**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”  
(УНИВЕРСИТЕТ ИТМО)**

**ЦЕНТР АВТОРИЗОВАННОГО ОБУЧЕНИЯ ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ**

**ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА**

Разработка программного обеспечения

для контроля работы и учета складской техники

Автор Банников Роман Тимофеевич \_\_\_\_\_\_\_\_\_\_\_

(Фамилия Имя Отчество) (Подпись)

**Центр авторизованного обучения информационным технологиям**

Наименование программы **«Python-разработчик»**

Руководитель ст. преподаватель Кузьмин Константин Михайлович \_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

**К защите допустить**

|  |  |  |
| --- | --- | --- |
| *Заместитель директора ЦАО ИТ,*  *к.т.н.* |  | / *Т.В. Зудилова*/ |

Санкт-Петербург, 2023г.

Обучающийся\_\_\_\_\_\_\_\_\_\_ Банников Роман Тимофеевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Группа 124/29

(Фамилия, И. О.)

Работа принята «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Работа выполнена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Секретарь ИАК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, И. О.) (подпись)

Листов хранения \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Демонстрационных материалов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОГЛАВЛЕНИЕ**

|  |  |
| --- | --- |
| 1. Введение………………………………………………………………….. | 4 |
| 2. Цель………………………………………………………………………. | 5 |
| 3. Описание…………………………………………………………………. | 5 |
| 3.1 Концепция………………………………………………………………. | 5 |
| 3.2 Бизнес-требования…………………………………………..…………. | 6 |
| 3.3 Пользовательские требования ………………………………………… | 6 |
| * 1. Функциональные требования……………………………….…………. | 7 |
| 3.5 UML…………………………………………………….…….…………. | 8 |
| 4. Реализация…………………………………………………….…………. | 9 |
| 4.1 Python……………………………………………………………………. | 9 |
| 4.2 PyQt5……………………………………………………………….……. | 10 |
| 4.3 Socket……………………………………………………………………. | 10 |
| 4.4 Threading………………………………………………………..………. | 10 |
| 4.5 OpenCV……………………………………………………….…………. | 10 |
| 4.6 Psycopg………………………………………………………….………. | 10 |
| 4.7 QtDesigner………………………………………………………………. | 10 |
| * 1. PgAdmin………………………………………………………………. | 11 |
| 4.9 MySQL Workbench……………………………………………..….…… | 11 |
| 5. Функции системы………………………………………………..….…… | 15 |
| 6. Структура данных……………………………………………….………. | 20 |
| 8. Развертывание……………………………………………………………. | 22 |
| 7. Тестирование…………………………………………………………….. | 23 |
| Заключение………………………………………………………….………. | 24 |
| Литература…………………………………………………………..………. | 25 |

# Введение

Современные предприятия используют большое количество складской техники, такой как погрузчики, штабелеры, электрокары и т.д. Однако учет и контроль за их использованием являются сложными задачами, которые требуют постоянного мониторинга. Недостаточный контроль за использованием складской техники может привести к серьезным проблемам, таким как повреждения оборудования, нерациональное использование ресурсов и дополнительные расходы на ремонт и замену техники. Для решения этих задач было разработано программное обеспечение, которое позволяет контролировать работу и учет складской техники.

Краткое описание программного обеспечения:

Основной задачей программного обеспечения для контроля работы и учета складской техники является автоматизация учета и контроля за работой складской техники. Для этого используются различные функции, такие как мониторинг действий оператора-водителя, учет выполненных ремонтных работ, и т.д.

Обоснование выбора ПО:

Выбор программного обеспечения для контроля работы и учета складской техники основан на следующих критериях:

- Функциональность: программное обеспечение должно обладать всеми необходимыми функциями для учета и контроля за работой складской техники.

- Надежность: программное обеспечение должно быть надежным и не должно вызывать сбоев или ошибок при работе.

- Простота использования: программное обеспечение должно быть простым в использовании и не требовать особых навыков для работы с ним.

- Масштабируемость: программное обеспечение должно быть масштабируемым и способным обрабатывать большой объем данных в случае необходимости.

# 2. Цель

Цель данного проекта заключается в разработке и внедрении программного обеспечения для контроля и учета работы складской техники, увеличение эффективности, сокращение затрат на ее обслуживание и ремонт, а также повышение безопасности на складе за счет более эффективного контроля за использованием техники.

Для достижения поставленной цели необходимо изучить особенности работы складской техники, определить требования к программному обеспечению и выбрать наиболее подходящие библиотеки доступные *Python* для решения данной задачи.

# 3. Описание ПО

*WTCS (Warehouse Technique Control Software)* **–** программное обеспечение для контроля работы и учета складской техники.

Основной целью данного программного обеспечения является оптимизация работы складской техники для эффективного и детального управления.

Программное обеспечение (далее ПО) включает в себя такие функции как:

- мониторинг состояния техники;

- коммуникацию между оператором техники и руководством;

- видеоконтроль за состояние оператора и груза;

- контроль выполненных заданий оператора;

- учет по выполненным ремонтным работам над техникой;

- инвентаризационный учет.

**3.1 Концепция**

**«**Концепция **~** vision **—** утверждение, описывающее стратегический принцип конечной цели и формы новой системы.» [1]

Концепция данного программного обеспечения опирается на предоставления работникам склада простого и удобного интерфейса для отслеживания функционирования складской техники.

**3.2 Бизнес-требования**

«Бизнес-требования ~ business requirement — объем информации, который в совокупности описывает потребность, которая инициирует один или больше проектов, призванных предоставить решение и получить требуемый конечный бизнесрезультат.» [1]

Основными требованиями к ПО «WTCS»являются:

- увеличение производительность выполнения задач, связанных со складской техникой.

- обеспечение контроля над сотрудниками склада во избежание нарушений требований склада.

- сокращение потерь, связанных с обслуживанием техники.

- оптимизация коммуникации между сотрудниками склада.

**3.3 Пользовательские требования**

«Пользовательское требование ~ user requirement — цель и задача, которую пользователи должны иметь возможность выполнять с системой, или положения об ожиданиях пользователей о качестве системы.» [1]

Пользователями ПО «WTCS»будут начальник колонны, оператор-водитель техники, мастер участка.

Далее будут описаны пользовательские истории:

- Начальник колонны: как начальник колоны я хочу видеть показатели техники, внешнее состояние оператора-водителя и груза, выполненные задания оператором-водителем, а также возможность общения с оператором-водителем и мастером участка.

- Оператор-водитель: как оператор-водитель я хочу видеть груз, возможность записи и отслеживания выполненных заданий, а также возможность общения с начальником колоны и мастером участка.

-Мастер участка: как мастер участка я хочу видеть показатели техники, внешнее состояние оператора-водителя и груза, возможность запись и просмотр по выполненным ремонтным работам над техникой, запись и просмотр наличия и количества техники, а также возможность общения с начальником колоны и мастером участка.

**3.4 Функциональные требования**

«Функциональные требования описывают ожидаемое поведение системы при определенных условиях и действия, которые система позволит выполнять пользователям.» [1]

Функциональные требования ПО «WTCS»представлены в таблице 1

Таблица 1 Функциональные требования

|  |  |  |  |
| --- | --- | --- | --- |
| ***№*** | ***Требование*** | ***№*** | ***Задачи*** |
| 1 | Открываться в собственном окне | 1.1 | Создать графический интерфейс |
|  |  | 1.2 | Создать исполняемый файл |
| 2 | Регистрация пользователей | 2.1 | Создать базу данных для хранения данных пользователя |
|  |  | 2.2 | Создать форму регистрации |
| 3 | Вход в систему | 3.1 | Создать форму входа при запуске исполняемого файла |
|  |  | 3.2 | В случае неверно введенных данных показывать ошибку |
| 4 | Отображение показаний датчиков | 4.1 | Открыть файл с данными |
|  |  | 4.2 | Обработать данные |
|  |  | 4.3 | Отобразить в интерфейсе |
| 5 | Доступ к камерам видеонаблюдения за оператором-водителем и товаром | 5.1 | Получить доступ к видеокамере |
|  |  | 5.2 | Обработать данные с видеокамеры |
|  |  | 5.3 | Отобразить видео в интерфейсе |
| 6 | Отображение задание для оператора-водителя | 6.1 | Создать форму для регистрации задания |
|  |  | 6.2 | Хранить выполненные задания в базе данных |
|  |  | 6.3 | Отобразить задание в интерфейсе |
| 7 | Возможность общаться через ПО | 7.1 | Создать чат |
|  |  | 7.2 | Обеспечить чат многопоточностью |
| 8 | Учет выполненных работ по обслуживанию техники | 8.1 | Создать форму для регистрации выполненных работ |
|  |  | 8.2 | Хранить выполненные работы в базе данных |
|  |  | 8.3 | Отображать выполненные работы |
| 9 | Учет техники | 9.1 | Создать форму для регистрации техники |
|  |  | 9.2 | Запись в базу данных |
|  |  | 9.3 | Отображать наличие и количество в интерфейсе при запросе |

**3.5 UML**

«UML (Unified Modeling Language) — набор стандартной нотации для создания различных визуальных моделей систем, особенно в объектно-ориентированном программировании.» [1]

«Диаграмма вариантов использования ~ use case diagram — модель анализа с указанием действующих лиц, которые могут взаимодействовать с системой для выполнения задач, и различные варианты использования, в которых может участвовать действующее лицо.» [1]

UML диаграмма вариантов использования (use-case diagram) представлена на рисунке 1.

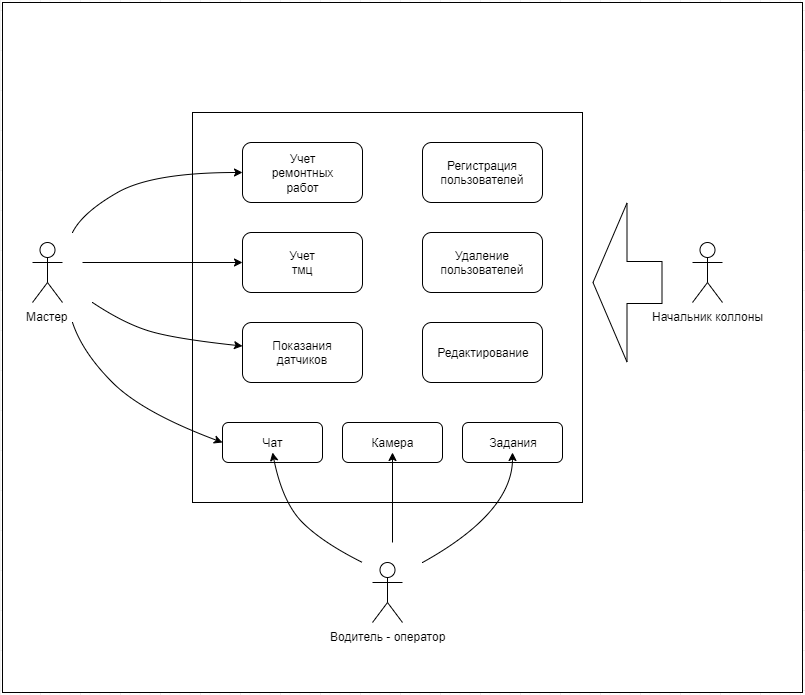


Рисунок 1. UML диаграмма вариантов использования

# 4. Реализация

Для реализации поставленной задачи необходимо использовать следующие технические средства:

* Персональный компьютер
* Терминалы (планшеты) для установки в кабине оператора
* Видеокамеры в кабине оператора и на вилах погрузчика
* Сервер для хранения и обработки данных

Для разработки программного обеспечения был использован язык программирования *Python,* система управления базами данных *Postgresql,* библиотеки *Python:* *OpenCV* для обработки данных с видеокамер*, threading* для запуска на сервере отдельных потоков обработки пакетов данных*, PyQt5* для создания графического интерфейса*, socket* для передачи данных*, psycopg2* для интеграции *Postgresql* в *Python,* программные обеспечения: *Qt Designer* для эффективного создания графического интерфейса*, pgAdmin* для быстрого создания и доступа к базе данных*, MySQL Workbench* для создание диаграммы базы данных*, IDLE PyCharm* для написания и редактирования кода*.*

*4.1 Python* - это высокоуровневый интерпретируемый язык программирования, который был разработан в конце 1980-х годов Гвидо ван Россумом. Он является одним из самых популярных языков программирования в мире, благодаря своей простоте, легкости в изучении и обширной библиотеке.

*Python* поддерживает различные парадигмы программирования, включая процедурное, объектно-ориентированное и функциональное программирование. Он также имеет динамическую типизацию, что означает, что типы данных могут быть определены во время выполнения программы, а не только на этапе компиляции.

*Python* используется в различных областях, включая научные исследования, машинное обучение, веб-разработку, анализ данных, автоматизацию задач и другие. Он имеет обширную библиотеку, которая предоставляет множество инструментов для решения различных задач. Кроме того, *Python* имеет огромное сообщество разработчиков, которые создают и поддерживают множество библиотек и инструментов для этого языка*.*

*4.2 PyQt5* - это библиотека, которая предоставляет возможность создания графических пользовательских интерфейсов на языке *Python*. Она включает в себя набор инструментов для создания различных элементов интерфейса, таких как кнопки, окна, таблицы, меню и другие. *PyQt5* базируется на библиотеке *Qt*, которая используется для разработки многоплатформенных приложений.

*4.3 Socket* - это библиотека, которая используется для создания сетевых приложений. Она позволяет устанавливать соединение между компьютерами и передавать данные между ними. *Socket* может использоваться для создания различных типов сетевых приложений, включая клиент-серверные приложения, мессенджеры, игры и другие.

*4.4 Threading* - это библиотека, которая предоставляет инструменты для создания многопоточных приложений. *Threading* позволяет запускать несколько процессов в рамках одного приложения, что может повысить его производительность. *Threading* также предоставляет механизмы для синхронизации работы нескольких потоков и управления доступом к общим ресурсам.

*4.5 OpenCV* - это библиотека, которая используется для обработки изображений и видео. Она предоставляет множество функций для работы с изображениями, таких как фильтрация, морфологические операции, сегментация и другие. *OpenCV* также имеет возможность работать с видео, анализировать его и извлекать информацию из него.

*4.6 Psycopg* - это библиотека, которая используется для работы с базой данных *PostgreSQL* на языке *Python*. Она предоставляет интерфейс для подключения к базе данных, выполнения запросов и получения данных из нее. *Psycopg* поддерживает множество возможностей *PostgreSQL*, включая хранение данных в различных форматах, использование транзакций и другие.

*4.7 QtDesigner* - это инструмент для создания пользовательских интерфейсов для приложений на основе библиотеки *Qt*. Он позволяет создавать графические элементы интерфейса, располагать их на форме, настраивать их свойства и взаимодействовать с кодом на языке программирования *Python*, используя библиотеку *PyQt*.

*4.8 PgAdmin* - это инструмент управления базами данных *PostgreSQL*. Он предоставляет графический интерфейс для создания и управления базами данных, таблицами, индексами, пользователями, правами доступа и другими объектами. *PgAdmin* также позволяет выполнять *SQL*-запросы, создавать отчеты и мониторить работу сервера *PostgreSQL*.

*4.9 MySQL Workbench* - это интегрированная среда разработки для работы с базами данных *MySQL*. Он предоставляет графический интерфейс для создания и управления базами данных, таблицами, индексами, пользователями и правами доступа. *MySQL Workbench* также позволяет выполнять *SQL*-запросы, создавать отчеты и мониторить работу сервера *MySQL*. Для создания диаграммы базы данных была использована данная среда разработки.

Видеокамеры устанавливаются на вилах как представлено на Рисунке 2.



Рисунок 2. Места установки камер.

Система состоит из следующих компонентов:

* Видеокамеры, установленные в кабине и на вилах погрузчика
* Терминалы (планшеты) в кабине погрузчика
* Персональный компьютер с установленным программным обеспечением
* В качестве сервера используется персональный компьютер
* Программное обеспечение

Описание действий пользователя программы в виде руководства пользователя:

Для работы с программным обеспечением необходимо настроить сервер, для этого необходимо установить СУБД и запустить сервер. В качестве сервера будет использоваться персональный компьютер начальника колоны. Далее необходимо будет установить и настроить программное обеспечение, а именно указать ИП адреса, порты. Также необходимо будет установить программное обеспечение на терминалах (планшетах) в погрузчиках. Подключить к ним исправные видеокамеры.

После запуска откроется главное окно, в котором будут отображаться все данные о складской техники. На главной странице слева будет отображаться чат для коммуникации между водителями-операторами и начальником колоны. Справа будет отображаться поле со вкладками: камеры, задачи, техника, ремонты, персонал.

Пользователь сможет управлять системой с помощью специального пользовательского интерфейса, который будет включать в себя следующие функции:

* Во вкладке камеры пользователям будет открыт доступ к видеокамерам, установленным в кабинах погрузчиков и на вилах.
* Во вкладке задачи пользователям будет доступен календарь с виджетом для записи выполненных заданий.
* Во вкладке техника пользователям будет доступна информация о наличии и количестве учтенной техники, а также возможность добавлять и удалять технику.
* Во вкладке ремонты пользователям будет доступна информация о проведенных ремонтных работах с техникой.
* Во вкладке персонал пользователям будет доступна информация о сотрудниках склада, а также возможность добавлять и удалять пользователей.

Практическая значимость работы:

Разработка программного обеспечения для контроля работы и учета складской техники имеет большую практическую значимость. Данная система позволит оптимизировать работу склада и увеличить эффективность его использования. Благодаря учету движений техники на складе, можно будет определить зоны наибольшей нагрузки и принимать меры для их оптимизации.

Также система позволит оперативно реагировать на аварийные ситуации и проблемы на складе, что повысит безопасность работы персонала и сохранность имущества.

Таким образом, разработка программного обеспечения для контроля работы и учета складской техники является актуальной и востребованной задачей, которая может принести значительные выгоды в управлении складом и повысить его эффективность.

# Функции системы

Программное обеспечение для контроля работы и учета складской техники имеет несколько функций, которые позволяют осуществлять контроль над складской техникой и ее использованием.

* 1. Видеомониторинг: Программные обеспечение обеспечивает возможность подключения камер видеонаблюдения для наблюдения за действиями погрузчика в реальном времени. Благодаря этой функции, операторы могут оперативно реагировать на возможные аварийные ситуации и принимать меры для предотвращения потенциальных проблем. Функция доступна во вкладке «Камеры». Во вкладке есть возможность выбора погрузчика, к камерам которого необходим доступ. Также есть функция сделать скриншот. На Рисунке 3 представлена данная функция.

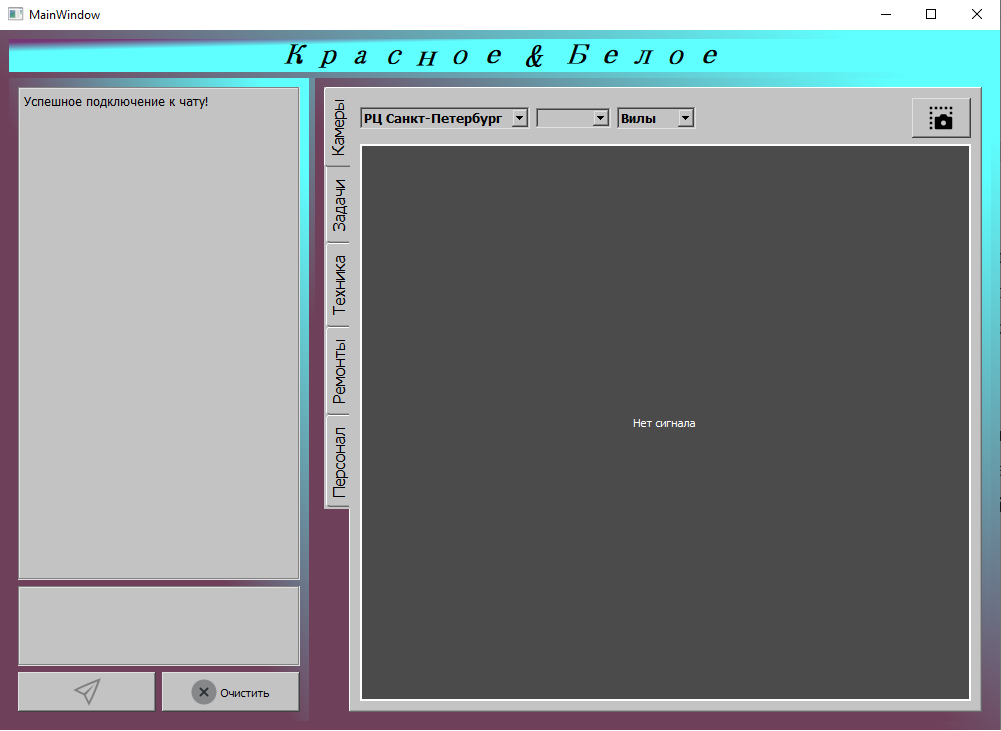


Рисунок 3. Вкладка «Камеры»

* 1. Учет техники: Программные обеспечение позволяет вести учет складской техники. Данная функция доступна во вкладке «Техника». После выбора вкладки, в окне отобразиться информация о наличии техники и данные. Также есть возможность добавлять технику и удалять при необходимости. На Рисунке 4 представлена данная функция.

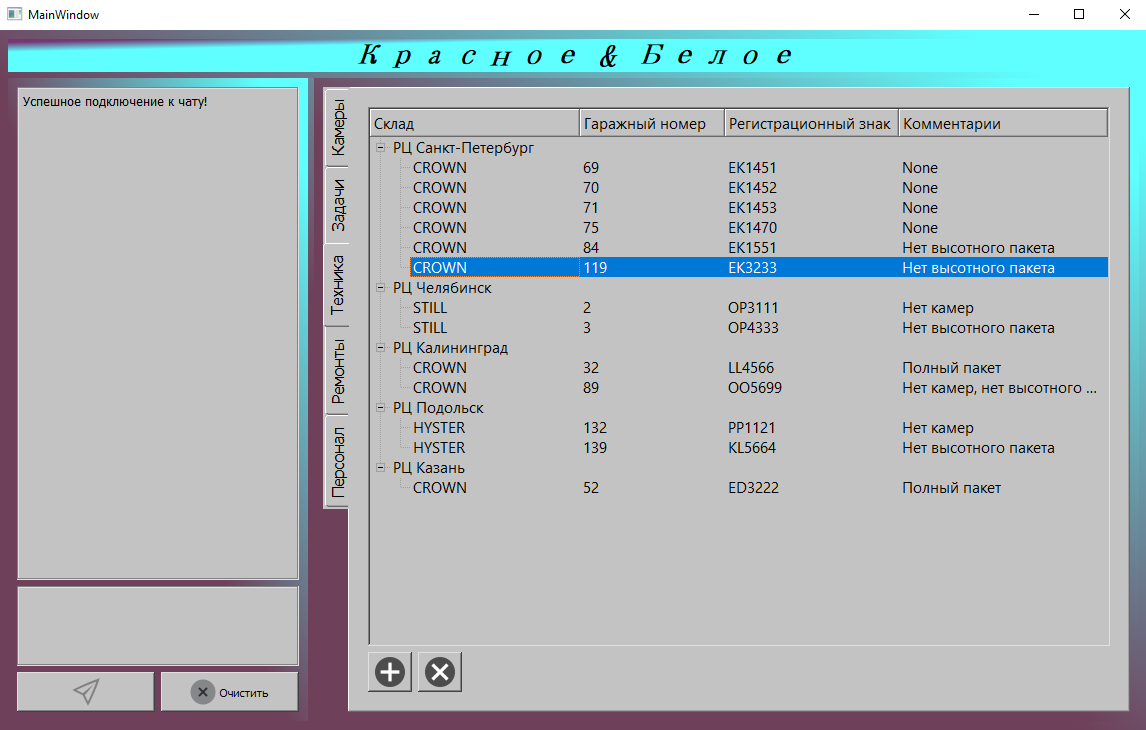


Рисунок 4. Вкладка «Техника»

При нажатии кнопки добавить, будет отображена форма с полями для заполнения. При нажатии кнопки удалить, будет представлена форма с выпадающим списком в котором будет все доступная техника.

* 1. Учет ремонтов: Программные обеспечение позволяет вести учет ремонтов складской техники, записывая даты и описания проведенных работ. Эта функция позволяет мастерам планировать и контролировать ремонтные работы, что увеличивает срок эксплуатации техники и снижает затраты на ее ремонт.

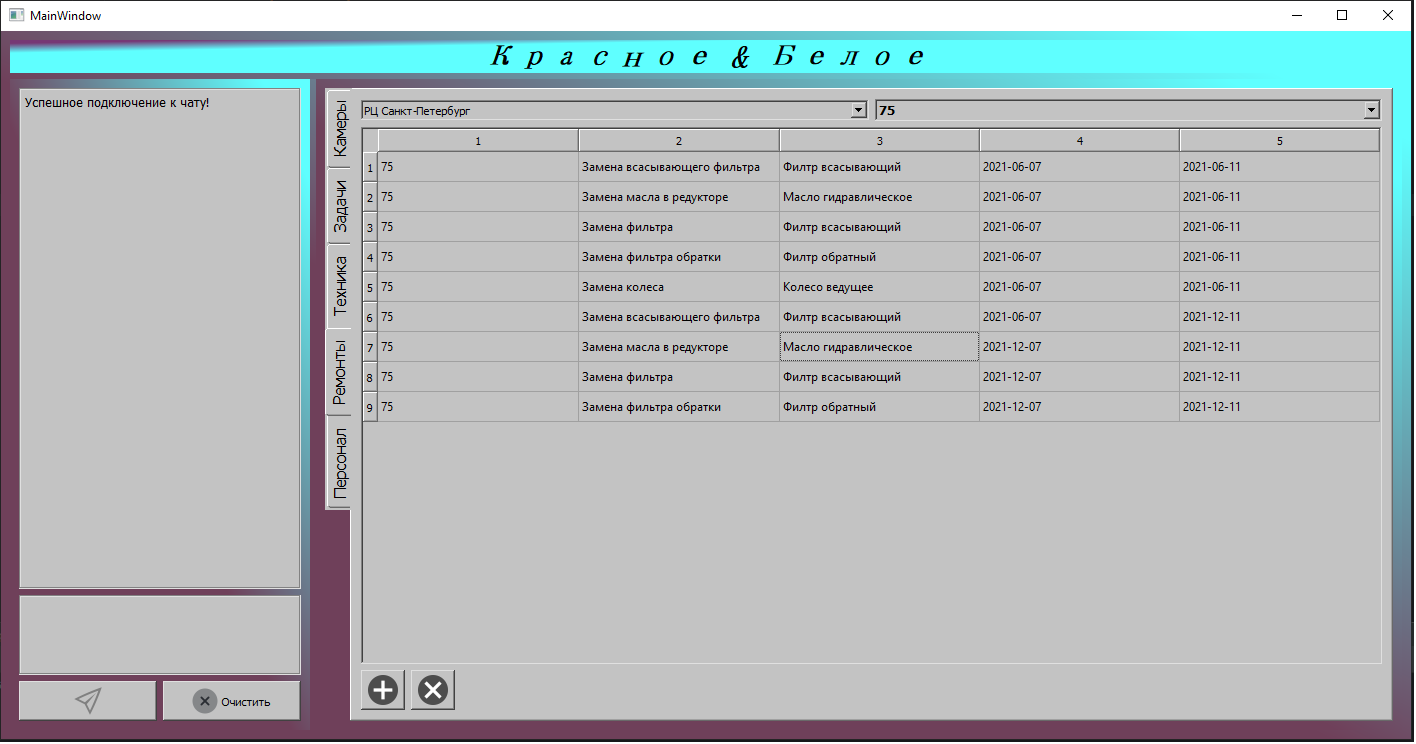


Рисунок 6. Вкладка «Персонал»

При нажатии кнопки добавить, будет отображена форма с полями для заполнения, в которую вноситься информация о выполненных работах, датах, запчастях и тд. При нажатии кнопки удалить, будет представлена форма с выпадающим списком для удаления записи..

* 1. Учет персонала: Программные обеспечение позволяет вести учет персонала, занимающегося эксплуатацией складской техники. При выборе вкладки «Персонал» в окне отобразиться информация о учтённом персонале. Также есть возможность добавлять и удалять персонал при необходимости. На Рисунке 5 представлена данная функция.

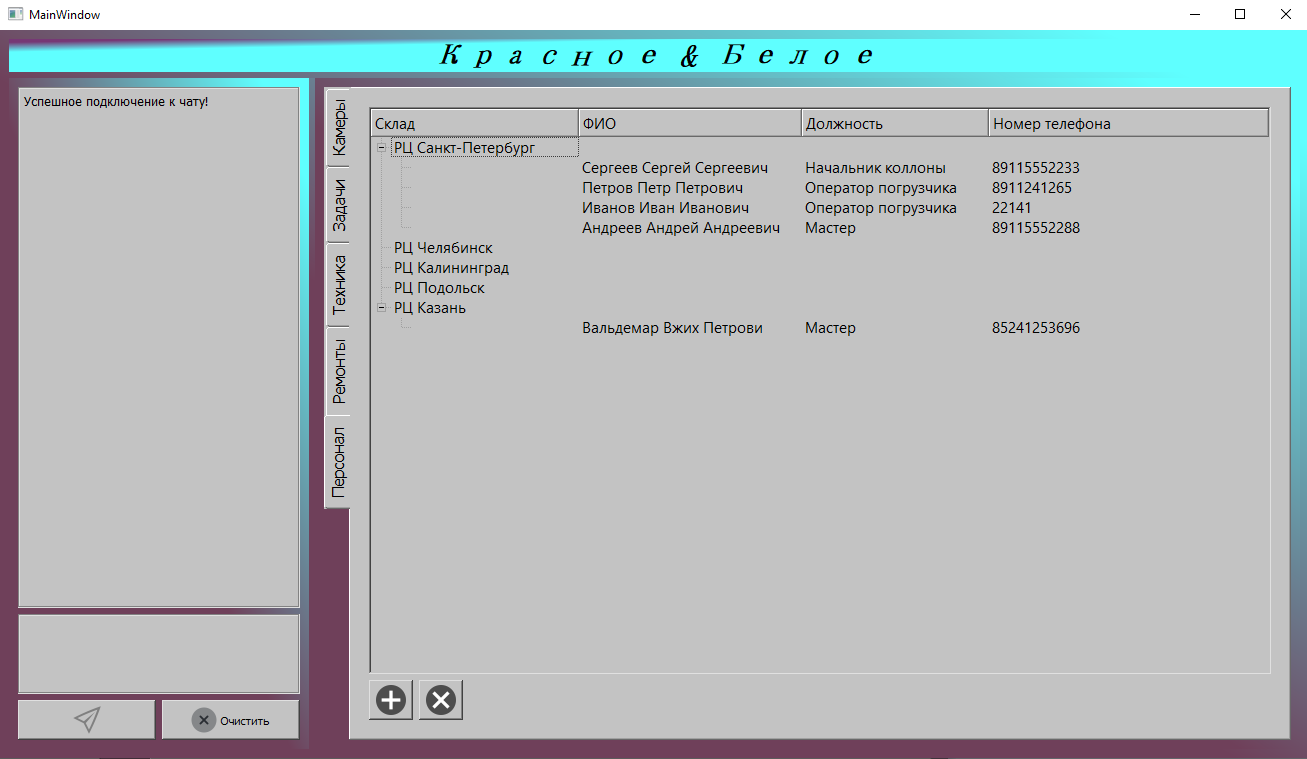


Рисунок 6. Вкладка «Персонал»

При нажатии кнопки добавить, будет отображена форма с полями для заполнения, в которой указываются необходимые данные. При нажатии кнопки удалить, будет представлена форма с выпадающим списком в котором будет весь персонал техника.

* 1. Виджет задачи: Программные обеспечение позволяет вести записи выполненных работ во вкладке «Задачи». Также там имеется календарь, который позволяет операторам отслеживать даты. Данная функция представлена на Рисунке 6.

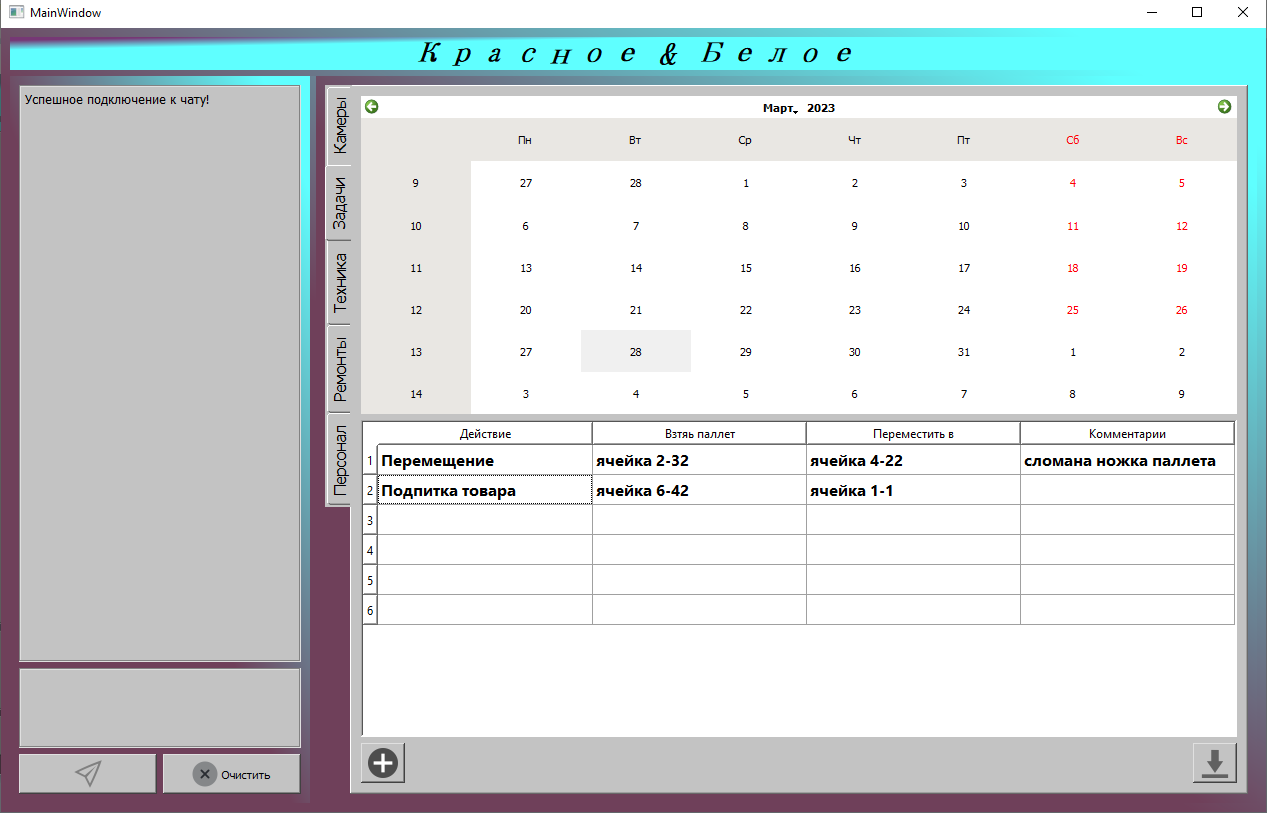


Рисунок 6. Вкладка «Задачи»

Все эти функции системы позволяют увеличить эффективность работы склада, повысить безопасность и контролировать использование складской техники, что делает ее необходимой и важной для многих предприятий и организаций.

# 6. Структура данных

В программном обеспечении для контроля работы и учета складской техники используется база данных *PostgreSQL. PostgreSQL* — это мощная, открытая реляционная СУБД, которая обеспечивает надежное хранение и управление данными.

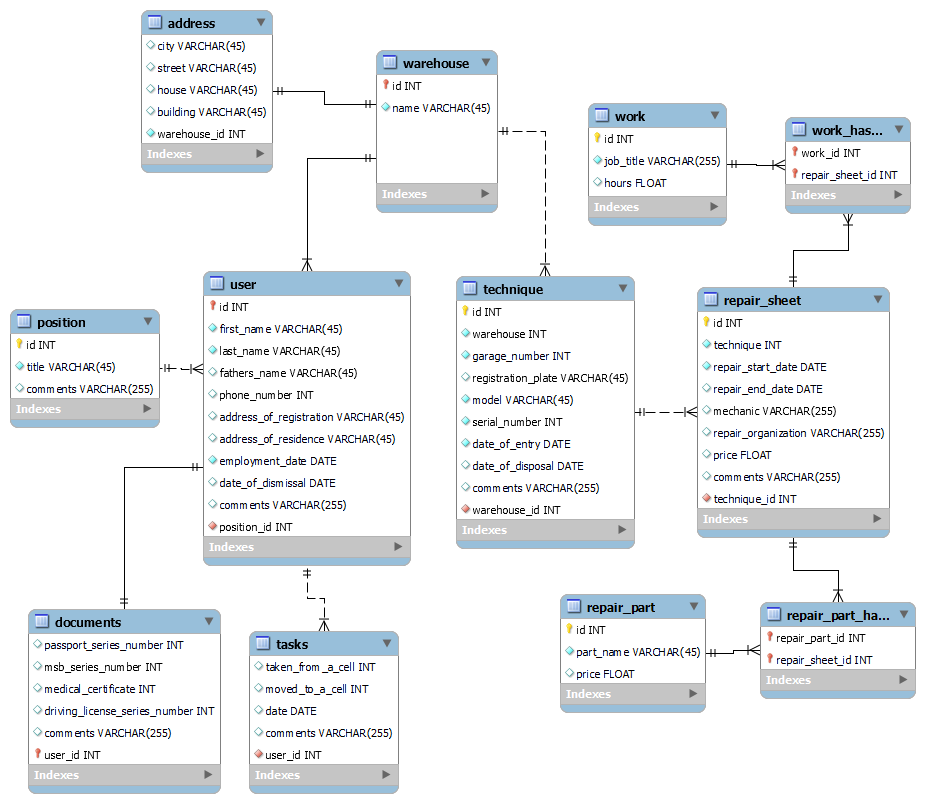
База данных *PostgreSQL* используется для хранения и управления информацией, связанной со складской техникой, такой как данные о перемещениях, ремонтах, использовании и персонале, занимающемся эксплуатацией техники.

В базе данных *PostgreSQL* используется реляционная модель данных, которая обеспечивает связь между таблицами. Для связывания таблиц используются ключи, которые обеспечивают целостность данных и эффективный поиск информации.

*PostgreSQL* предоставляет множество функций и возможностей для обеспечения безопасности и защиты данных, таких как аутентификация, авторизация и шифрование. Также *PostgreSQL* обладает высокой производительностью и масштабируемостью, что позволяет обрабатывать большие объемы данных и поддерживать высокую нагрузку.

В целом, использование базы данных *PostgreSQL* обеспечивает надежность, эффективность и безопасность хранения и управления информацией о складской технике, что делает ее оптимальным выбором для данного программного обеспечения.

На Рисунке 7 представлена структура в виде диаграммы базы данных.

Рисунок 7. Структура базы данных

**8. Развертывание**

Развертывание программного обеспечения - это процесс установки и настройки приложения на целевой системе. Чтобы развернуть наше программное обеспечение на другой машине, мною было использована библиотека *pyinstaller*, который позволяет создавать исполняемые файлы из исходного кода на Python. Для этого нужно выполнить команду "pyinstaller -F main.py", где main.py - это главный файл нашего приложения. Флаг -F позволяет все упаковать в 1 файл EXE.

Для развертывания нашего приложения на другой машине, следует выполнить следующие шаги:

Создать базу данных *PostgreSQL* на сервере, используя *pgAdmin*. В базе данных нужно создать таблицы, которые необходимы для работы нашего приложения.

Запустить приложение на целевой машине, дважды кликнув на созданный исполняемый файл.

Проверить работоспособность приложения. Для этого нужно взаимодействовать с приложением и проверить, что все функции работают корректно.

Развертывание программного обеспечения с использованием исполняемого файла EXE и библиотеки *pyinstaller* - это более удобный и простой способ установки приложения на целевую машину, который позволяет сократить время и упростить процесс установки для конечных пользователей.

Однако, перед развертыванием приложения, необходимо провести тестирование, чтобы убедиться, что оно работает корректно на разных платформах и операционных системах. В тестировании приложения можно использовать автоматические и ручные тесты.

**7. Тестирование**

Тестирование программного обеспечения является важной частью разработки любого ПО. Оно позволяет выявлять ошибки и дефекты в ПО и гарантирует его работоспособность. Существуют различные виды тестирования, такие как:

- модульное тестирование. Оно заключается в тестировании отдельных модулей программы на соответствие требованиям. В этом случае каждый модуль тестируется независимо от других модулей.

- функциональное тестирование. Оно заключается в тестировании функциональных возможностей программы на соответствие требованиям. В этом случае тестируются все функции программы, начиная от базовых функций и заканчивая более сложными.

- интеграционное тестирование. Оно заключается в тестировании взаимодействия модулей программы и их работоспособности в составе программы в целом.

- приемочное тестирование. Оно проводится с целью проверки того, что программа соответствует требованиям заказчика и выполняет все функции, которые были оговорены в техническом задании.

- ручное тестирование — это процесс проверки ПО путем ручного запуска, взаимодействия и проверки различных функций программы, с целью выявления ошибок и дефектов.

В данном проекте мною был использован вид ручного тестирования, так как ручное тестирование является необходимым этапом тестирования, так как некоторые дефекты и ошибки могут быть выявлены только при активном взаимодействии с программой. Ручное тестирование включает в себя проверку пользовательского интерфейса, функциональных возможностей, работоспособности, производительности и безопасности программы.

# Заключение

В результате разработки программного обеспечения для контроля работы и учета складской техники была достигнута цель создания инструмента для автоматизации учета и контроля работы складской техники. Были разработаны функции видеомониторинга, учета техники, учета ремонтов, учета персонала и виджет календаря, которые обеспечивают полный контроль и учет за складской техникой.

Для разработки программного обеспечения были использованы язык программирования *Python*, библиотеки *OpenCV*, *threading*, *PyQt5*, *socket* и ПО *QtDesigner, PgAdmin, MySQL Workbench, PyCharm*. База данных *PostgreSQL* была использована для хранения и управления информацией о складской технике.

Использование данного программного обеспечения позволяет ускорить процессы учета и контроля работы складской техники, а также повысить эффективность и безопасность работы склада в целом. Программное обеспечение может быть использовано на любых складах, что делает его универсальным и востребованным в различных сферах бизнеса.

Таким образом, разработка программного обеспечения для контроля работы и учета складской техники является актуальной и востребованной задачей в современном бизнесе, а разработанное ПО является эффективным инструментом для решения этой задачи

# Литература

1. **К. Вигерс Д. Битти** Разработка требований к программному обеспечению. -.: «Русская Редакция», 2014.
2. **Г.С. Иванова** Технология программирования. -: «Издательство КноРус», 2016.
3. **Билл Любанович** Простой Python -СПб: «Питер», 2016.