker), связан с другими пользователями и складами.
* Warehouse: склад, может иметь нескольких админов.
* Video: видеофайл, загруженный рабочим, привязан к складу.
* Report: отчёт с данными о несоответствиях, привязан к складу и пользователю.

1. admin.py — **настройка админ-панели Django.** Позволяет удобно управлять объектами через интерфейс Django Admin:

* регистрирует модели User, Warehouse, Video, Report.
* добавляет дополнительные поля и фильтры для отображения в админке.
* контролирует доступ (например, кто может добавлять склады).

1. serializers.py — **сериализация данных (DRF).** Преобразует модели в json (и обратно):

* UserSerializer: сериализует пользователя, управляет паролем, ролями и привязкой к складам.
* WarehouseSerializer: сериализует склады и админов.
* VideoSerializer: сериализует видеофайлы (без video\_data).
* ReportSerializer: сериализует отчёты (без file\_data).

1. views.py — **контроллеры / ViewSet’ы. О**брабатывает запросы и содержит бизнес-логику:

* UserViewSet: создание пользователей с правами по ролям.
* WarehouseViewSet: CRUD для складов, с фильтрацией по ролям.
* VideoViewSet: загрузка и просмотр видео.
* ReportViewSet: работа с отчетами.
* sysadmin\_dashboard, admin\_dashboard, worker\_dashboard: заглушки или страницы дашбордов для разных ролей.

Также содержит классы прав доступа:

* IsSysAdmin, IsAdmin, IsWorker: используются для ограничения доступа к действиям в ViewSet’ах.

1. urls.py — **Маршруты API.** Определяет доступные адреса:

* users/, warehouses/, videos/, reports/ — маршруты для ViewSet’ов.
* dashboard/sysadmin/, admin/, worker/ — пути для дашбордов.

1. apps.py — конфигурация приложения Django. Определяет настройки приложения inventory:

* устанавливает имя приложения (inventory), под которым оно регистрируется в Django.
* указывает тип поля автоинкремента для моделей по умолчанию (BigAutoField).

1. models.py — модель отчёта об инвентаризации.   
   Содержит одну модель InventoryReport, которая описывает:

* рабочего, загрузившего видео;
* склад, к которому относится отчёт;
* видео (файл);
* конфигурацию анализа (список пар коробка-стеллаж в json);
* результат анализа (json);
* статус (pending или done);
* дату создания отчёта.

1. serializers.py — сериализация отчётов. Преобразует модель InventoryReport в json и обратно:

* читает видео, конфигурацию, результат, статус и дату;
* поля worker, status, result и created\_at — только для чтения;
* используется для передачи отчётов через API.

1. urls.py — маршруты для инвентаризации. Регистрирует маршруты API для отчётов:

* использует DRF DefaultRouter;
* доступен путь /reports/ с CRUD-операциями и дополнительным действием /reports/{id}/run/ для запуска анализа видео.

1. utils.py — видеоанализ штрихкодов. Функция analyze\_video\_fast():

* сканирует видеофайл с помощью OpenCV и Pyzbar;
* распознаёт QR-/штрихкоды коробок и стеллажей;
* сопоставляет найденные пары с ожидаемыми;
* возвращает результат анализа в виде json (все пары найдены или нет, где что найдено).

1. views.py — контроллер для отчётов. InventoryReportViewSet реализует:

* создание отчётов с привязкой к текущему пользователю и его складу;
* фильтрацию отчётов по ролям: рабочий видит свои, админ — по своим складам, сисадмин — все;
* метод run\_analysis, запускающий обработку видео, сохраняющий результат и меняющий статус отчёта на done.

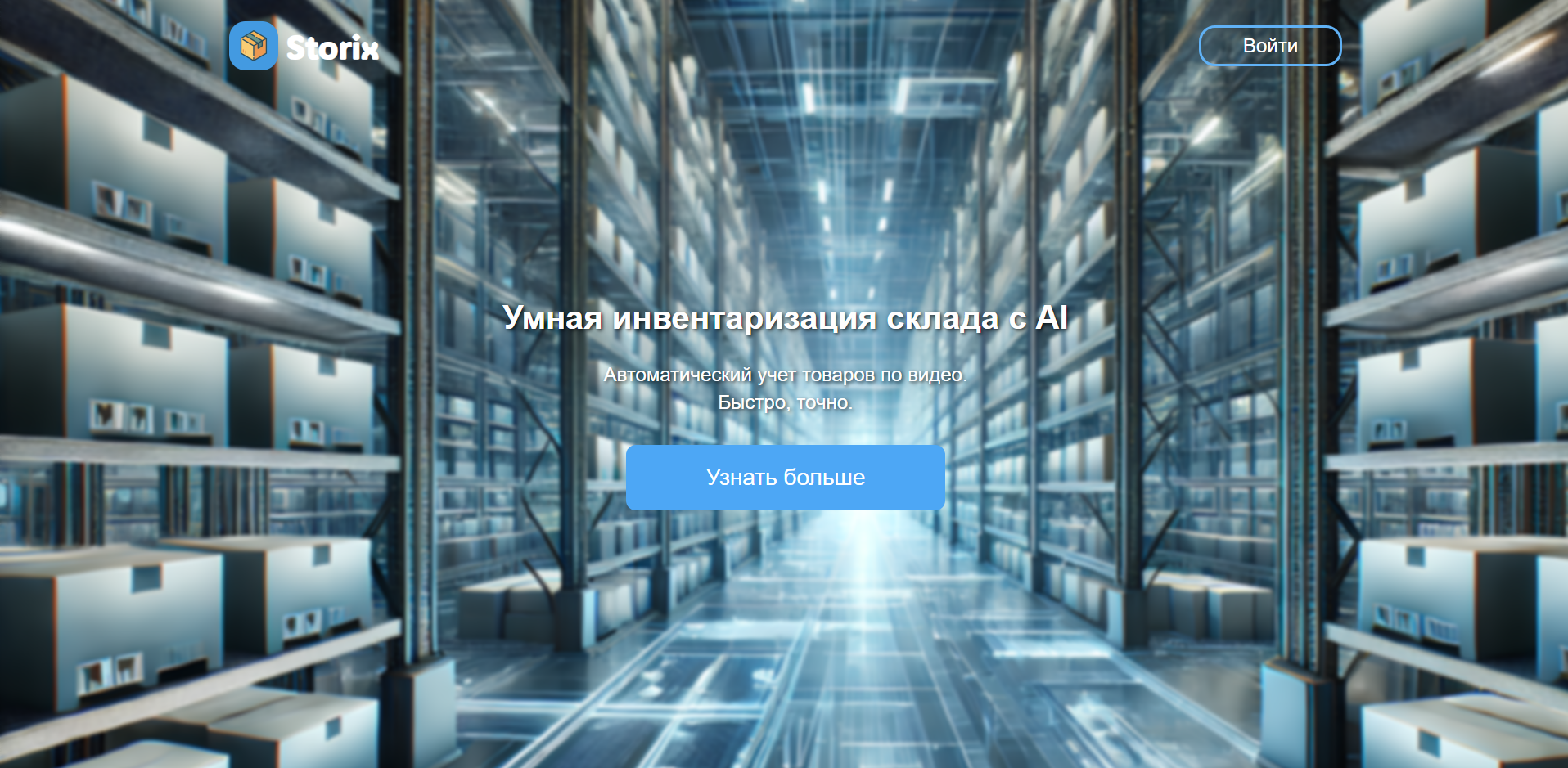
Вся документация API автоматически генерируется с помощью **Swagger (OpenAPI)** при использовании библиотеки **drf-spectacular**. Она доступна по следующим путям:

* /api/schema/ — спецификация OpenAPI в формате json;
* /api/docs/swagger/ — интерактивная Swagger-документация.

Документация охватывает все маршруты и модели, включая:

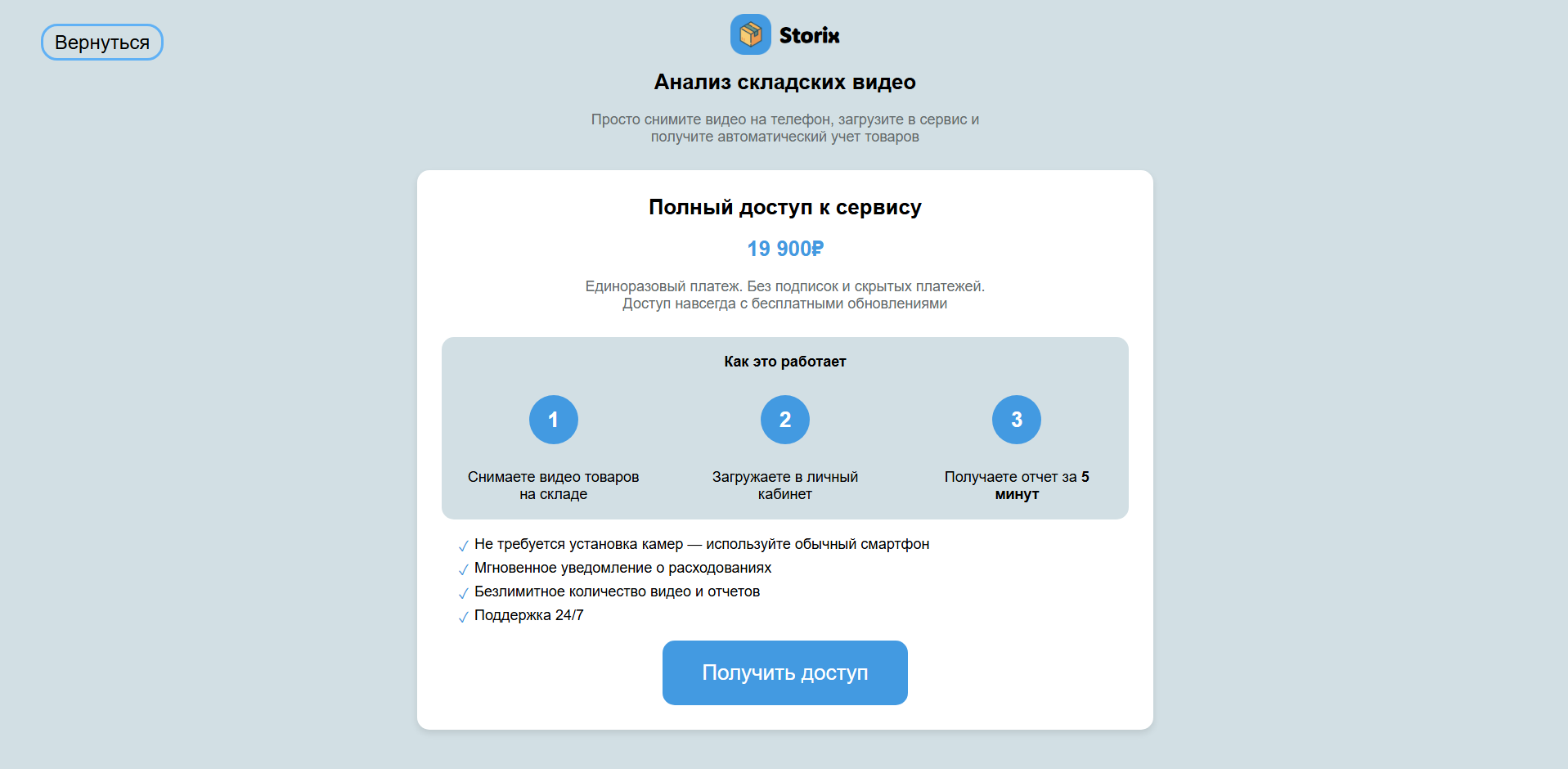
* описание входных и выходных данных;
* методы доступа (GET, POST, PUT, DELETE);
* примеры запросов и ответов;
* описание прав доступа для каждого ViewSet’а.
  1. Реализация интерфейса
     1. Страницы неавторизованного пользователя

При входе на сайт пользователю открывается главная страница. На рисунке 9 показана главная страница сайта Storix до авторизации в системе.

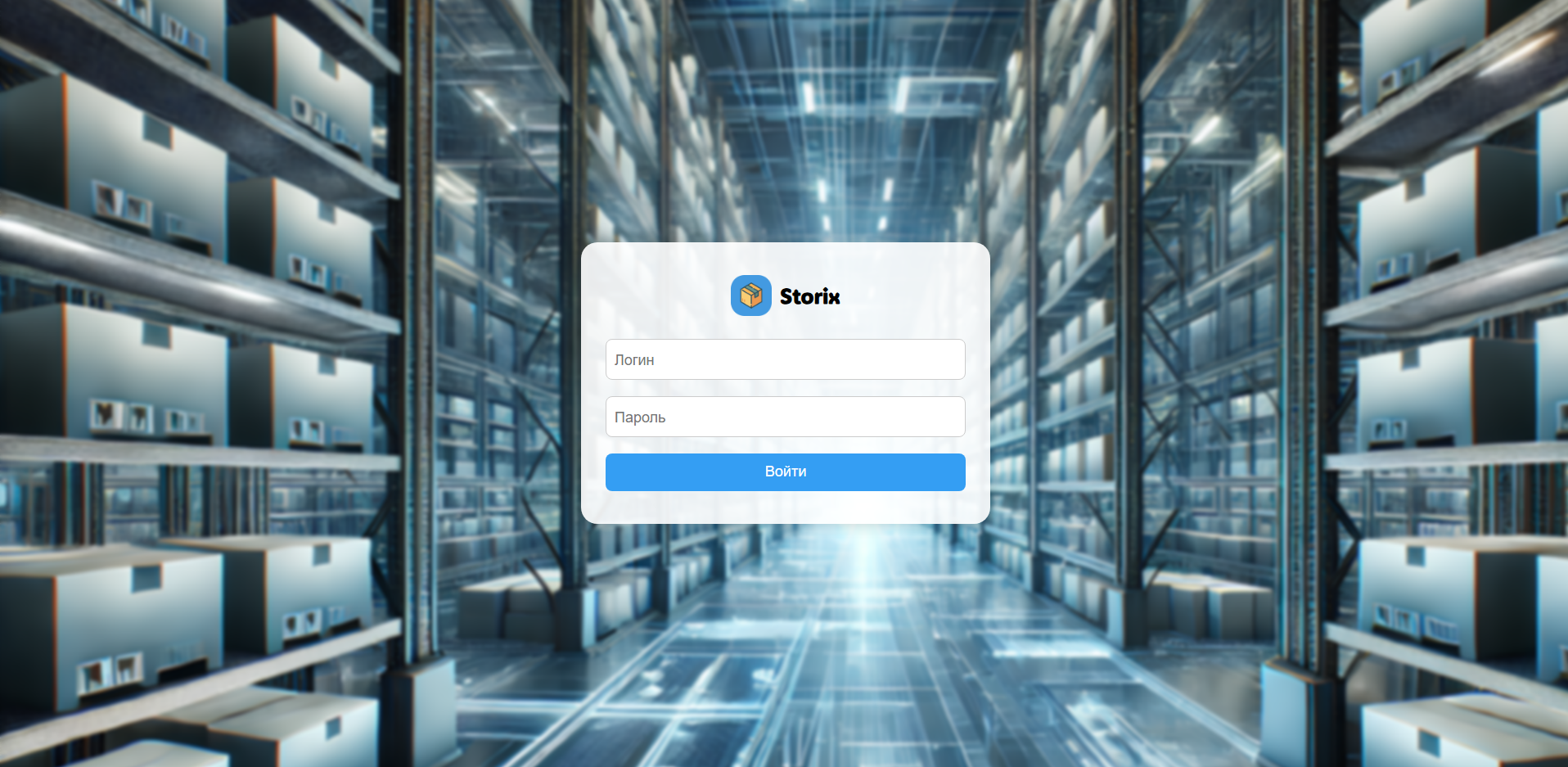


1. Главная страница

На этой странице пользователь по кнопке «Узнать больше» перейти страницу с подробной информацией о сайте, рисунок 10. Кроме этого, пользователь может авторизоваться, нажав на соответствующую кнопку, рисунок 11.

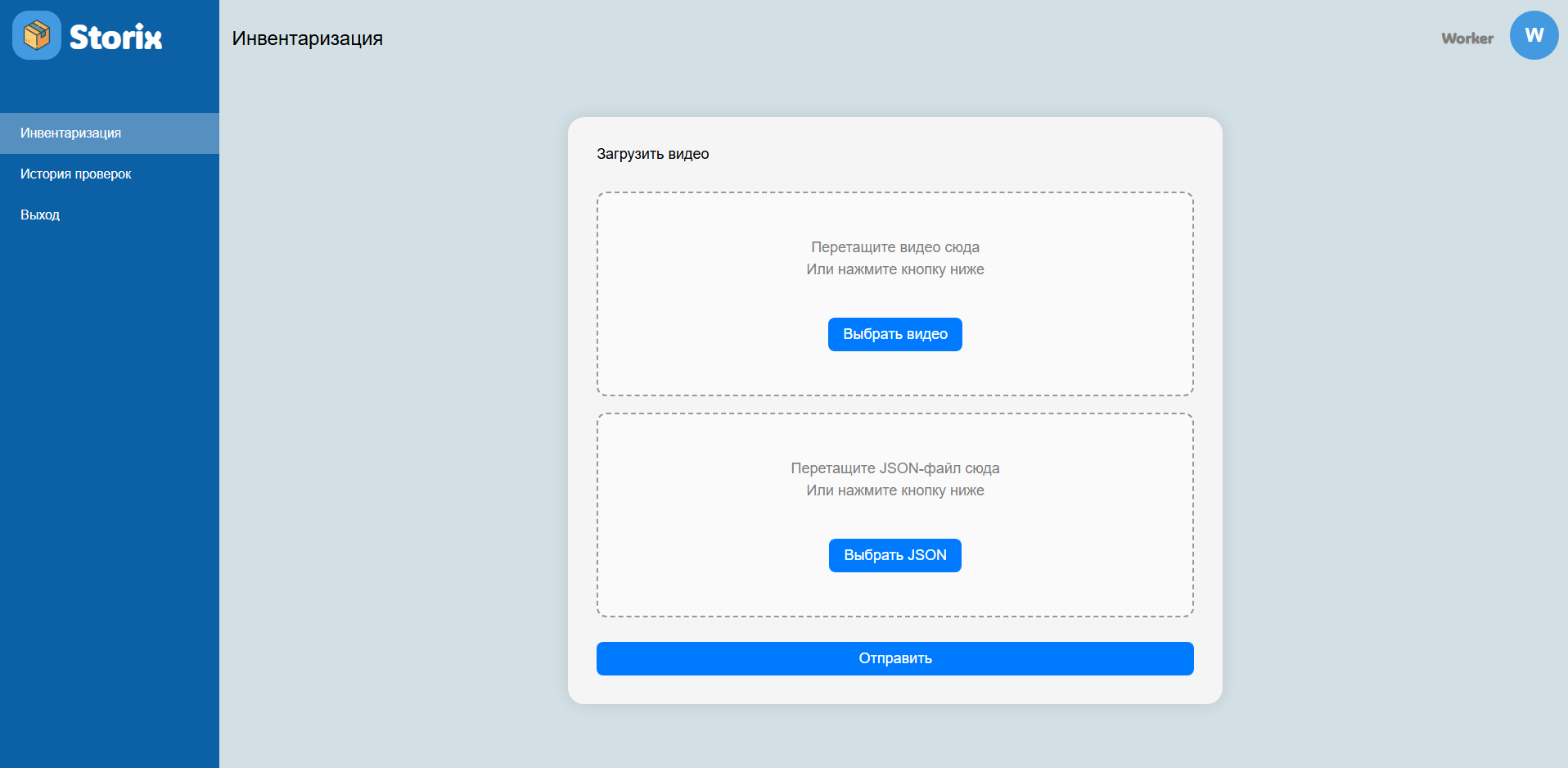


1. Страница с подробной информацией

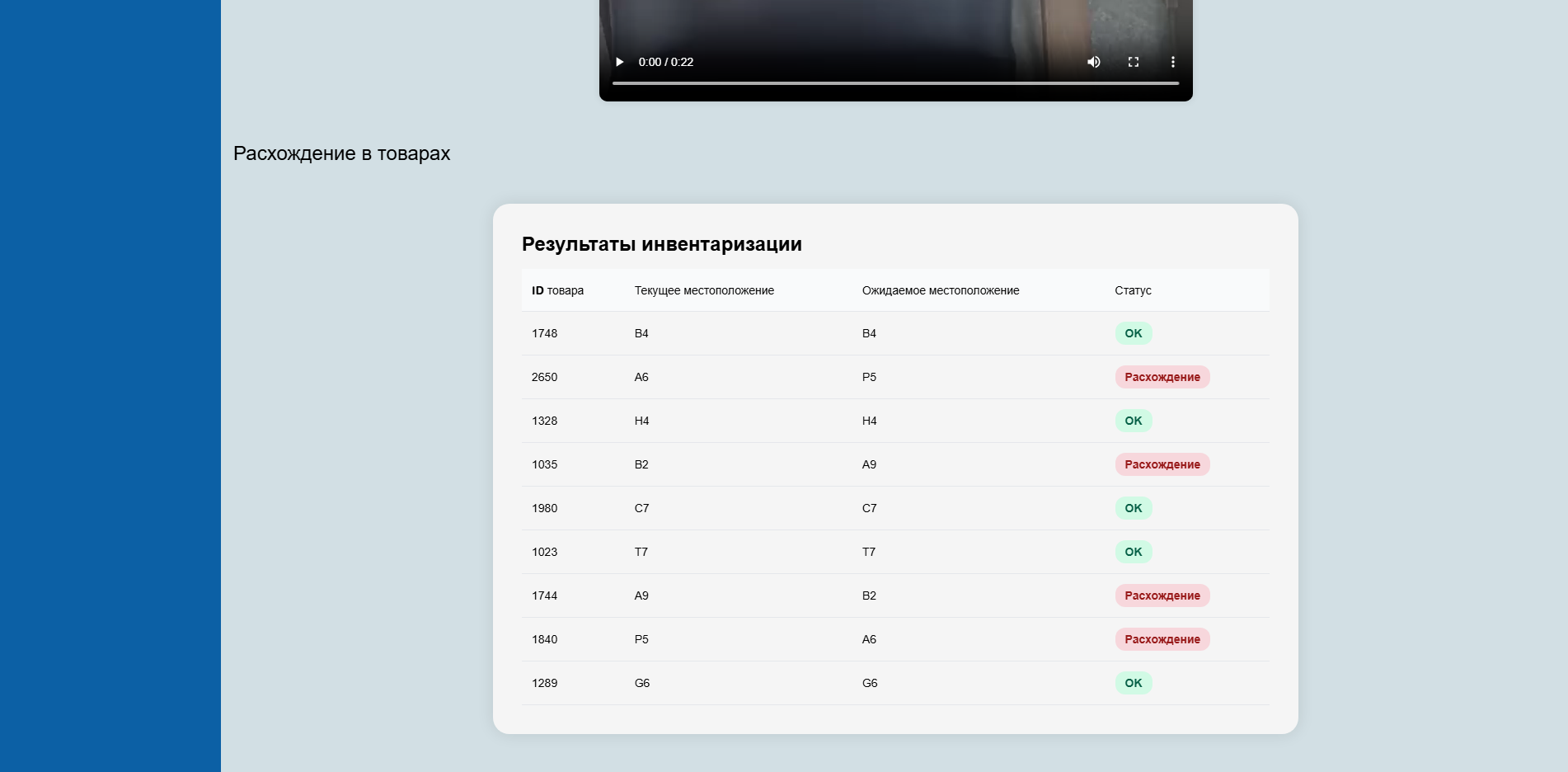


1. Страница авторизации
   * 1. Страницы рабочего

На странице с инвентаризацией рабочий может загрузить видео и json для проведения инвентаризации. Ниже рабочий может просмотреть видео, а также посмотреть результат проведения инвентаризации. На рисунках 12-13 показана страница с инвентаризацией рабочего.

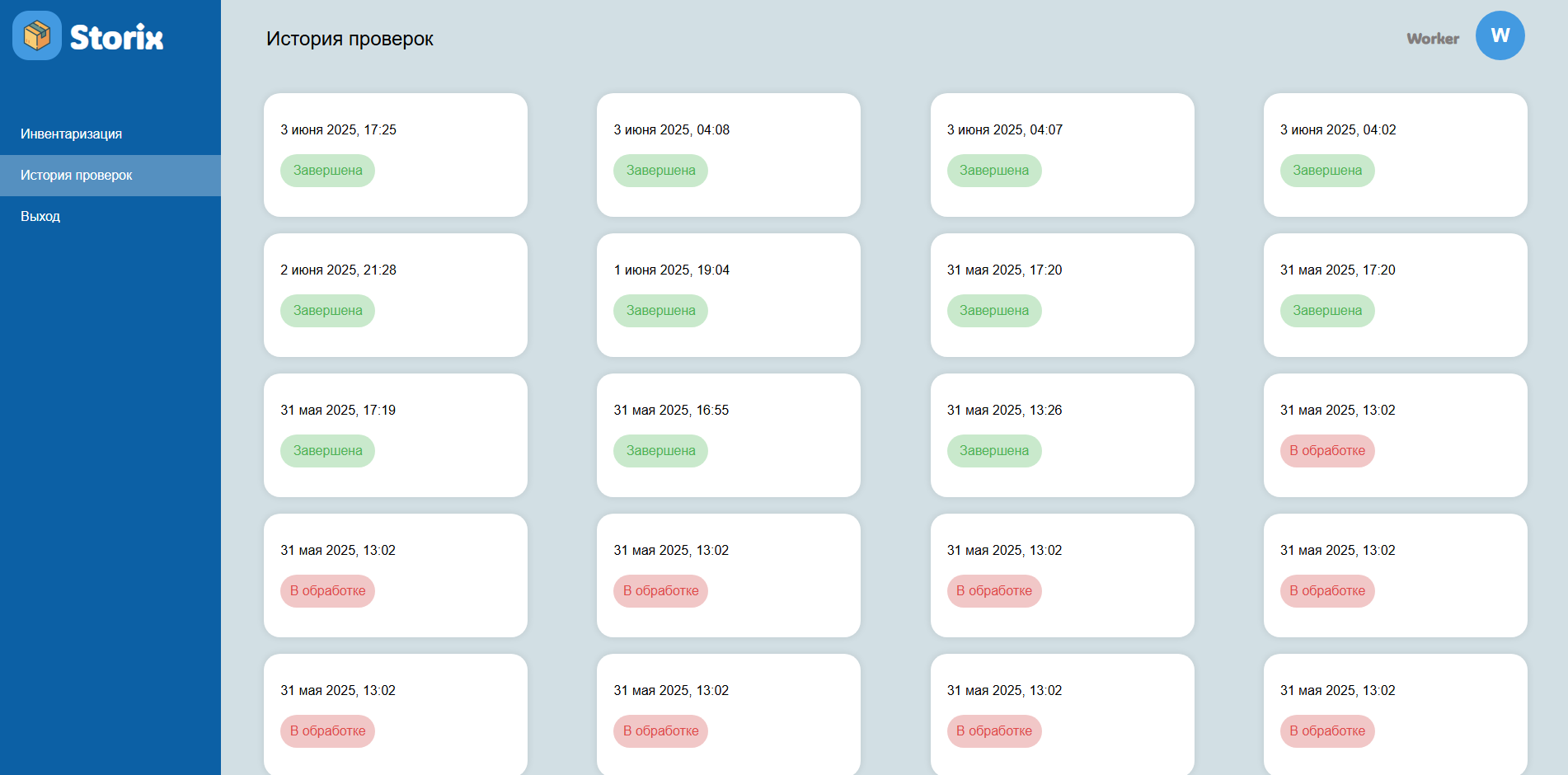


1. Страница с инвентаризацией, часть с загрузкой фидео и json

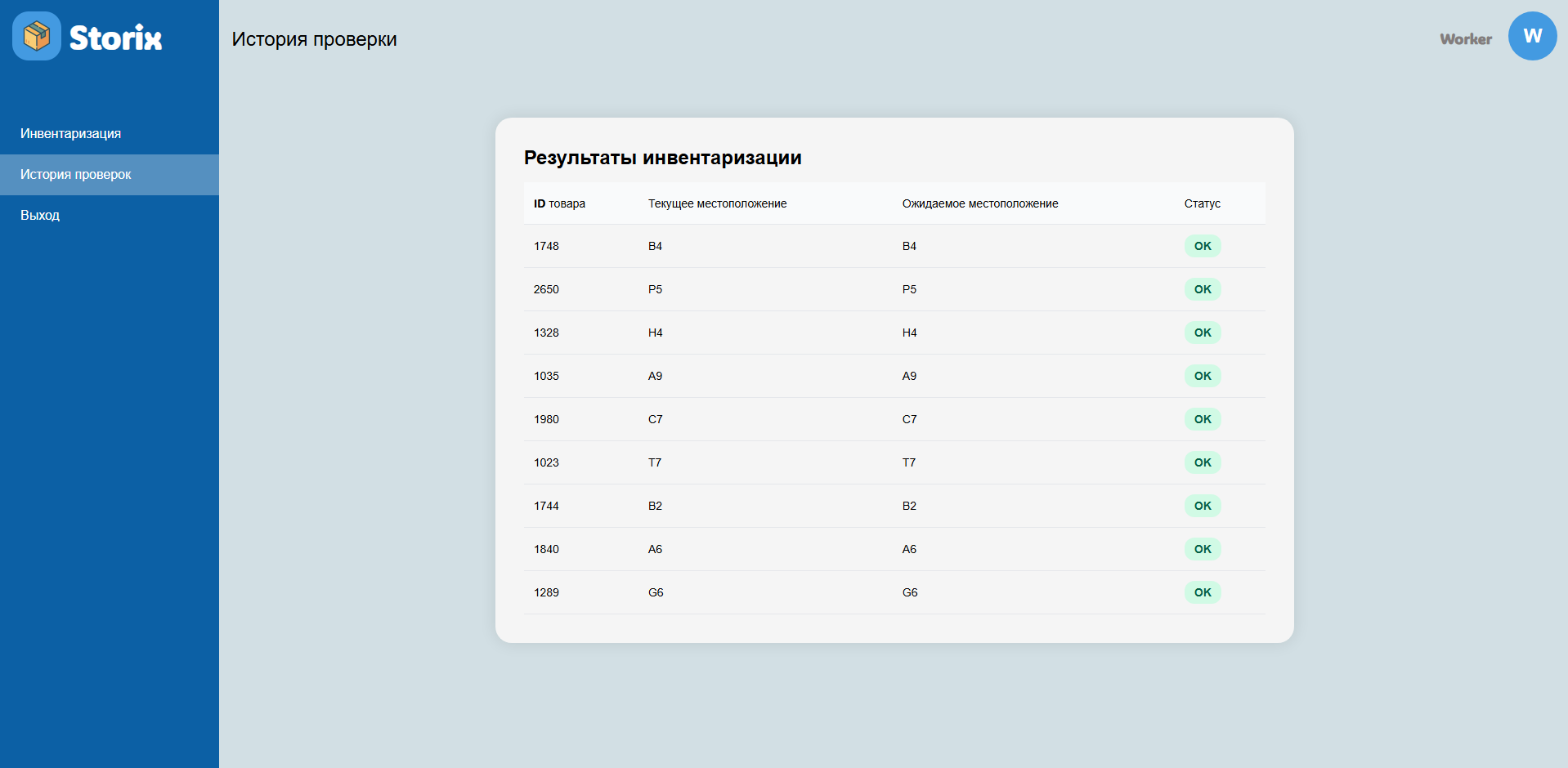


1. Страница с инвентаризацией, часть с результатом проверки

Рабочий может просмотреть историю проверок по соответствующей кнопке в боковом меню. При нажатии на любую из проверок, рабочий может посмотреть результат проверки. На рисунках 14-15 показана страница с историей инвентаризаций.

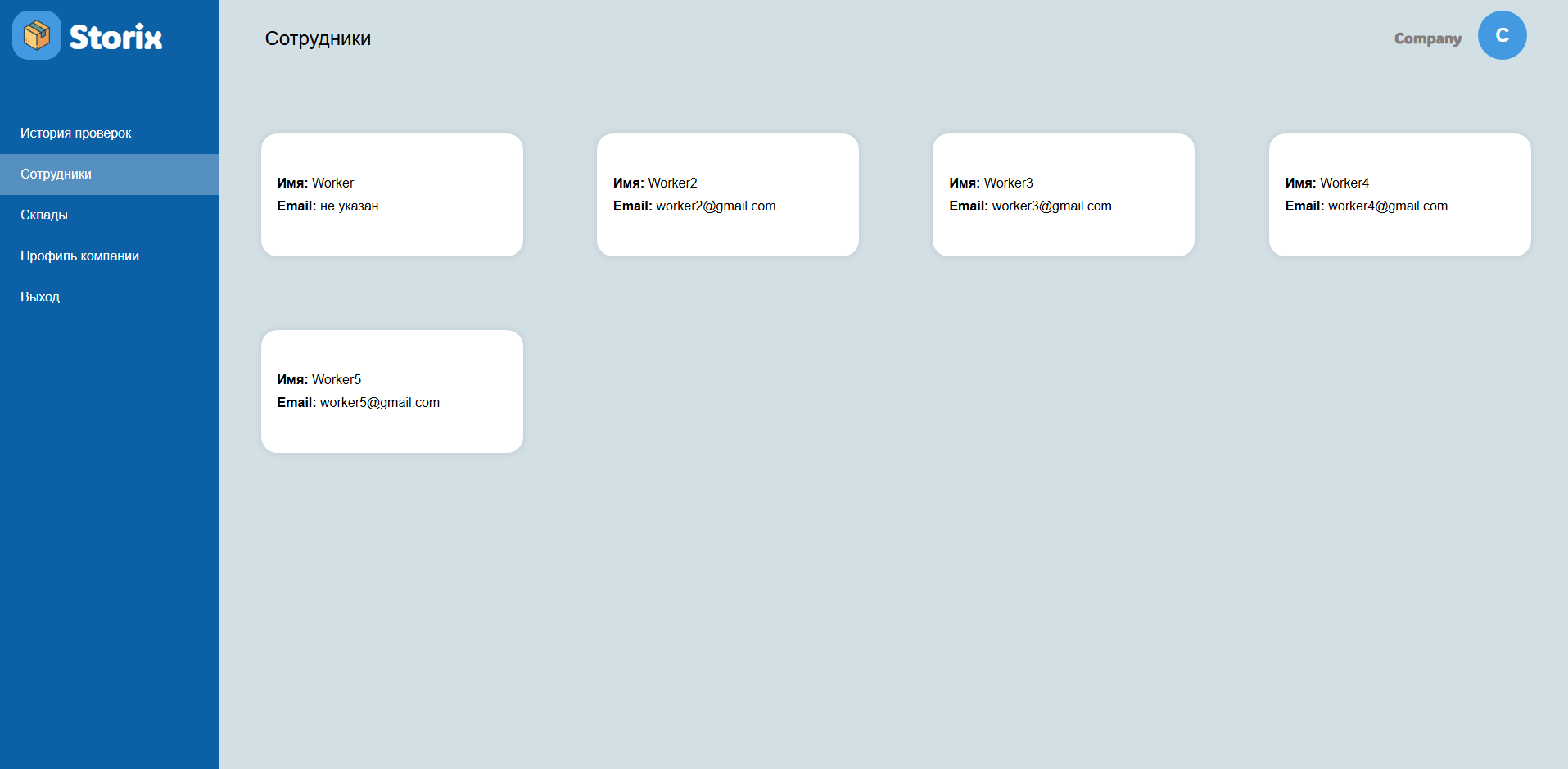


1. Страница с историей проверок



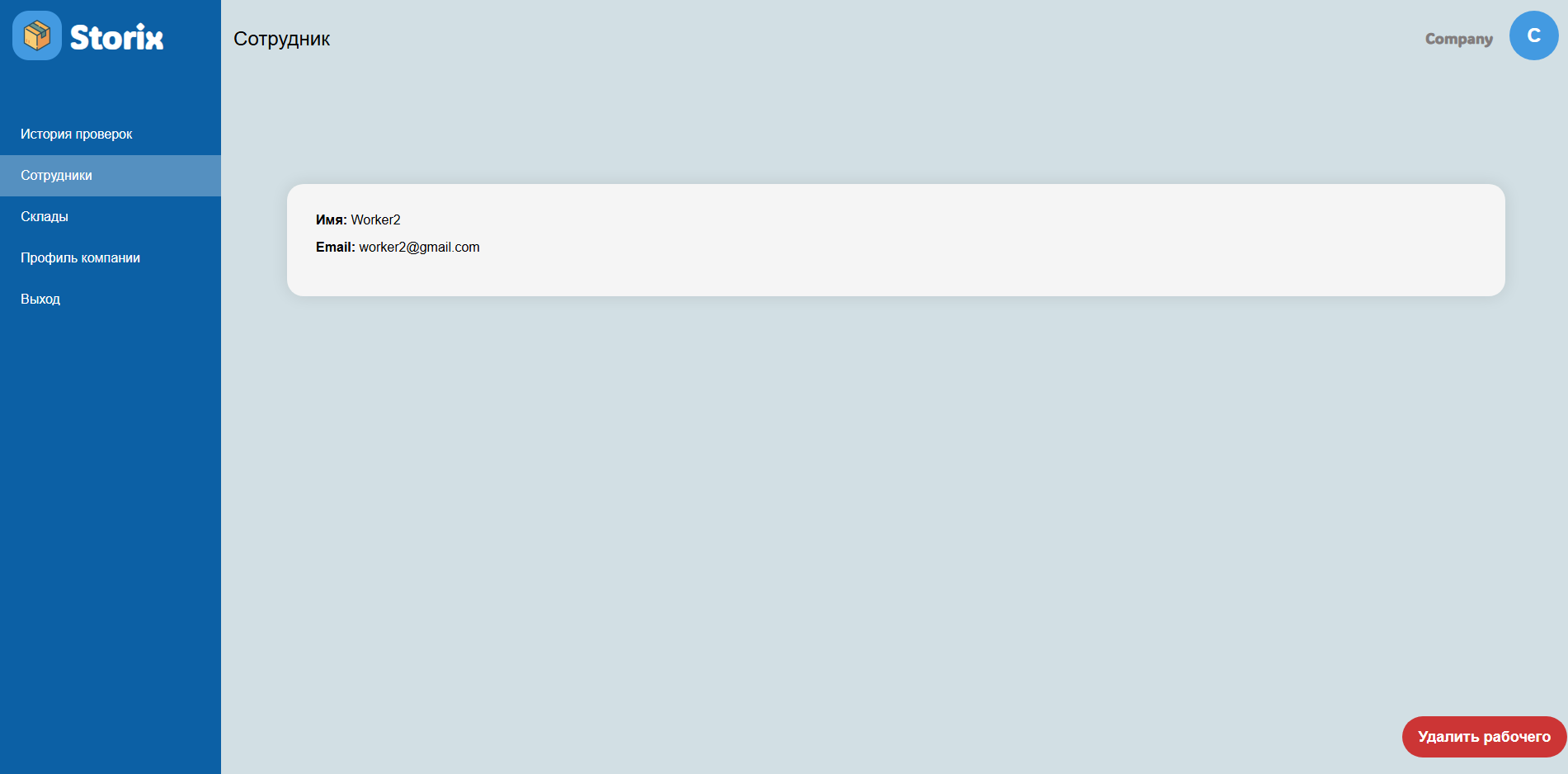
1. Страница с историей конкретной проверки
   * 1. Страницы администратора (компании)

После входа в систему администратор видит список своих сотрудников. На рисунке 16 показана страница с сотрудниками.



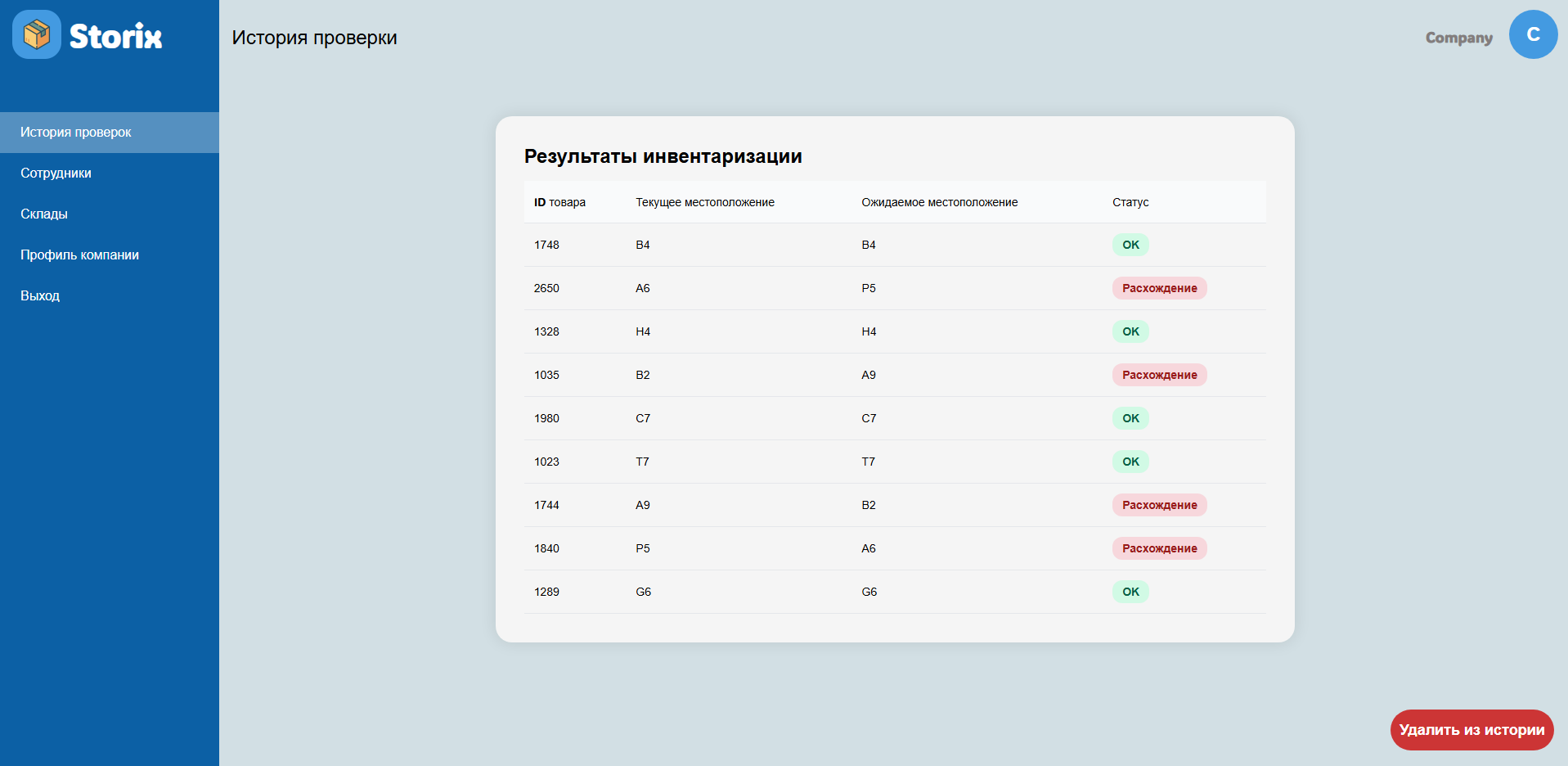
1. Страница со списком сотрудников

На каждую из карт работников можно нажать и перейти на страницу с информацией об этом рабочем. На этой странице администратор может удалить аккаунт рабочего. На рисунке 17 показана страница с информацией о рабочем.



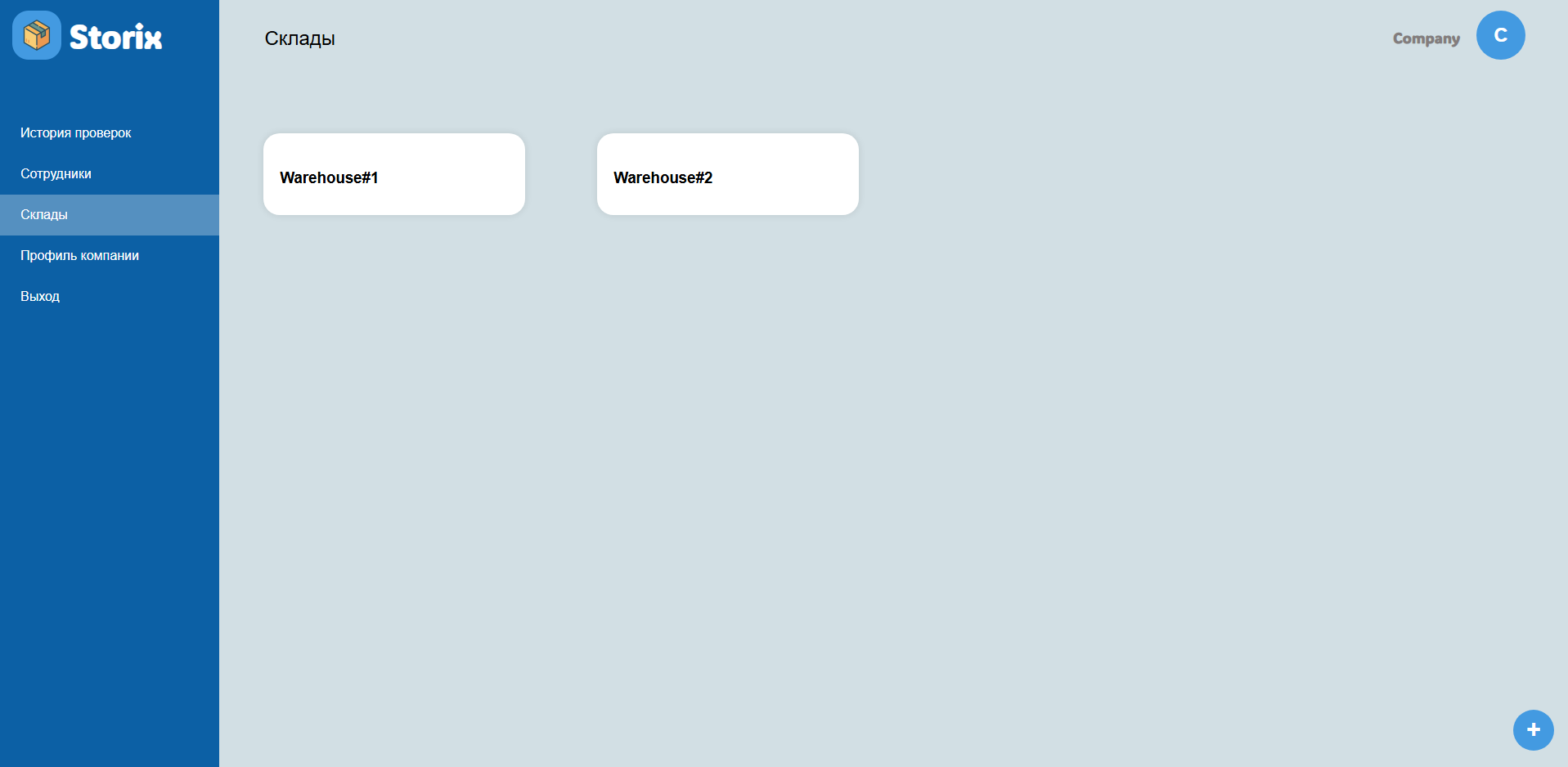
1. Страница с информацией о рабочем

Администратор может, как и рабочий посмотреть историю проверок. Также администратор может удалить запись о проверке. На рисунке 18 показана страница с результатом проверки, которую можно удалить.

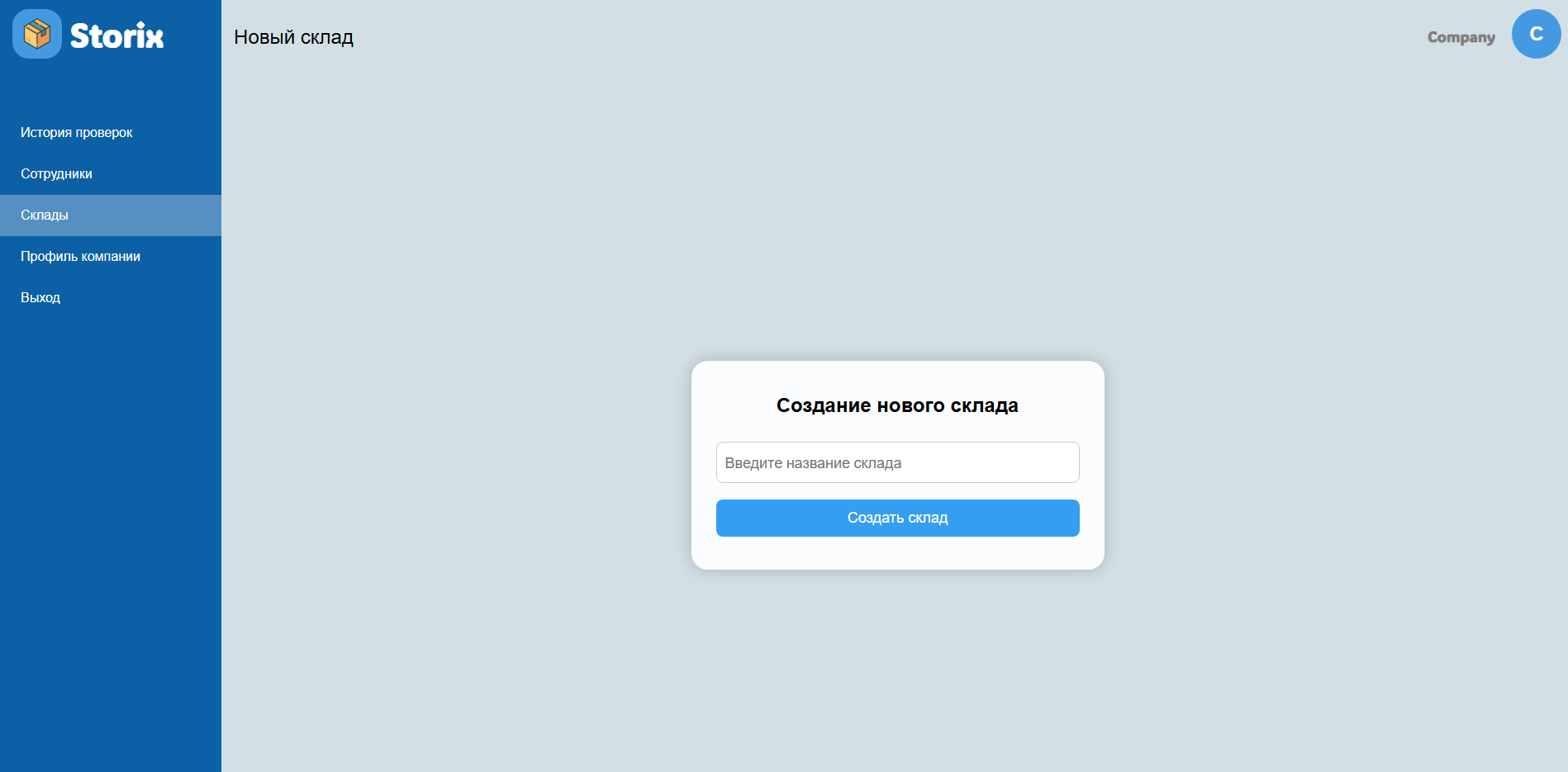


1. Страница с результатом проверки, которую можно удалить

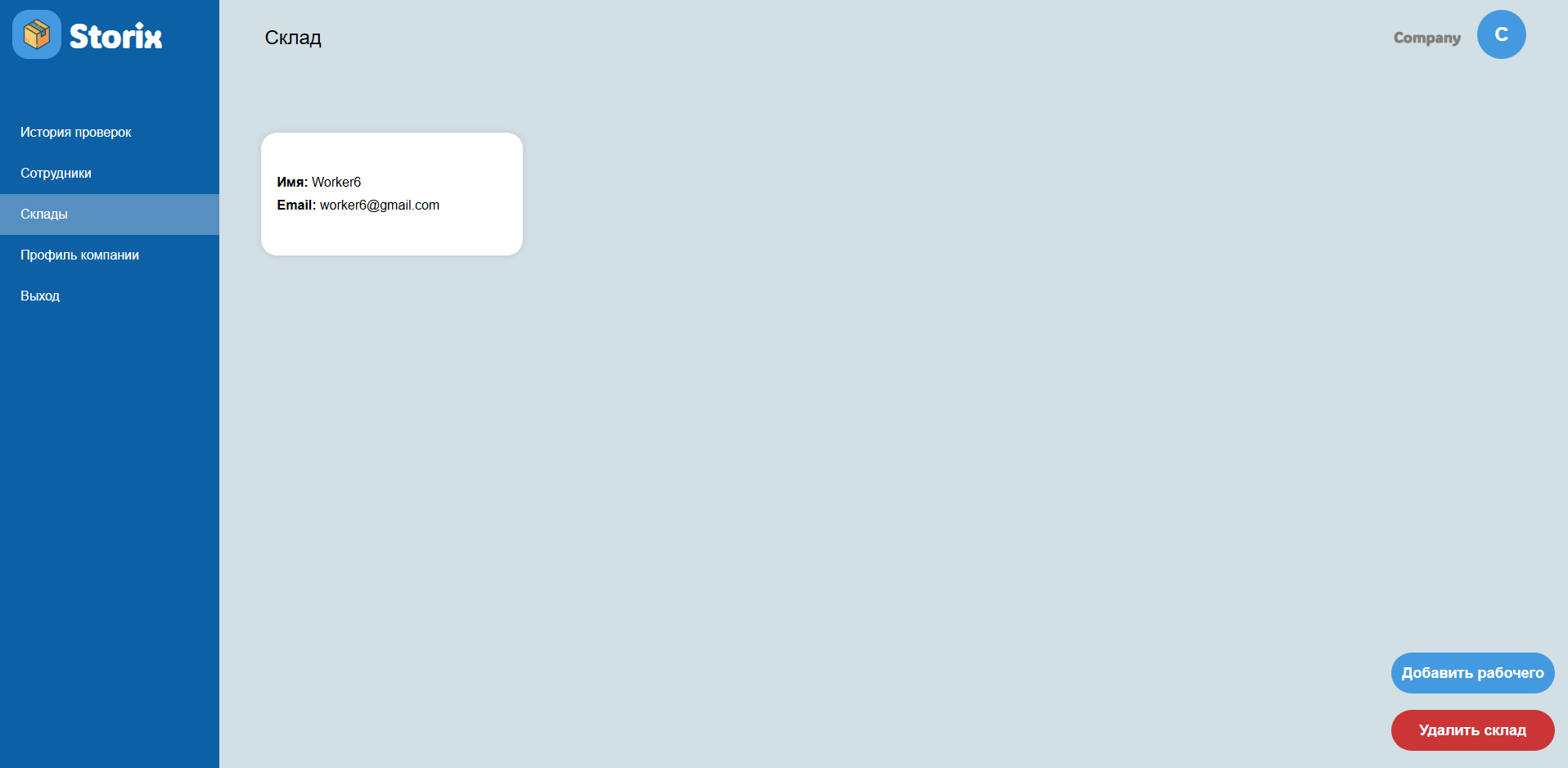
Администратор может посмотреть список своих складов. По кнопке с плюсом он может создать новый склад. Также администратор при нажатии на нужный склад может увидеть список рабочих, назначенных на этот склад, добавить нового рабочего на склад и удалить этот склад. На рисунках 19-21 показаны страницы по работе со складами.



1. Страница со списком складов

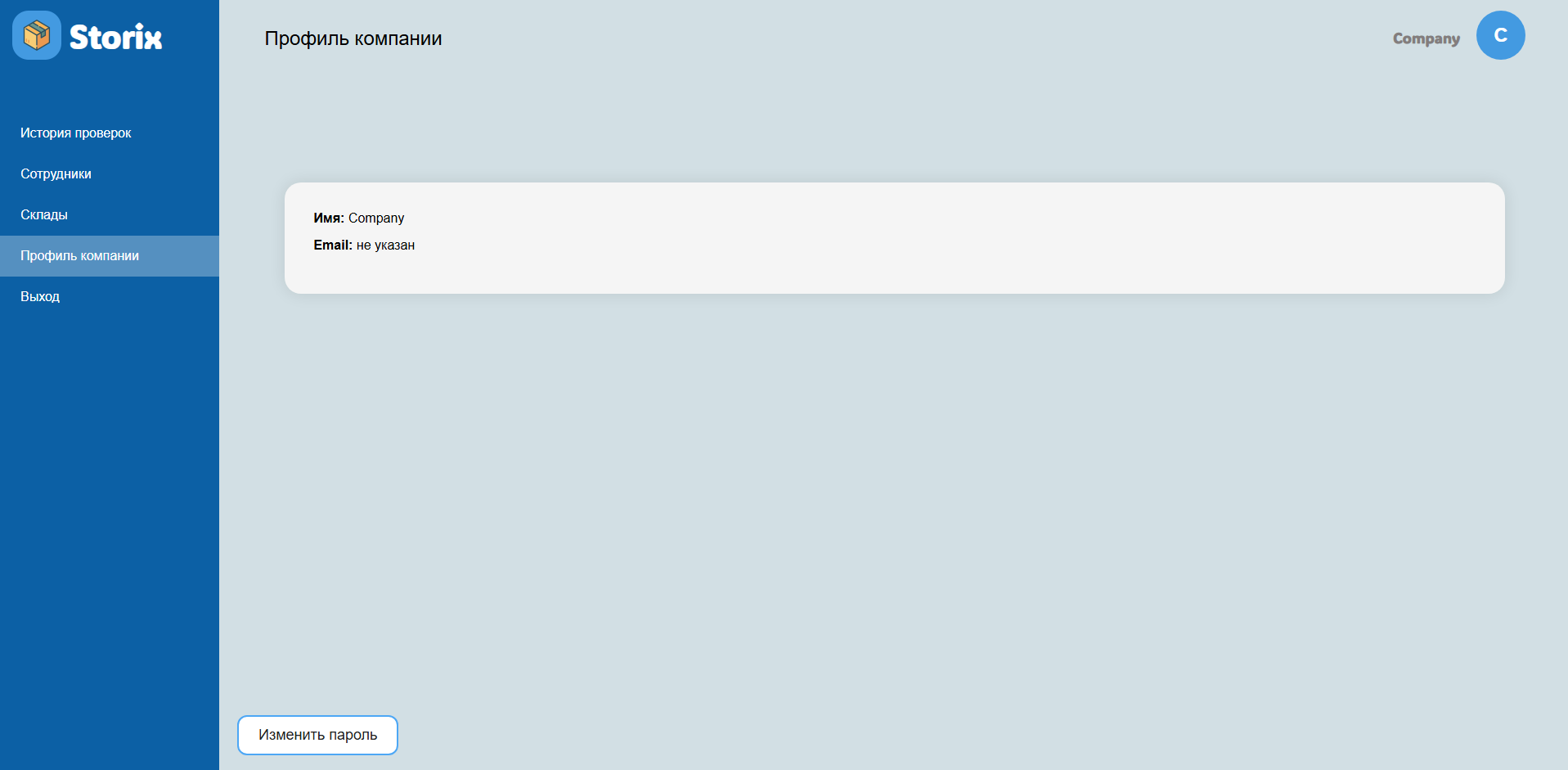


1. Страница с созданием нового склада



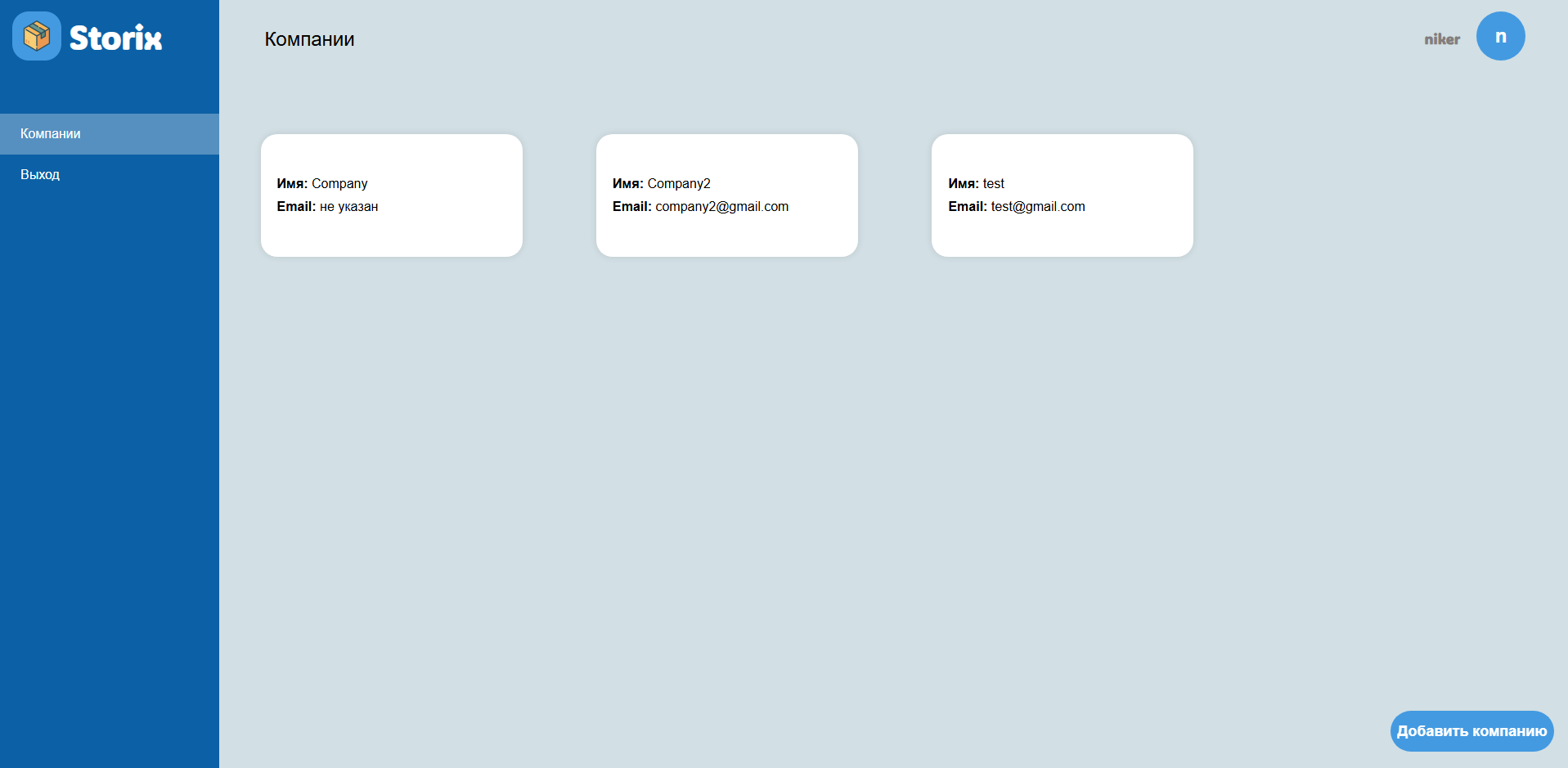
1. Страница с информацией о складе

Администратор может посмотреть профиль компании, а также изменить пароль, который был установлен при создании аккаунта системным администратором. На рисунке 22 показана страница профиля компании.

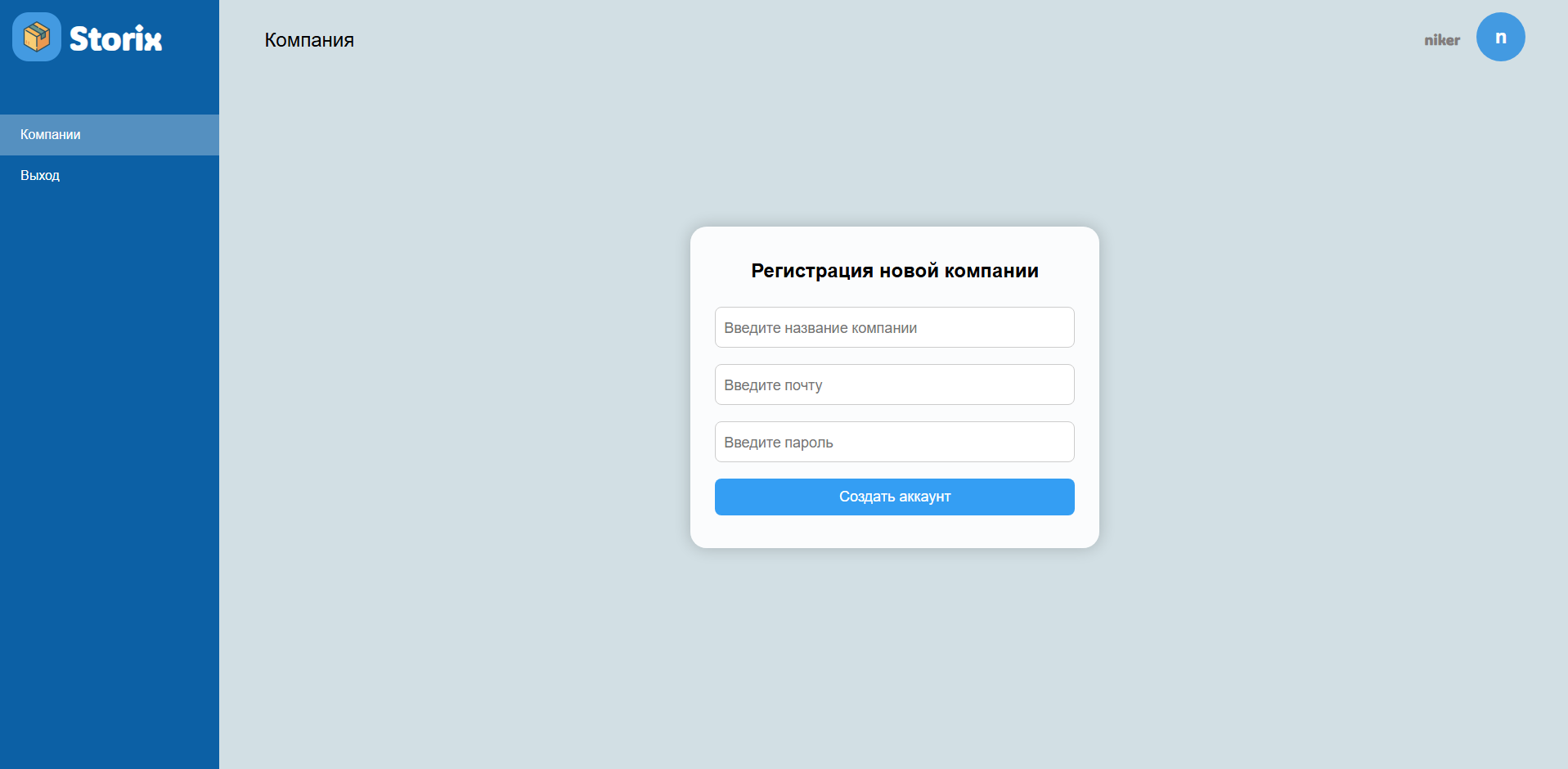


1. Страница профиля компании
   * 1. Страницы системного администратора

Системный администратор, при входе в аккаунт попадает на страницу со списком компаний. Он может добавить новую компанию, а также посмотреть подробнее информацию об уже существующей компании. На рисунках 23-24 показаны страницы со списком компаний и созданием новой компании.

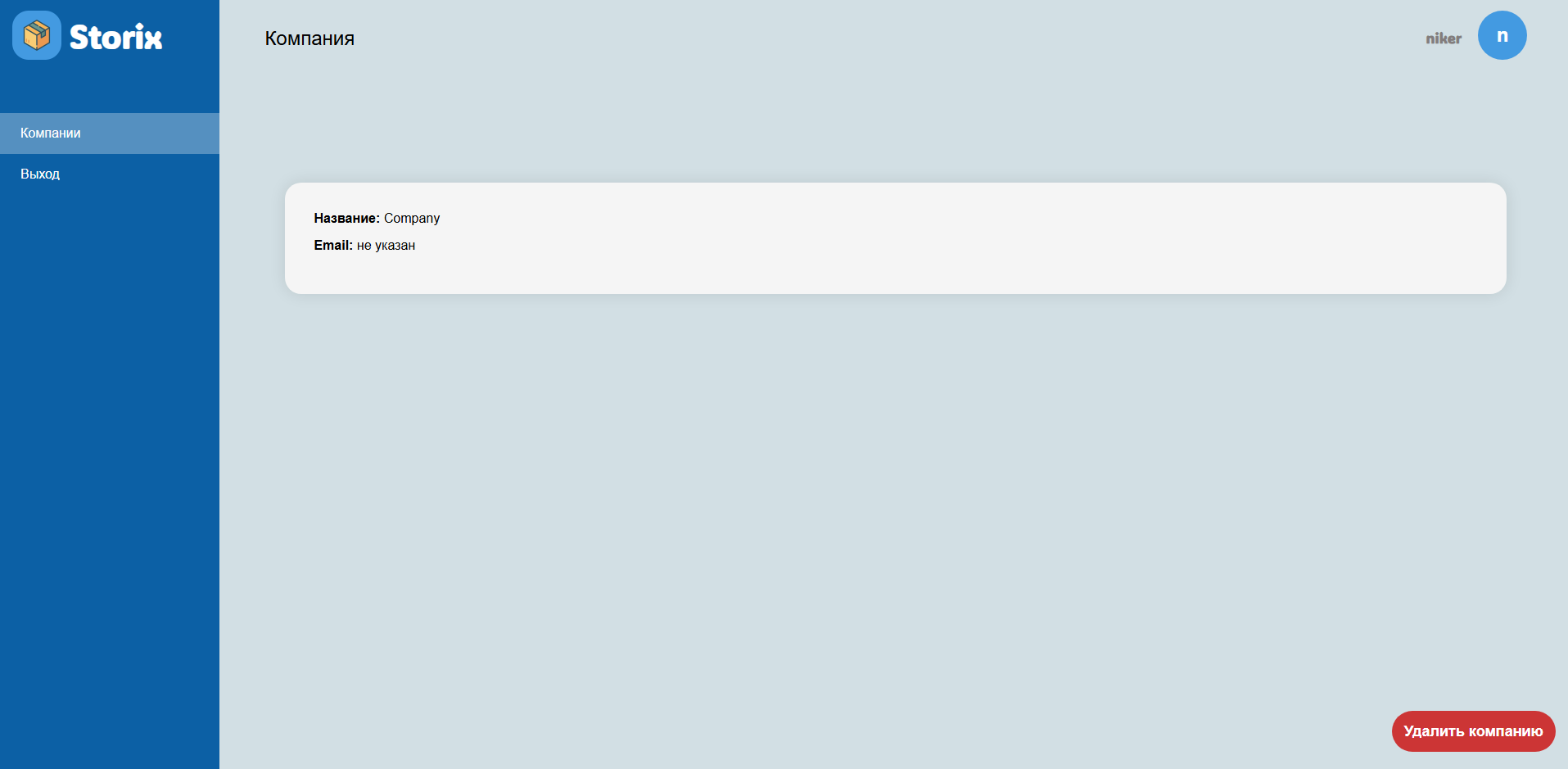


1. Страница со списком компаний



1. Страница создании компании

Системный администратор может удалить любую компанию из списка. На рисунке 25 показана страница с информацией о компании, которую можно удалить.



1. Страница с информацией о компании
2. Тестирование

Для проверки работоспособности разработанного приложения было проведено UI-тестирование.

UI-тестирование (тестирование пользовательского интерфейса) представляет собой процесс оценки графического интерфейса приложения или веб-сайта на соответствие заданным требованиям, удобство взаимодействия, визуальную привлекательность и корректность работы элементов интерфейса. Его основная задача — убедиться, что внешний вид и поведение интерфейса соответствуют ожиданиям, а пользователи могут без затруднений пользоваться функционалом приложения или сайта.

В таблицах 2-5 продемонстрированы результаты UI-тестирования пользовательского интерфейса для всех ролей.

1. Результаты UI-тестирования для неавторизованного пользователя

|  |  |  |
| --- | --- | --- |
| Тестовый сценарий | Ожидаемый результат | Статус теста |
| Нажатие на кнопку «Узнать больше» | Переход на страницу с информацией о сайте | Пройден |
| Нажатие на кнопку «Вернуться» | Переход на главную страницу | Пройден |
| Нажатие на кнопку «Войти» | Переход на страницу авторизации | Пройден |
| Нажатие на кнопку «Войти» на странице авторизации с незаполненными полями | Вывод ошибки | Пройден |
| Нажатие на кнопку «Войти» на с правильно заполненными полями аккаунта рабочего | Переход на страницу инвентаризации | Пройден |
| Нажатие на кнопку «Войти» на с правильно заполненными полями аккаунта администратора | Переход на страницу с сотрудниками | Пройден |
| Нажатие на кнопку «Войти» на с правильно заполненными полями аккаунта системного администратора | Переход на страницу с компаниями | Пройден |

1. Результаты UI-тестирования для рабочего

|  |  |  |
| --- | --- | --- |
| Нажатие на кнопку «Выбрать видео» | Открытие окна с выбором файла | Пройден |
| Нажатие на кнопку «Выбрать JSON» | Открытие окна с выбором файла | Пройден |
| Нажатие на кнопку «Отправить» на странице инвентаризации без загрузки видео или json | Вывод ошибки | Пройден |
| Нажатие на кнопку «Отправить» на странице инвентаризации с загруженными видео или json | Отображение результата проверки | Пройден |
| Нажатие на кнопку «История проверок» в боковом меню | Переход на страницу с историей проверок | Пройден |
| Нажатие на проверку из истории проверок | Переход на страницу с результатом проверки | Пройден |
| Нажатие на кнопку «Выход» | Переход на главную страницу | Пройден |

1. Результаты UI-тестирования для администратора

|  |  |  |
| --- | --- | --- |
| Нажатие на ячейку одно из сотрудников | Переход на страницу с информацией о работнике | Пройден |
| Нажатие на кнопку «Удалить рабочего» | Переход на страницу со списком рабочих, а также удаление рабочего | Пройден |
| Нажатие на кнопку «Склады» в боковом меню | Переход на страницу со списком складов | Пройден |
| Нажатие на кнопку с плюсом | Переход на страницу с созданием нового склада | Пройден |
| Попытка создать склад без заполнения поля | Вывод ошибки | Пройден |
| Нажатие на ячейку со складом | Переход на странице этого склада с отображением списка рабочих | Пройден |
| Нажатие на кнопку «Добавить рабочего» | Переход на страницу с полями для создания аккаунта рабочего | Пройден |
| Нажатие на кнопку «Создать аккаунт» без заполнения полей | Вывод ошибки | Пройден |
| Нажатие на кнопку «Удалить склад» | Переход на страницу со складами, а также удаление склада | Пройден |
| Нажатие на кнопку «Профиль компании» | Переход на страницу с профилем компании | Пройден |
| Нажатие на кнопку «Изменить пароль» | Отображение полей для ввода нового пароля | Пройден |
| Нажатие на кнопку «Выход» | Переход на главную страницу | Пройден |

1. Результаты UI-тестирования для системного администратора

|  |  |  |
| --- | --- | --- |
| Нажатие на кнопку «Добавить компанию» | Переход на страницу с полями для создания аккаунта компании | Пройден |
| Попытка создать компанию без заполнения полей | Вывод ошибки | Пройден |
| Нажатие на ячейку одной из компаний | Переход на страницу с информацией о компании | Пройден |
| Нажатие на кнопку «Удалить компанию» | Переход на страницу со списком компаний, а также удаление компаний | Пройден |
| Нажатие на кнопку «Выход» | Переход на главную страницу | Пройден |

ЗАКЛЮЧЕНИЕ

В ходе данной работы были выполнены все поставленные цели. Мы изучили предметную область и рассмотрели существующие решения поставленной проблемы.

В результате был реализован сайт для проведения инвентаризаций «Storix», основная функциональность которого включает:

* вход;
* проведение инвентаризации;
* просмотр истории инвентаризаций;
* добавление/удаление аккаунтов рабочих;
* добавление/удаление складов;
* редактирование пароля аккаунта компании;
* добавление/удаление аккаунтов компаний.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дронов, В. А. React 19. Разработка веб-приложений на JavaScript / В. А. Дронов. – Москва : БХВ-Петербург, 2025. – 360 с. – ISBN 978-5-9775-2036-2. – Текст : непосредственный.
2. Документация React [Электронный ресурс]. – Режим доступа: https://ru.legacy.reactjs.org/docs/getting-started.html/ – Заглавие с экрана. – (Дата обращения 03.06.2025).
3. Документация Django [Электронный ресурс]. – Режим доступа: https://docs.djangoproject.com/en/5.0/ – Заглавие с экрана. – (Дата обращения 03.06.2025).