Государственное автономное профессиональное образовательное учреждение Мурманской области

«Мурманский колледж экономики и информационных технологий»

УТВЕРЖДАЮ

Начальник учебного отдела

ГАПОУ МО «МКЭиИТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

И.Г. Никитина

«\_\_\_» июня 2022 г.

Учёт хранения грузов в АО «Кипрей»

Наименование темы дипломного проекта

Д И П Л О М Н Ы Й П Р О Е К Т

пояснительная записка

по специальности 09.02.07 «Информационные системы и программирование»

Выпускник Селегененко Юрий Алексеевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество, подпись, дата)

Руководитель Кузнецова Ксения Валерьевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество, подпись, дата)

Рецензент Кабанов Сергей Павлович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество, подпись, дата)

Нормоконтролёр Фогт Ирина Александровна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество, подпись, дата)

2022

Государственное автономное профессиональное образовательное учреждение Мурманской области

«Мурманский колледж экономики и информационных технологий»

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Начальник учебного отдела  ГАПОУ МО «МКЭиИТ»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  И.Г. Никитина  «\_\_\_\_» \_\_\_\_\_\_ 20\_\_ г. |

**ЗАДАНИЕ НА ДИПЛОМНОЕ ПРОЕКТИРОВАНИЕ**

Студента(ки) Селегененко Юрия Алексеевича

Отделение: дневное

Курс 4 группа П2

Специальность 09.02.07 «Информационные системы и программирование»

Тема Учёт хранения грузов в АО «Кипрей»

Дипломный проект должен содержать:

* разработанный программный продукт;
* пояснительную записку.

Программный продукт должен реализовывать следующие основные функции:

* + ведение таблиц и справочников;
  + формирование выходных документов;
  + поиск информации;
  + авторизация пользователей по паре логин-пароль;
  + протоколирование входа пользователей в систему;
  + просмотр всех записей в табличной форме.

Дата выдачи задания "18" января 2022 года

Дата сдачи работы "\_\_" июня 2022 года

Руководитель проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кузнецова Ксения Валерьевна

(подпись) (расшифровка)

**Содержание**

[Введение 5](#_Toc104822765)

[1 Описание постановки задачи 6](#_Toc104822766)

[1.1 Описание предметной области 6](#_Toc104822767)

[1.1.1 Характеристика организации 6](#_Toc104822768)

[1.1.2 Описание технологии обработки информации о хранимых грузах 7](#_Toc104822769)

[1.2 Характеристика задачи 8](#_Toc104822770)

[1.3 Описание выходной информации 9](#_Toc104822771)

[1.4 Описание входной информации 10](#_Toc104822772)

[2 Организация и ведение информационной базы 11](#_Toc104822773)

[2.1 Описание систем классификации и систем кодирования 11](#_Toc104822774)

[2.2 Описание внутримашинной базы 11](#_Toc104822775)

[3 Описание программного обеспечения 17](#_Toc104822776)

[3.1 Описание ОС 17](#_Toc104822777)

[3.2 Описание СУБД 19](#_Toc104822778)

[3.3 Описание среды программирования 21](#_Toc104822779)

[3.4 Описание языка программирования 23](#_Toc104822780)

[3.5 Описание используемых компонентов 25](#_Toc104822781)

[3.6 Описание проектных методов 27](#_Toc104822782)

[4 Руководство пользователя 32](#_Toc104822783)

[4.1 Подготовка к работе 32](#_Toc104822784)

[4.2 Описание действий пользователя во время работы с программным продуктом 33](#_Toc104822785)

[4.2.1 Инструкция для менеджера 33](#_Toc104822786)

[4.2.2 Инструкция для директора 34](#_Toc104822787)

[4.3 Перечень диагностических сообщений 34](#_Toc104822788)

[4.4 Действия пользователя в аварийных ситуациях 35](#_Toc104822789)

[5 Выводы и предложения 36](#_Toc104822790)

[6 Перечень принятых сокращений 37](#_Toc104822791)

[7 Перечень терминов 38](#_Toc104822792)

[Список использованной литературы 39](#_Toc104822793)

[Приложение А – Диаграмма вариантов использования 41](#_Toc104822794)

[Приложение Б – Образцы выходных документов 43](#_Toc104822795)

[Приложение В – Экранные формы 46](#_Toc104822796)

[Приложение Г – Образцы входных документов 57](#_Toc104822797)

[Приложение Д – Содержимое таблиц базы данных 61](#_Toc104822798)

[Приложение Е – Листинг фрагмента программного кода 65](#_Toc104822799)

[Приложение Ж – USB накопитель 70](#_Toc104822800)

# Введение

В эпоху активно развивающегося мира, всё чаще требуется автоматизировать монотонные и трудоёмкие процессы, хранить, систематизировать и обрабатывать большие объёмы информации. Одним из видов деятельности компании АО «Кипрей» является предоставление услуг по хранению грузов. Этот вид деятельности связан со сложными структурами данных. Поэтому, необходимо наличие системы, которая, в данном случае, позволит эффективно и безошибочно вести учёт хранения грузов. Все склады расположены в пределах имеющейся у организации недвижимости и используются в целях предоставления услуг по хранению различных грузов. Для выполнения этих целей определяется место, где будет храниться груз и заключается договор между клиентом и менеджером на срок от одного дня, плата по которому оговаривается заранее.

Целью дипломного проекта является разработка системы, реализующей автоматизацию учёта хранения грузов в АО «Кипрей».

Задачей дипломного проекта является написание программы на языке C#, позволяющей работать с базой данных.

Существуют программы с похожим назначением, например, 1С Торговля и склад. Но она обладает существенными недостатками: громоздкий интерфейс, устаревший внешний вид, много лишних функций и недостаток нужных. Поэтому возникло решение разработать новый программный продукт.

# 1 Описание постановки задачи

## 1.1 Описание предметной области

### 1.1.1 Характеристика организации

Наименование организации: АО «Кипрей»

Наименование отрасли, в которой работает организация – Операции с недвижимым имуществом.

Основные виды деятельности организации:

* аренда и управление собственным или арендованным недвижимым имуществом;

Дополнительные виды деятельности организации:

* хранение и складирование прочих грузов;
* аренда и лизинг легковых автомобилей и легких автотранспортных средств;
* аренда и лизинг грузовых транспортных средств;
* аренда и лизинг прочих видов транспорта, оборудования и материальных средств, не включенных в другие группировки.

Организационная структура организации представлена на рисунке 1

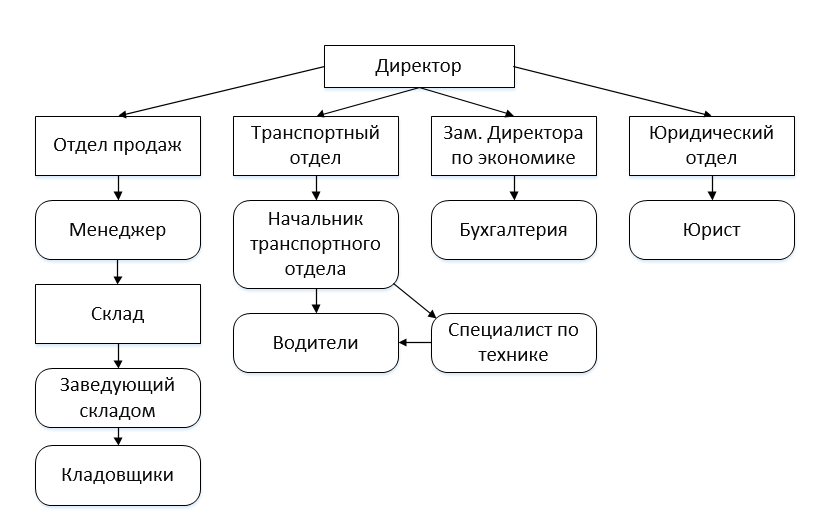


Рисунок 1 – Организационная структура

### 1.1.2 Описание технологии обработки информации выбранного для автоматизации вида деятельности

Модель работы системы показана на рисунке 1 в виде диаграммы IDEF0

Входная информация:

* план склада;
* данные о клиенте;
* список сотрудников;

Управление:

* статья 907 ГК РФ. Договор складского хранения;

Механизм:

* управляющий персонал;

Выход:

* счёт-договор;
* прайс лист.

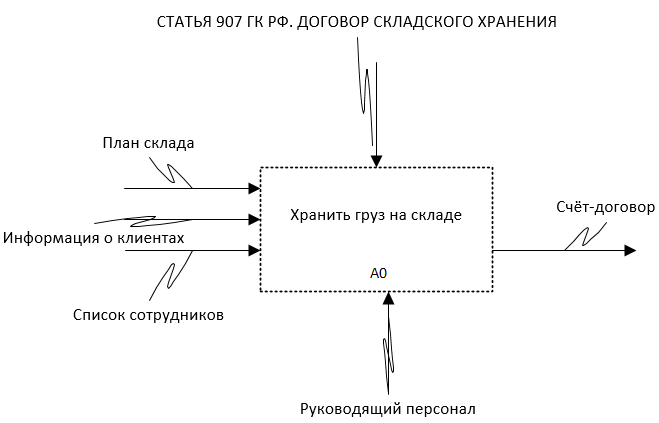


Рисунок 1 ‒ IDEF0 диаграмма

Для формирования счёта необходимо записать данные о клиенте, определить размер ежедневной платы на хранение, заключить договор между менеджером и клиентом. Для расчёта суммы к оплате необходимо вычислить количество дней хранения у каждой позиции в договоре и умножить её на актуальную цену за день хранения того места, на котором хранится груз. Необходимость автоматизации данного процесса объясняется объёмом участвующих в нём данных, также, упущения и ошибки могут повлечь за собой серьёзные убытки для организации.

## 1.2 Характеристика задачи

Задача «Учёт хранения грузов в АО Кипрей» предназначена для учёта хранения грузов на складах.

Перечень функций, реализуемых задачей:

* ведение справочников: «Типы грузов»;
* ввод и редактирование информации о:

1. сотрудниках;
2. договорах;
3. физических клиентах;
4. складах;
5. местах для складирования;
6. юридических клиентах;
7. хранении;

* поиск информации в любой таблице:
* формирование выходных документов:

1. «Счёт-договор»;
2. «Прайс лист»;

* авторизация пользователей по паре логин-пароль;
* протоколирование входа пользователей в систему

Объекты, участвующие в системе:

* директор;
* менеджер;
* кладовщик.

Задача локальна, связи с другими задачами не имеет.

Диаграмма вариантов использования (Use-Case) представлена в приложении А.

## 1.3 Описание выходной информации

Описание выходной информации представлено в таблице 1.

Таблица 1 – Описание выходной информации

|  |  |  |
| --- | --- | --- |
| **Наименование выходной информации** | **Форма представления** | **Перечень получателей** |
| Счёт-договор | Документ форматов Excel-файл, docx-файл, pdf-файл | Клиент |
| Прайс-лист | Документ форматов docx-файл, pdf-файл | Клиент |
|  |  |  |
| Продолжение таблицы 1 | | |
| **Наименование выходной информации** | **Форма представления** | **Перечень получателей** |
| Информация о хранении | Экранная форма | Директор, менеджер, кладовщик |
| Информация о сотрудниках | Экранная форма | Директор |
| Информация о физических клиентах | Экранная форма | Менеджер, директор |
| Информация о местах для складирования | Экранная форма | Директор, менеджер, кладовщик |
| Информация о складах | Экранная форма | Директор, менеджер, кладовщик |
| Информация о юридических клиентах | Экранная форма | Директор, менеджер |
| Информация о договорах | Экранная форма | Директор, менеджер |

Выходные документы представлены в приложении Б.

Образцы экранных форм представлены в приложении В.

## 1.4 Описание входной информации

Описание входной информации представлено в таблице 2.

Таблица 2 – Описание входной информации

|  |  |  |
| --- | --- | --- |
| **Наименование входного документа** | **Перечень входной информации** | **Периодичность поступления входных документов** |
| План склада | Адрес, тип груза | При необходимости |
| Анкета юридического клиента | ОКПО, Адрес, ФИО директора, телефон, название, БИК, ИНН, банк | При необходимости |
| Анкета сотрудника | ФИО, телефон, должность | При необходимости |
| Анкета физического клиента | ФИО, телефон, паспорт | При необходимости |

Входные документы представлены в приложении Г.

# 2 Организация и ведение информационной базы

## 2.1 Описание систем классификации и систем кодирования

Во всех таблицах базы данных используется порядковая система кодирования записей.

Порядковая система кодирования – система кодирования, в которой нумерация объектов производится по порядку.

Пример кодирования записей в справочнике «Типы груза» приведён в таблице 3.

Таблица 3 – Кодирование записей в справочнике «Типы груза»

|  |  |
| --- | --- |
| **Код типа** | **Тип** |
| 1 | Насыпной |
| 2 | Опасный |
| 3 | Хрупкий |

Пример кодирования записей в таблице «Склады» приведён в таблице 4.

Таблица 4 – Кодирование записей в таблице «Склады»

|  |  |  |
| --- | --- | --- |
| **ID склада** | **Код типа** | **Адрес** |
| 1 | 3 | ул. Военного городка тер, дом 128 |
| 2 | 2 | ул. Жидков пер, дом 124 |
| 3 | 1 | ул. Подбельского 7-й проезд, дом 116 |

## 2.2 Описание внутримашинной базы

Перечень наименований таблиц базы данных и их описание представлено в таблице 5.

Таблица 5 – Наименование таблиц и их описание

|  |  |
| --- | --- |
| **Наименование таблицы** | **Описание** |
| clients | Физические клиенты |
| dogovori | Договоры |
| places | Места на складах |
| Продолжение таблицы 5 | |
| **Наименование таблицы** | **Описание** |
| s\_roles | Список ролей |
| storages | Информация о складах |
| storaging | Информация о хранении |
| types | Типы грузов |
| uridclients | Юридические клиенты |
| users | Пользователи |
| pricehistory | История цен |

Описание структур таблиц базы данных представлено в таблицах 6-15.

Таблица 6 – Структура записей таблицы «clients»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Идентификатор поля** | **Наименование поля** | **Тип поля** | **Ключ** | **Обяз. к зап.** |
| idClient | id клиента | Числовой | Первичный | + |
| fio | ФИО | Текст (45) |  | + |
| phone | телефон | Текст (45) |  | + |
| passport | паспорт | Текст (45) |  | + |

Таблица 7 – Структура записей таблицы «dogovori»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Идентификатор поля** | **Наименование поля** | **Тип поля** | **Ключ** | **Обяз. к зап.** |
| idDogovor | Id договора | Числовой | Первичный | + |
| idUser | id пользователя | Текст (45) | Внешний(users) | + |
| idClient | id клиента | Числовой | Внешний(clients) | + |
| dataBegin | дата начала | Дата |  | + |
| dataEnd | дата конца | Дата |  | + |
| entity | юрид/физ | Числовой |  | + |

Таблица 8 – Структура записей таблицы «places»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Идентификатор поля** | **Наименование поля** | **Тип поля** | **Ключ** | **Обяз. к зап.** |
| idPlace | id места | Числовой | Первичный | + |
| idStorage | id склада | Числовой | Внешний(storages) | + |
| placeX | длина места | Числовой |  | + |
| Продолжение таблицы 8 | | | | |
| **Идентификатор поля** | **Наименование поля** | **Тип поля** | **Ключ** | **Обяз. к зап.** |
| placeY | высота места | Числовой |  | + |
| placeZ | ширина места | Числовой |  | + |
| cost | цена/день | Числовой |  | + |

Таблица 9 – Структура записей таблицы «roles»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Идентификатор поля** | **Наименование поля** | **Тип поля** | **Ключ** | **Обяз. к зап.** |
| codeRole | код роли | Числовой | Первичный | + |
| role | роль | Текст (45) |  | + |

Таблица 10 – Структура записей таблицы «storages»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Идентификатор поля** | **Наименование поля** | **Тип поля** | **Ключ** | **Обяз. к зап.** |
| idStorage | id склада | Числовой | Первичный | + |
| codeType | код типа | Числовой | Внешний(types) | + |
| adress | адрес | Текст (45) |  | + |

Таблица 11 – Структура записей таблицы «storaging»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Идентификатор поля** | **Наименование поля** | **Тип поля** | **Ключ** | **Обяз. к зап.** |
| idStoraging | Id хранения | Числовой | Первичный | + |
| idDogovor | id договора | Числовой | Внешний(dogovori) | + |
| idPlace | id места | Числовой | Внешний(places) | + |
| weight | вес | Числовой |  | + |
| storeCost | стоимость | Числовой |  | + |
| cargoX | длина | Числовой |  | + |
| cargoY | высота | Числовой |  | + |
| cargoZ | ширина | Числовой |  | + |
| desc | описание | Текст (90) |  | + |

Таблица 12 – Структура записей таблицы «types»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Идентификатор поля** | **Наименование поля** | **Тип поля** | **Ключ** | **Обяз. к зап.** |
| codeType | код типа | Числовой | Первичный | + |
| type | тип | Текст (45) |  | + |

Таблица 13 – Структура записей таблицы «uridclients»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Идентификатор поля** | **Наименование поля** | **Тип поля** | **Ключ** | **Обяз. к зап.** |
| okpo | окпо | Текст (45) | Первичный | + |
| urArdes | юрид. адрес | Текст (90) |  | + |
| directorFio | ФИО директора | Текст (45) |  | + |
| firmPhone | телефон | Текст (45) |  | + |
| bik | бик | Текст (45) |  | + |
| inn | инн | Текст (45) |  | + |
| bank | банк | Текст (45) |  |  |

Таблица 14 – Структура записей таблицы «users»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Идентификатор поля** | **Наименование поля** | **Тип поля** | **Ключ** | **Обяз. к зап.** |
| idUser | id пользователя | Числовой | Первичный | + |
| codeRole | код роли | Числовой | Внешний(roles) | + |
| login | логин | Текст (45) |  | + |
| password | пароль | Текст (45) |  | + |
| fio | ФИО | Текст (45) |  | + |
| phone | телефон | Текст (45) |  | + |
| isDeleted | удалён | Числовой |  | + |

Таблица 15 – Структура записей таблицы «companies»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Идентификатор поля** | **Наименование поля** | **Тип поля** | **Ключ** | **Обяз. к зап.** |
| IdCompany | Id компании | Числовой | Первичный | + |
| CompanyName | Название | Текст (60) |  | + |
| CompEmail | Почта компании | Текст (60) |  | + |
| CompPhone | Телефон компании | Текст (20) |  | + |
| Продолжение таблицы 15 | | | | |
| **Идентификатор поля** | **Наименование поля** | **Тип поля** | **Ключ** | **Обяз. к зап.** |
| CompAdress | Адрес компании | Текст (60) |  | + |

Инфологическая модель базы данных представлена на рисунке 2

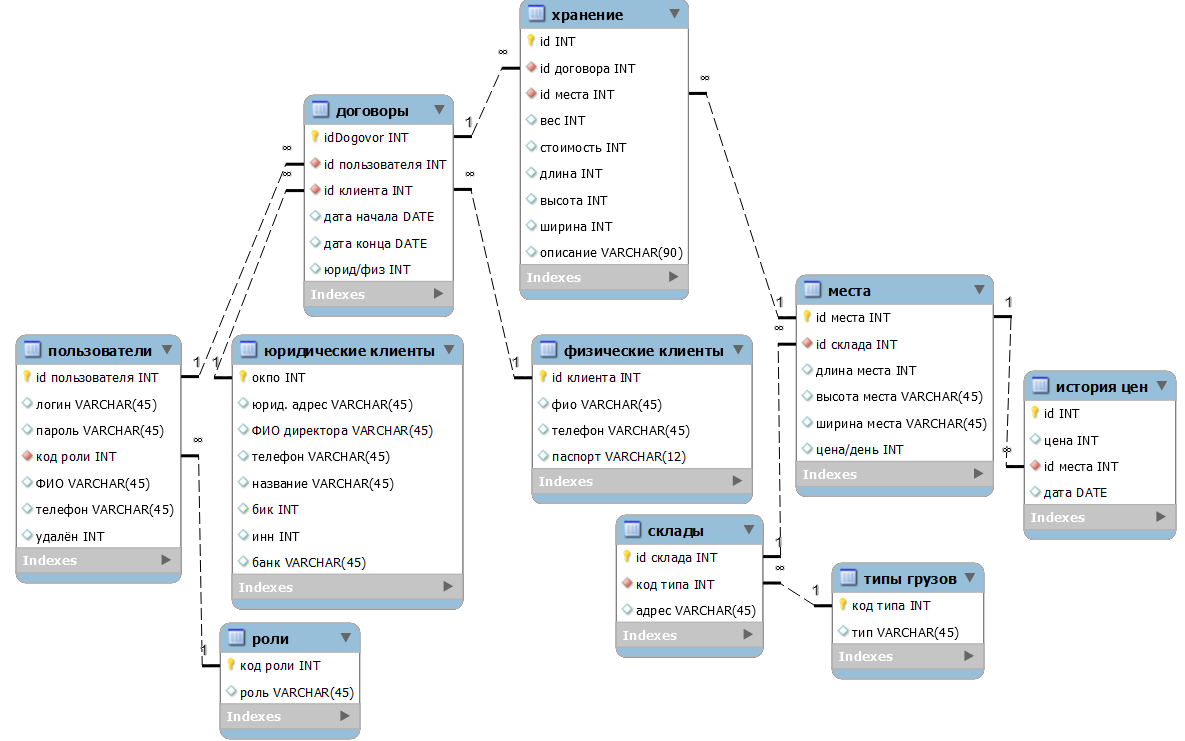


Рисунок 2 ‒ ER диаграмма

Для корректной работы программы необходимо заполнять таблицы БД в приведенной ниже последовательности.

Директор заполняет: пользователи, типы грузов, склады, места.

Затем менеджер заходит в свою учётную запись и заполняет следующие таблицы: юридические клиенты, физические клиенты, договоры, хранение.

Содержимое таблиц базы данных представлено в приложении Д.

В разработанном приложении применяются следующие средства защиты БД от ошибок, несанкционированного доступа, и обеспечения   
целостности БД:

* авторизация пользователей системы по паре имя пользователя-пароль;
* шифрование паролей с помощью алгоритма MD5;
* проверка при удалении на существование записи в других таблицах.

# 3 Описание программного обеспечения

## 3.1 Описание ОС

Для разработки программного продукта, специализирующегося на учёте хранения грузов в АО «Кипрей» была выбрана операционная система windows 10.

Windows 10 — операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT  [[1]](https://ru.wikipedia.org/wiki/Windows_10).

Основная функциональность Windows 10 заключается в предоставлении графической операционной системы с сенсорным управлением, которая использует графические интерфейсы и функции клавиатуры, чтобы пользователь мог эффективно общаться с компьютером. Некоторые из основных функций включают в себя голосовой помощник «Cortana», который помогает пользователю ориентироваться в операционной системе, и режим планшета, который помогает гибридным устройствам, таким как ПК 2-в-1, вести себя как планшет, когда он автономен и как ПК при подключении к клавиатуре.

Ошибки пользователя предотвращаются в Windows 10 с помощью обновленной функции поиска. Эта функция предоставляет рекомендации для возможных результатов поиска, когда пользователь вводит буквы в строку поиска. Это предотвращает открытие пользователем нежелательных файлов, приложений и т. Д. И, следовательно, сокращает время, необходимое для выполнения желаемой задачи. Кроме того, во многих случаях, если пользователь пытается изменить системные настройки или любые конфиденциальные данные, операционная система предупреждает, что для этой задачи требуется административный доступ, который защищает пользователя от потенциально разрушительных действий. Пользователь не полностью защищен от ошибок пользователя, поскольку Windows не может полностью предсказать намерения пользователя, но Windows делает все возможное, чтобы предотвратить ошибки пользователя.

В Windows есть много горячих клавиш, которые позволяют легко выполнять повторяющиеся функции, такие как переключение между задачами с помощью клавиши alt + tab, что ускоряет рабочий процесс. В Windows 10 такие функции, как копирование и вставка, включены в командной строке, добавляя непрерывность и согласованный рабочий процесс через ОС. Когда пользователь впервые загружает окна, ему предлагается войти в свою учетную запись Microsoft, в которой хранятся системные настройки и пользовательские данные, что делает его более удобным в долгосрочной перспективе. В Windows 10 также имеется игровой режим аппаратного ускорения, который повышает игровую производительность за счет сокращения фоновых процессов, которые вместо этого можно использовать для ускорения игры. Эти оптимизации, предоставляемые Windows, отлично справляются с оптимизацией пользовательских операций и помогают пользователю легко и быстро работать с несколькими задачами одновременно [[2]](https://essay-pro.ru/sochinenie/osnovnye-funkczionalnye-vozmozhnosti-windows-10/).

Плюсы:

* полноэкранные приложения можно уменьшить до нужных размеров, чего нельзя было сделать в восьмой версии;
* поддержка нескольких рабочих столов без зависаний и торможений, характерных для предыдущей системы линейки;
* системные требования для установки Windows 10 такие же, как для седьмой версии. Она без проблем запустится на устаревших системах с ограниченным объемом оперативной памяти и слабым процессором. Официальная спецификация очень демократична;
* обновления устанавливаются автоматически в фоновом режиме и не требуют перезагрузки компьютера.

Минусы:

* проблемы с региональными настройками. Многие пользователи отметили, что старались внести настройки своего региона, но ОС автоматически устанавливает США, с чем ничего нельзя поделать;
* проблемы с приватностью. Лицензионное соглашение W10 даёт Microsoft право на сбор вашей персональной информации и статистики – данных о местоположении, истории браузера и прочего в таком духе. Как заверяют Microsoft, информация используется для подбора рекламных материалов и усовершенствования программных продуктов компании [[3]](https://myfirstcomp.ru/operation_systems/windows-10-minusy-i-plyusy-operacionnoj-sistemy/).

## 3.2 Описание СУБД

Для разработки программного продукта, специализирующегося на учёте хранения грузов в АО «Кипрей» была выбрана СУБД MySql 8.0.

MySQL 8.0 — свободная реляционная система управления базами данных [[4]](https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html). Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц [[5]](https://ru.wikipedia.org/wiki/MySQL).

MySQL Workbench — инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL [[6]](https://ru.wikipedia.org/wiki/MySQL_Workbench).

MySQL Workbench является инструментом, который позволяет создавать ER диаграммы для баз данных MySQL. Можно использовать его для разработки нового формата базы данных, уже существующего документа, или для выполнения комплексного перехода. Приложение обеспечивает визуальное представление таблиц базы данных, видов, хранимых процессов и внешних ключей. Кроме того, оно может синхронизировать модель разработки с фактической базой данных.  
MySQL Workbench может также создавать необходимые скрипты для создания базы данных, созданные в эскизах. Она совместима с моделями баз данных DBDesigner 4 и поддерживает новые функции по MySQL 5 [[7]](https://mysql-workbench.ru.uptodown.com/windows).

MySQL обладает большим количеством преимуществ перед другими системами:

* СУБД MySQL является одной из самых быстрых баз данных среди имеющихся на современном рынке;
* СУБД MySQL является высокопроизводительной и относительно простой в использовании СУБД, которую значительно проще инсталлировать и администрировать, чем многие другие большие системы;
* СУБД MySQL распространяется бесплатно для домашнего использования.
* MySQL понимает команды SQL, поддерживает интерфейс ODBC, протокол интерфейса с базами данных, разработанный компанией Microsoft;
* Сервер позволяет подключаться одновременно неограниченному количеству пользователей. Доступ к серверу можно осуществить в интерактивном режиме с помощью различных интерфейсов, позволяющих вводить запросы и просматривать полученные результаты: это программы-клиенты, работающие с командной строкой, Web-броузеры, программы-клиенты, работающие в системе Windows, это, наконец, программные интерфейсы для языков C, Perl, Java, PHP и Python. Так что можно использовать как готовое клиентское программное обеспечение, так и создавать свое собственное;
* MySQL предназначена для работы в сети, может быть доступна через Интернет, но при этом снабжена развитой системой защиты от несанкционированного доступа;
* MySQL отлично работает как под управлением разных версий UNIX, так и под управлением других систем: Windows и OS/2. При этом система работает как на мощных серверах, так и на домашних ПК;
* дистрибутив СУБД MySQL можно получить, воспользовавшись Web-броузером. Также, можно получить и исходный код и внести в него коррективы [[8]](https://lectmania.ru/1x5c7e.html).

## 3.3 Описание среды программирования

Для разработки программного продукта, специализирующегося на учёте хранения грузов в АО «Кипрей» была выбрана среда разработки Microsoft Visual Studio 2019.

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, UWP а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, .NET Core, .NET, MAUI, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server)  [[9]](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio).

Visual Studio IntelliCode повышает эффективность разработки программного обеспечения с помощью искусственного интеллекта. Для создания рекомендаций IntelliCode анализирует 2000 проектов с открытым кодом на GitHub.

Новый индикатор работоспособности документа дополнен новой командой очистки кода. Эту новую команду можно использовать для определения и устранения предупреждений и предложений одним действием или одним нажатием кнопки [[10]](https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2019?view=vs-2022).

Ниже перечислены основные преимущества IDE Visual Studio:

* встроенный Web-сервер. Для обслуживания Web-приложения ASP.NET необходим Web-сервер, который будет ожидать Web-запросы и обрабатывать соответствующие страницы. Наличие в Visual Studio интегрированного Web-сервера позволяет запускать Web-сайт прямо из среды проектирования, а также повышает безопасность, исключая вероятность получения доступа к тестовому Web-сайту с внешнего компьютера, поскольку тестовый сервер может принимать соединения только с локального компьютера;
* поддержка множества языков при разработке. Visual Studio позволяет писать код на своем языке или любых других предпочитаемых языках, используя все время один и тот же интерфейс. Более того, Visual Studio позволяет создавать Web-страницы на разных языках, но помещать их все в одно и то же Web-приложение. Единственным ограничением является то, что в каждой Web-странице можно использовать только один язык;
* меньше кода для написания. Для создания большинства приложений требуется приличное количество стандартного стереотипного кода, и Web-страницы ASP. NET тому не исключение. Например, добавление Web-элемента управления, присоединение обработчиков событий и корректировка форматирования требует установки в разметке страницы ряда деталей. В Visual Studio такие детали устанавливаются автоматически.
* интуитивный стиль кодирования. По умолчанию Visual Studio форматирует код по мере его ввода, автоматически вставляя необходимые отступы и применяя цветовое кодирование для выделения элементов типа комментариев. Такие незначительные отличия делают код более удобным для чтения и менее подверженным ошибкам. Применяемые Visual Studio автоматически параметры форматирования можно даже настраивать, что очень удобно в случаях, когда разработчик предпочитает другой стиль размещения скобок (например, стиль K&R, при котором открывающая скобка размещается на той же строке, что и объявление, которому она предшествует);
* более высокая скорость разработки. Многие из функциональных возможностей Visual Studio направлены на то, чтобы помогать разработчику делать свою работу как можно быстрее. Удобные функции, вроде функции IntelliSense (которая умеет перехватывать ошибки и предлагать правильные варианты), функции поиска и замены (которая позволяет отыскивать ключевые слова как в одном файле, так и во всем проекте) и функции автоматического добавления и удаления комментариев (которая может временно скрывать блоки кода), позволяют разработчику работать быстро и эффективно [[11]](https://studbooks.net/2258619/informatika/opisanie_sredy_razrabotki_microsoft_visual_studio).

## 3.4 Описание языка программирования

Во время разработки программного продукта, специализирующегося на учёте хранения грузов в АО «Кипрей» использовался язык программирования C#.

C# 9 — объектно-ориентированный язык программирования. относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML [[12]](https://ru.wikipedia.org/wiki/C_Sharp).

Предназначен для разработки самых разнообразных приложений, предназначенных для выполнения в среде .NET Framework. Язык C# прост, безопасен и объектно-ориентирован. Благодаря множеству нововведений C# обеспечивает возможность быстрой разработки приложений, но при этом сохраняет выразительность и элегантность, присущую языкам C [[13]](https://flexberry.github.io/ru/gbt_csharp.html#:~:text=C%23%20(%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%BD%D0%BE%D1%81%D0%B8%D1%82%D1%81%D1%8F%20%E2%80%9C%D0%A1%D0%B8%2D%D1%88%D0%B0%D1%80%D0%BF,%2C%20%D1%82%D0%B8%D0%BF%D0%BE%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%B5%D0%BD%20%D0%B8%20%D0%BE%D0%B1%D1%8A%).

На сегодняшний момент язык программирования C# один из самых мощных, быстро развивающихся и востребованных языков в ИТ-отрасли. В настоящий момент на нем пишутся самые различные приложения: от небольших десктопных программ до крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей [[14]](https://metanit.com/sharp/tutorial/1.1.php).

С# популярен за счет своей «простоты». Простоты для современных программистов и больших команд разработчиков, чтобы те могли в сжатые сроки создавать функциональные и производительные приложения. Этому способствуют нетипичные конструкции языка и специфичный синтаксис, помогающий максимально органично реализовать намеченные функции.

Язык C# практически универсален. Можно использовать его для создания любого ПО: продвинутых бизнес-приложений, видеоигр, функциональных веб-приложений, приложений для Windows, ПО для защиты систем, macOS, мобильных программ для iOS и Android [[15]](https://timeweb.com/ru/community/articles/chto-takoe-csharp).

У C# выделяют много преимуществ:

* поддержка подавляющего большинства продуктов Microsoft;
* бесплатность ряда инструментов для небольших компаний и некоторых индивидуальных разработчиков — Visual Studio, облако Azure, Windows Server, Parallels Desktop для Mac Pro и др.;
* типы данных имеют фиксированный размер (32-битный int и 64-битный long), что повышает «мобильность» языка и упрощает программирование, так как вы всегда знаете точно, с чем вы имеете дело;
* автоматическая «сборка мусора». Это значит, что в большинстве случаев не придётся заботиться об освобождении памяти. общеязыковая среда CLR сама вызовет сборщик мусора и очистит память;
* большое количество специальных конструкций, разработанных для понимания и написания кода. Они не имеют значения при компиляции;
* низкий порог вхождения. Синтаксис C# имеет много схожего с другими языками программирования, благодаря чему облегчается переход для программистов. Язык C# часто признают наиболее понятным и подходящим для новичков;
* с помощью Xamarin на C# можно писать программы и приложения для таких операционных систем, как iOS, Android, MacOS и Linux;
* сегодня в любом регионе России имеется немало вакантных мест на должность C#-программиста.

Также у языка C# есть некоторые недостатки:

* приоритетная ориентированность на платформу Windows;
* язык бесплатен только для небольших фирм, индивидуальных программистов, стартапов и учащихся. Крупной компании покупка лицензионной версии этого языка обойдётся в круглую сумму [[16]](https://gb.ru/posts/yazyk-programmirovaniya-c-sharp-istoriya-specifika-mesto-na-rynke).

## 3.5 Описание используемых компонентов

Описание используемых компонентов в программном продукте представлены в таблице 16

Таблица 16 - Описание используемых компонентов

|  |  |
| --- | --- |
| **Компонент** | **Назначение** |
| DataGridView | Элемент управления DataGridView предоставляет настраиваемую таблицу для отображения данных. Класс DataGridView позволяет настраивать ячейки, строки, столбцы и границы. |
| TextBox | Отображает текст, введенный на этапе разработки, который пользователи могут |
|  | изменять во время выполнения или с помощью программных средств. |
| MaskedTextBox | Ограничивает формат вводимых пользователем данных |
| Label | Отображает текст, недоступный для непосредственного изменения пользователем. |
| ProgressBar | Отображает для пользователя текущий ход выполнения операции. |
| CheckBox | Отображает флажок и метку для текста. Обычно используется для задания параметров. |
| RadioButton | Отображает кнопку, которую можно включить или отключить. |
| DateTimePicker | Отображает графический календарь, позволяющий пользователю выбрать дату или время. |
| Продолжение таблицы 16 |  |
| **Компонент** | **Назначение** |
| SaveFileDialog | Отображает диалоговое окно, позволяющее пользователям сохранять файл. |
| MenuStrip | Создает настраиваемые меню |
| Button | Запускает, останавливает или прерывает процесс. |
| Panel | Группируют набор элементов управления на кадре без меток с возможностью прокрутки. |

## 3.6 Описание проектных методов

Описание проектных методов, использованных при создании программного продукта, представлено в таблице 17

Таблица 17 – Описание проектных методов

|  |  |  |
| --- | --- | --- |
| **Имя класса** | **Заголовок метода** | **Назначение метода** |
| DBUsers | DataTable Combo () | Получение таблицы Users для comboBox |
| DataTable Get(string and) | Получение таблицы Users |
| Void Add(string codeRole, string fio, string phone, string login) | Добавление записи в таблицу users |
| Void Delete() | Удаление записи из таблицы users |
| Void PassReset(string name) | Сброс пароля пользователя из таблицы users |
| Void Edit(string codeRole, string fio, string phone, string login) | Редактирование записи в таблице users |
| Void EditPassw(string password) | Редактирование пароля пользователя |
| Продолжение таблицы 17 | | |
| **Имя класса** | **Заголовок метода** | **Назначение метода** |
|  | Bool AlreadyExists(string val, string field) | Проверяет выбранное поле на наличие выбранного значения |
| Bool HasDogovor(string id) | Проверяет наличие активных договоров у пользователя |
| Void Restore() | Возврат пользователя из архива |
| DBUrClients | DataTable Combo() | Получение uridclients для comboBox |
| DataTable Get(string where) | Получение таблицы uridclients |
| Void Add(string okpo,string urArdes,string directorFio,string firmPhone,string firmName) | Добавление записи в таблицу uridclients |
| Void Delete () | Удаление записи из таблицы uridclients |
| Void Edit(string urArdes,string directorFio,string firmPhone, string firmName,string bik,string inn,string bank) | Редактирование записи в таблице uridclients |
| InDogovor(string id) | Проверяет, существует ли связанная запись |
| bool AlreadyExists(string val, string field) | Проверяет поле на наличие значения |
| DBTypes | DataTable Combo() | Получение таблицы types для comboBox |
| DataTable Get(string where) | Получение таблицы types |
| void Add (string type) | Добавление записи в таблицу types |
| void Delete() | Удаление записи из таблицы types |
| void Edit (string type) | Редактирование записи в таблицу types |
| bool AlreadyExists(string val, string field) | Проверяет поле на наличие значения |
| bool HasStorage(string id) | Проверяет, существует ли связанная запись |
| Продолжение таблицы 17 | | |
| **Имя класса** | **Заголовок метода** | **Назначение метода** |
| DBStoraging | DataTable Get(string and) | Получение таблицы storaging |
| Add(string idDogovor, string idPlace, string cargoX, string cargoY, string cargoZ, string weight, string desc, string perBegin,string perEnd) | Добавление записи в таблицу Storaging |
| void Delete() | Удаление записи из таблицы Storaging |
| void Edit (string idDogovor, string idPlace,string cargoX, string cargoY, string cargoZ, string weight, string desc, string perBegin, string perEnd) | Редактирование записи в таблице Storaging |
| bool Fits (string cargoX, string cargoY, string cargoZ, string idPlace) | Определяет, вместится ли груз в место на складе |
| DBStorages | DataTable Get(string and) | Получение таблицы Users |
| void Add(string codeType, string adress) | Добавление записи в таблицу Storages |
| void Delete() | Удаление записи из таблицы Storages |
| void Edit(string codeType, string adress) | Редактирование записи в таблице storages |
| bool AlreadyExists(string val, string field) | Проверяет поле на наличие значения |
| DBRoles | DataTable Get() | Получение таблицы roles |
| DBPlaces | DataTable Combo (string and, string perBegin, string perEnd) | Получение таблицы places для combobox |
| DataTable Get(string and) | Получение таблицы places |
| void Add(string idStorage, string placeX, string placeY, string placeZ, string cost) | Добавление записи в таблицу places |
| Продолжение таблицы 17 | | |
| **Имя класса** | **Заголовок метода** | **Назначение метода** |
|  | bool Delete() | Удаление записи из таблицы places |
| void Edit(string idStorage, string placeX, string placeY, string placeZ, string cost) | Редактирование записи в таблице places |
| bool IsNowUsed (string id) | Проверяет, занято ли сегодня место |
| DBDogovori | DataTable Combo () | Получение списка для вывода в comboBox |
| DataTable Get(string and, bool entity) | Получение таблицы dogovori |
| DataTable Get() | Получение "сырой" таблицы dogovori |
| void Add(string idUser, string idClient, string dataBegin, string dataEnd, int entity) | Добавление записи в таблицу dogovori |
| void Delete() | Удаление записи из таблицы dogovori |
| void Edit(string idUser, string idClient, string dataBegin, string dataEnd, int entity) | Редактирование записи в таблице dogovori |
| bool HasStoraging(string id) | Проверяет, существует ли связанная запись |
| DBClients | DataTable Combo(string where) | Получение таблицы clients для comboBox |
| DataTable Get(string where) | Получение таблицы clients |
| void Add(string fio, string phone, string passport) | Добавление записи в таблицу clients |
| void Delete() | Удаление записи из таблицы clients |
| Продолжение таблицы 17 | | |
| **Имя класса** | **Заголовок метода** | **Назначение метода** |
|  | void Edit(string fio, string phone, string passport) | Редактирование записи в таблице clients |
| bool InDogovor(string id) | Проверяет, существует ли связанная запись |
| bool AlreadyExists(string val, string field) | Проверяет значение выбранного поля на наличие выбранного значения |
| DataBase | bool Connect() | Соединение с БД |
| bool Authorization(string Login, string Password) | Авторизация |
| bool Export() | Экспорт БД |
| bool Import() | Импорт БД |
| bool Import(string table) | Сохранение таблицы в буфер |
| int CoordinatesOf<T>(this T [,] matrix, T value) | Поиск элемента в двумерной матрице |
| bool Export(string table) | Выгрузка таблицы из буфера |

Листинг фрагмента программного кода представлен в приложении Е.

# 4 Руководство пользователя

## 4.1 Подготовка к работе

Состав устройств ПК:

* системный блок;
* блок питания;
* материнская плата;
* процессор с тактовой частотой не ниже 1,8 ГГц.
* ОЗУ: 2 ГБ;
* жёсткий диск, место на жестком диске минимум 800 МБ;
* видеоадаптер: минимальное разрешение 720p (1280 на 720 пикселей) или более высокое;
* монитор;
* клавиатура;
* мышь.

Состав и содержание дистрибутивного носителя программ и данных:

* db.sql – архивная копия демонстрационной БД;
* папка sklad.

Для работоспособности приложения необходимо предварительно импортировать базу данных из файла «db.sql» на сервер. Инструкция по импорту находится по URL адресу <https://docs.microsoft.com/ru-ru/azure/mysql/single-server/concepts-migrate-import-export>. Дистрибутив приложения поставляется со съемного носителя. Для запуска приложения необходимо дважды кликнуть по ярлыку «АО Кипрей склад», далее можно приступить к работе с приложением.

USB накопитель вложен в приложение Ж.

## 4.2 Описание действий пользователя во время работы с программным продуктом

# 4.2.1 Инструкция для менеджера

Для начала необходимо авторизоваться под своей учётной записью. При входе в систему в первый раз, пароль – 1.

Для оформления договора, в главном меню, изображённом на рисунке В.2 открыть окно клиентов, выбрав пункт в меню слева "Работа с клиентами" => "Физические клиенты". В окне, изображённом на рисунке В.5 Нажать кнопку "Добавить". В окне, изображённом на рисунке В.16 Заполнить поля, затем нажать кнопку «Сохранить». Открыть меню договоров, выбрав пункт в меню слева "Работа с клиентами" => "Договоры". В окне, изображённом на рисунке В.6 нажать кнопку «Добавить». В окне, изображённом на рисунке В.17 в поле "Найти" ввести ФИО добавленного только что клиента, заполнить поля, затем нажать кнопку "Сохранить". В списке появился новый договор. Чтобы узнать подробности о добавленном договоре, нужно найти в таблице строку с информацией о нём и дважды щёлкнуть по нему. В окне, изображённом на рисунке В.4 нажать кнопку «Добавить». Далее, в окне, изображённом на рисунке В.15, следует выбрать номер места в соответствии с нужным типом груза и адресом, заполнить поля, затем нажать кнопку "Сохранить". После этого нужно вернуться в меню договоров, нажав кнопку "Назад" в правом нижнем углу. Выбрать только что созданный договор и нажать кнопку "Сформировать счёт". В окне, изображённом на рисунке В.12, нажать на кнопку с названием желаемого формата документа. Получившийся документ можно будет найти в проводнике компьютера, в разделе "Документы.

Аналогичные действия можно выполнять при заключении договоров с юридическими лицами.

# 4.2.2 Инструкция для директора

Для начала необходимо авторизоваться под своей учётной записью. При входе в систему в первый раз, пароль – 1.

Для добавления нового пользователя необходимо нажать на пункт "Пользователи" в меню слева. В окне, изображённом на рисунке В.3, нажать кнопку "Добавить". В открывшемся окне, представленном на рисунке В.14, заполнить все поля и нажать кнопку "Сохранить".

Для добавления места хранения нужно нажать на пункт "Типы груза" в меню слева. В окне, изображённом на рисунке В.10, нажать кнопку "Добавить". В открывшемся окне, представленном на рисунке В.21, заполнить все поля и нажмите кнопку "Сохранить".

Рекомендуется периодически сохранять резервную копию базы данных на случай потери данных или возникновения неисправности. Это можно сделать в пункте меню «Сервис».

## 4.3 Перечень диагностических сообщений

Таблица 18 - Перечень диагностических сообщений

|  |  |
| --- | --- |
| **Сообщение** | **Действия** |
| Неверный пароль или логин | Ввести верные данные |
| Ошибка при авторизации... | Связаться с разработчиком по электронной почте |
| Выбран неверный файл | Выбрать подходящую резервную копию БД |
| Ошибка во время импортирования | Связаться с разработчиком по электронной почте |
| Ошибка во время экспортирования | Связаться с разработчиком по электронной почте |
| Ошибка при подключении к БД | Запустить сервер MySql |

|  |
| --- |
| Продолжение таблицы 18 |

|  |  |
| --- | --- |
| **Сообщение** | **Действия** |
| Поля должны быть заполнены | Заполните поля |
| Не должны повторяться:(далее следует список полей) | Введите значения, которых ещё нет в программе |
| Введены некорректные значения | Введите корректные значения |
| Срок пересекается со сроком другого хранения в выбранном месте! | Выберите другое место либо измените срок хранения |
| Груз не подходит по размерам | Измените размеры груза или выберите другое место |
| Данный тип был добавлен ранее | Введите значение, которых ещё нет в программе |
| Произошла ошибка во время добавления | Связаться с разработчиком по электронной почте |
| Невозможно выполнить пока с данным клиентом заключен договор | Удалить договор(ы) с этим клиентом и повторить попытку |

Электронная почта разработчика: yura.selegenenko@mail.ru

## 4.4 Действия пользователя в аварийных ситуациях

Таблица 19 – Описание аварийных ситуаций

|  |  |
| --- | --- |
| **Аварийная ситуация** | **Действия пользователя** |
| Программный продукт не отвечает на действия пользователя | Открыть диспетчер задач и остановить процесс программы |
| При формировании выходного документа открывается пустое окно word и появляется сообщение об ошибке | Открыть диспетчер задач и остановить все процессы word. перейти к месту хранения .exe файла. Удалить все имеющиеся в той папке файлы .doc и .docx. Скопировать из папки «d» резервные копии образцов и вставить их на место удалённых. |

# 5 Выводы и предложения

В результате проделанной работы была создана АИС, позволяющая эффективно вести учёт хранения грузов в АО «Кипрей».

Были реализованы следующие функции:

* ведение справочников: «Типы грузов»
* ввод и редактирование информации о:

1. сотрудниках;
2. договорах;
3. физических клиентах;
4. складах;
5. местах для складирования;
6. юридических клиентах;
7. хранении;

* поиск информации в любой таблице:
* формирование выходных документов:

1. «Счёт-договор»;
2. «Прайс лист»;

* авторизация пользователей по паре логин-пароль;
* протоколирование входа пользователей в систему.

В перспективе он может быть дополнен новыми функциями, такими как: ввод капчи, хранение фотографий сотрудников, новые выходные документы.

# 6 Перечень принятых сокращений

Аббревиатуры:

* АО – акционерное общество;
* БД – база данных;
* ИТ – информационные технологии;
* ОС – операционная система;
* ПО – программное обеспечение;
* ФИО – фамилия имя отчество.

Принятые сокращения:

* т.д. – так далее;
* т.е. – то есть;
* т.к. – так как.

# 7 Перечень терминов

Акционерное общество — хозяйственное общество (корпорация), уставный капитал которого разделён на определённое число акций. Участники акционерного общества не отвечают по его обязательствам и несут риск убытков, связанных с деятельностью общества, в пределах стоимости принадлежащих им акций (ограниченная ответственность).

# Список использованной литературы

1 Алан Бьюли. Изучаем SQL. – Пер. с англ. – СПб: ООО «Символ-Плюс», 2017. – 312 с.

2 Албахари, Джозеф, Албахари, Бен. А45 C# 7.0. Карманный справочник. Пер. с англ. – СПб. ООО «Альфа-книга», 2017. — 224 с.: ил. — Парал. тит. англ.

3 Алексей Паутов, MySQL. Руководство администратора. ВИЛЬЯМС, 2005. – 622 с.

4 Брайан Джонсон. Основы Microsoft Visual Studio .NET 2003. – Microsoft, Пер. с англ. – СПб «Кэпитал групп». 2021. – 489 с.

5 Бэрон Шварц. MySQL по максимуму. – «O’Reilly» (издание третье). Перевод с англ. ООО Издательство «Питер», 2018. – 866 с.

6 Владислав Марченко. Сочинение на тему Основные функциональные возможности Windows 10. URL: <https://essay-pro.ru/sochinenie/osnovnye-funkczionalnye-vozmozhnosti-windows-10/>

7 Герасимов Пётр. «Библиофонд» – электронная библиотека для студента. Информационное обеспечение и программирование. База данных MySQL. URL: https://www.bibliofond.ru/view.aspx?id=533433 (дата обращения 24.05.2022).

8 Джо Майо. Самоучитель Microsoft Visual Studio. – Пер. с англ. – СПб. ООО «БХВ-Петербург». 2019. – 464 с.

9 Джон Скит. C# для профессионалов: тонкости программирования, 3-е издание, новый перевод = C# in Depth, 3rd ed.. — М.: «Вильямс», 2014. — 608 с.

10 Джон Шарп. Microsoft Visual C#. – Microsoft. – Пер. с англ. – Москва. ООО Издательство «Питер»), 2017. – 845 с.

11 Егор Лубенев. Windows 10. Минусы и плюсы операционной системы. URL: https://myfirstcomp.ru/operation\_systems/windows-10-minusy-i-plyusy-operacionnoj-sistemy/

12 Зайцев Пётр, Ткаченко Вадим, и Шварц Бэрон, MySQL по максимуму. 3-е издание. ПИТЕР. – 866 с.

13 Кристиан Нейгел и др. C# 5.0 и платформа .NET 4.5 для профессионалов. Professional C# 5.0 and .NET 4.5. — М.: «Диалектика», 2013. — 1440 с.

14 Мартишин Сергей Анатольевич, Симонов Владимир Львович. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench. Учебное пособие. Инфра-М, Форум, 2012. – 459 с.

15 Ник Рендольф, Дэвид Гарднер, Майкл Минутилло, Крис Андерсон. Visual Studio 2010 для профессионалов. Диалектика, 2011 – 1177 с.

16 Прайс Марк. C# 7 и .NET Core. Кроссплатформенная разработка для профессионалов. Издание третье. – Пер. с англ. – СПб. ООО Издательство «Питер», 2016. – 640 с.

17 Редакторы сайта «Metanit». Язык C# и платформа .NET. URL: https://metanit.com/sharp/tutorial/1.1.php

18 Скит Джон. C# для профессионалов. Тонкости программирования. – Пер. с англ. – Казань. ООО Издательство «Вильямс», 2019. – 608 с.

19 Сотрудники корпорации «Microsoft». Новые возможности Visual Studio 2019. URL: https://docs.microsoft.com/ru-ru/visualstudio/ide/whats-new-visual-studio-2019?view=vs-2022.

20 Сотрудники корпорации «Microsoft». помощь и обучение по Windows. URL: https://support.microsoft.com/ru-ru/windows

21 Эдвард Стиллмен, Дж. Грин. Изучаем C#. 2-е издание Head First C#, 2ed. — СПб.: «Питер», 2012. — 704 с.

# Приложение А

**Диаграмма вариантов использования**

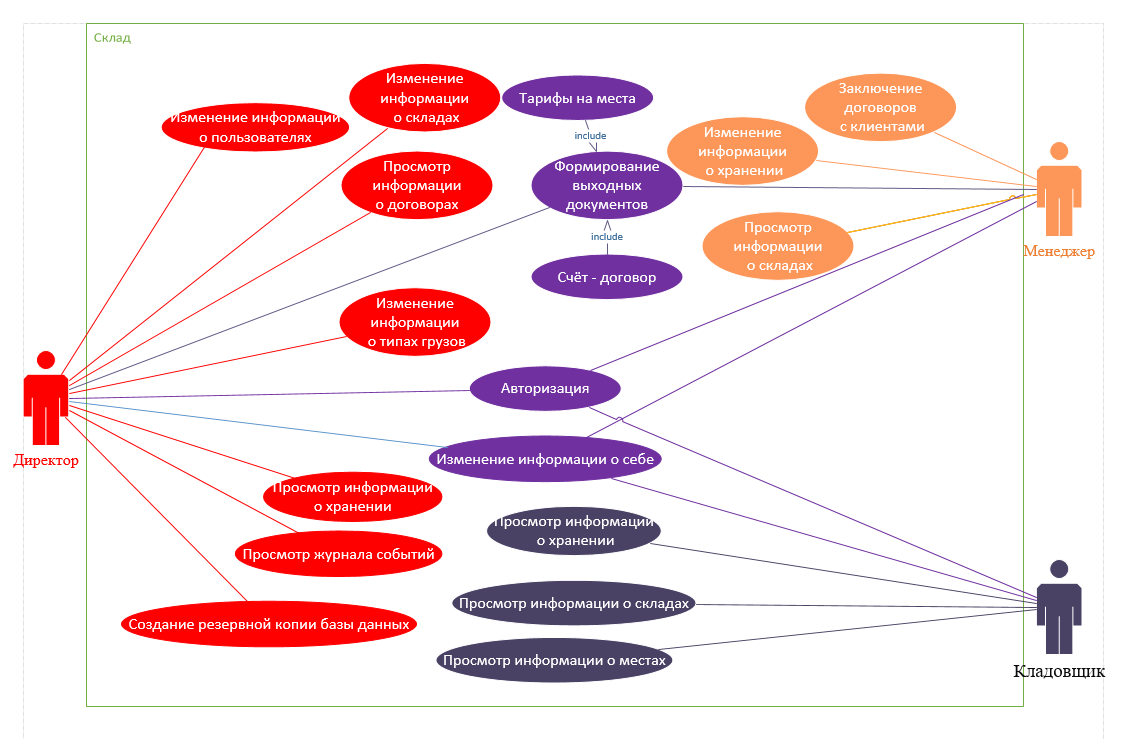
****

Рисунок А.1 ‒ Диаграмма вариантов использования

# Приложение Б

**Образцы выходных документов**

|  |  |  |
| --- | --- | --- |
| Сбербанк | БИК | 624797001 |
| Сч. № | 89234572365629356232345972 |
| Банк получателя |
| ИНН 5193100992 | Сч. № | 19347568913457616713815060 |
| АО “Кипрей” |
|
| Получатель |

|  |
| --- |
| **Счет № 191696 от 21 апреля 2022 г.** |

|  |  |
| --- | --- |
| Поставщик: | **АО “Кипрей”** **Мурманская обл., г. Мурманск, проспект Кольский, д. 158 к. 1 тел.** **+7 (965) 520-86-77** |
| Покупатель: | **Ремонт г. Рославль, ул. Прокатная, дом 180 тел.+7 (982) 288 53 81** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Наименование работ, услуг** | **Кол-вo** | **Ед** | **Цена руб/день** | **Сумма** |
| 1 | Хранение груза размером 34x34x34 | 36 | Дней | 343 | 100 |
| 2 | Хранение груза размером 32x32x32 | 36 | Дней | 455 | 2730 |

|  |  |
| --- | --- |
| **Итого:** | 2830 |
| **В том числе НДС:** | 283 |
| **Всего к оплате:** | 3113 |
| Всего наименований 2, на сумму 3113 руб. | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель | |  | | --- | |  | | Бартель Валерьян Тимурович |
| Менеджер | |  | | --- | |  | | Соловьева Изабелла Петровна |

Стоимость аренды мест в АО «Кипрей»

|  |  |
| --- | --- |
|  | |
| ул. Военного городка тер, дом 128 (Хрупкий) | |
|  | |
| Размер | Стоимость в день |
| 30x45x24 | 800 |
| 44x44x44 | 670 |
|  | |
| ул. Жидков пер, дом 124 (Опасный) | |
|  | |
| Размер | Стоимость в день |
| 23x22x66 | 790 |
|  | |
| ул. Подбельского 7-й проезд, дом 116 (Насыпной) | |
|  | |
| Размер | Стоимость в день |
| 666x33x77 | 560 |
|  | |
| ул. Подбельского 7-й (Хрупкий) | |
|  | |
| Размер | Стоимость в день |
| 77x77x77 | 450 |

28.05.2022 Бартель Валерьян Тимурович

# Приложение В

**Экранные формы**

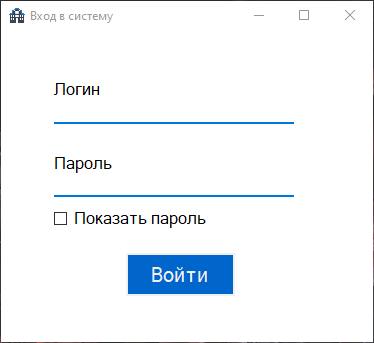


Рисунок В.1 ‒ Форма авторизации

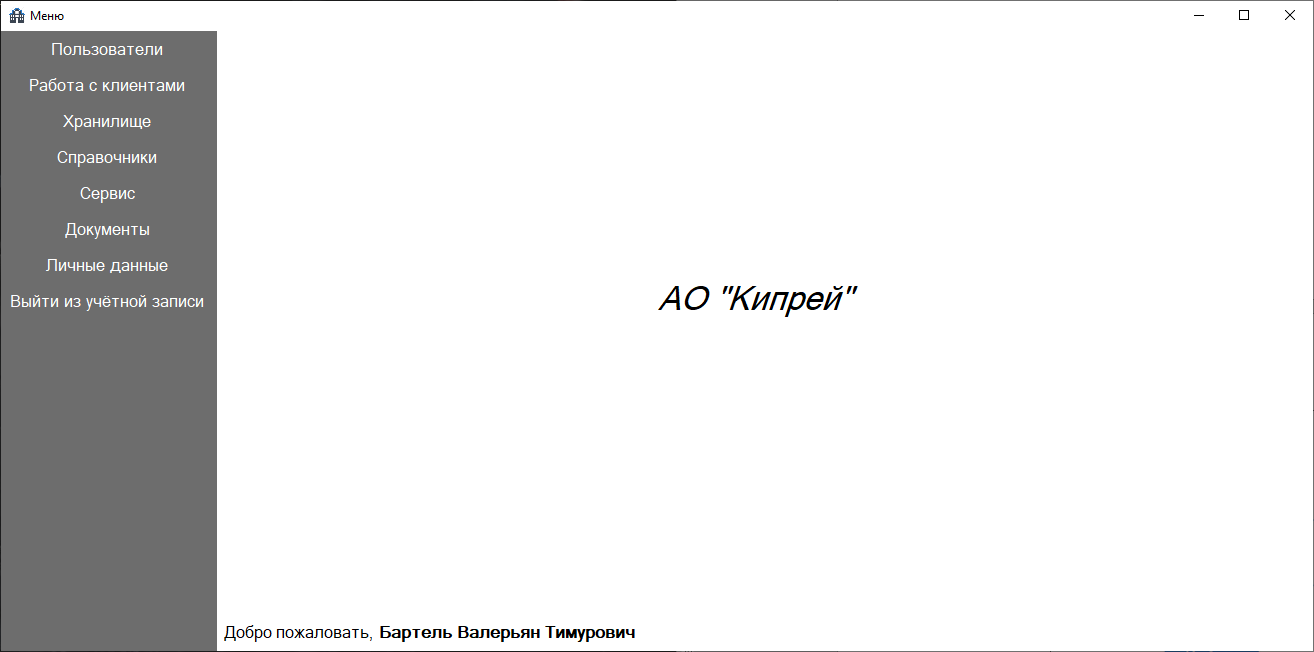


Рисунок В.2 ‒ Главное меню

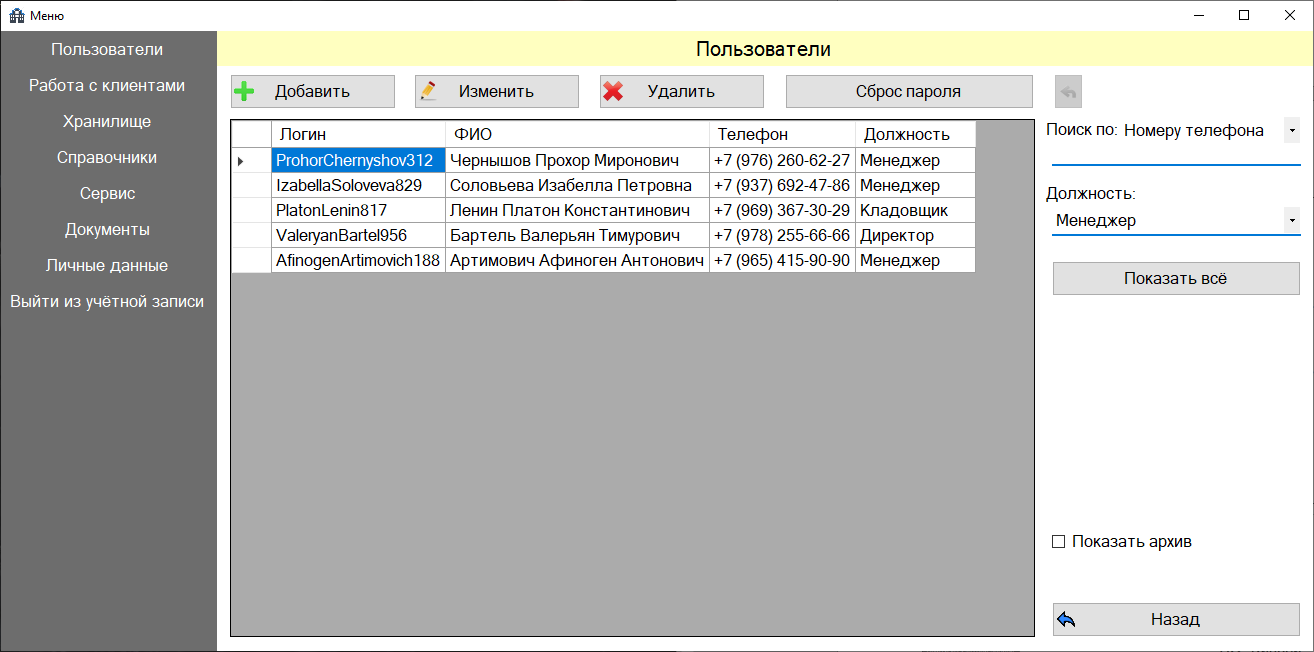


Рисунок В.3 ‒ Список пользователей

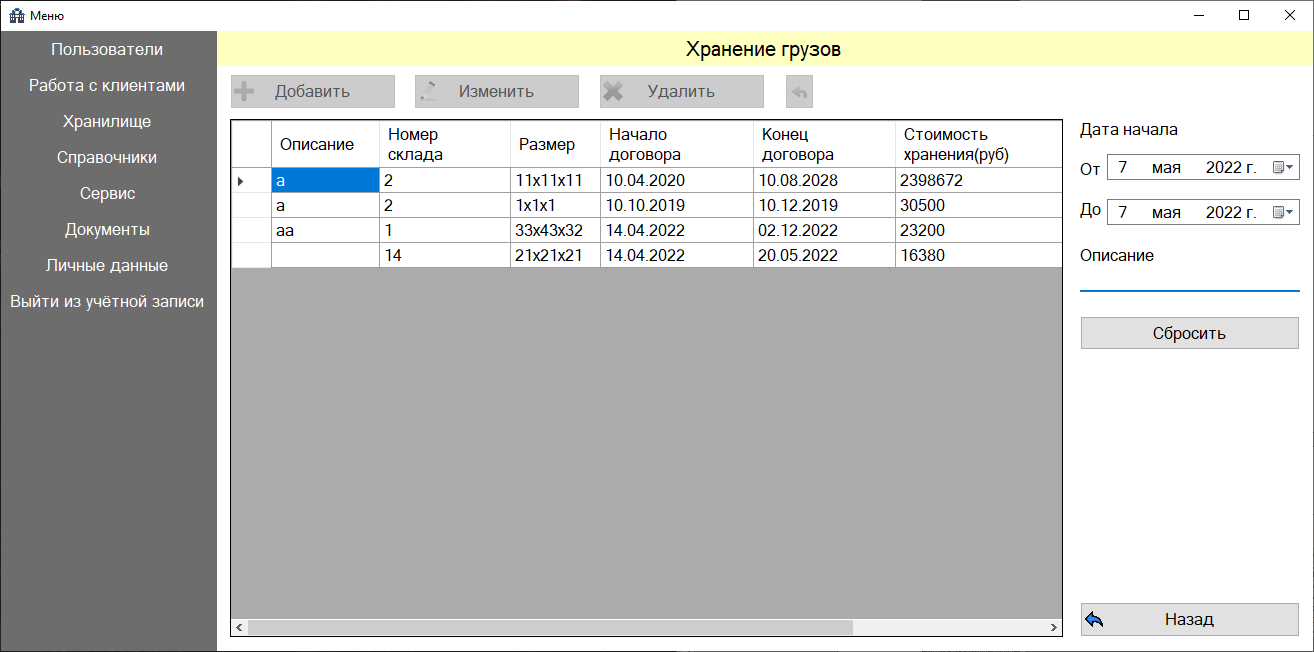


Рисунок В.4 ‒ Список грузов на хранении

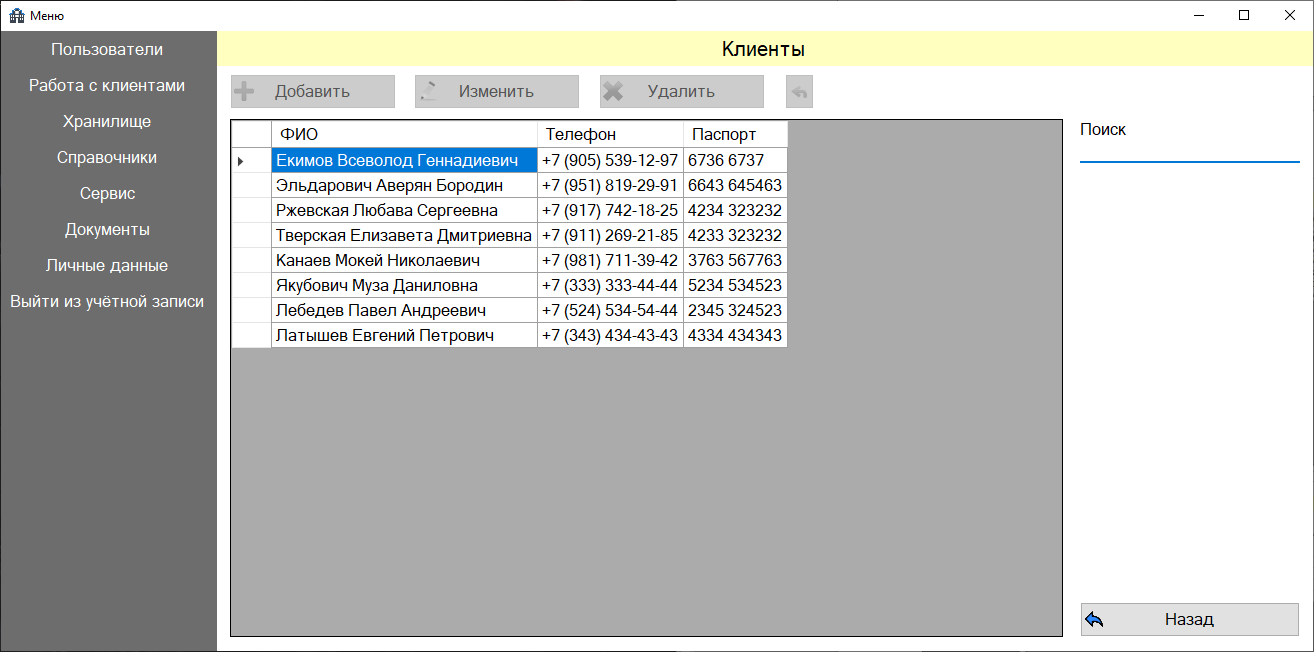


Рисунок В.5 ‒ Список клиентов

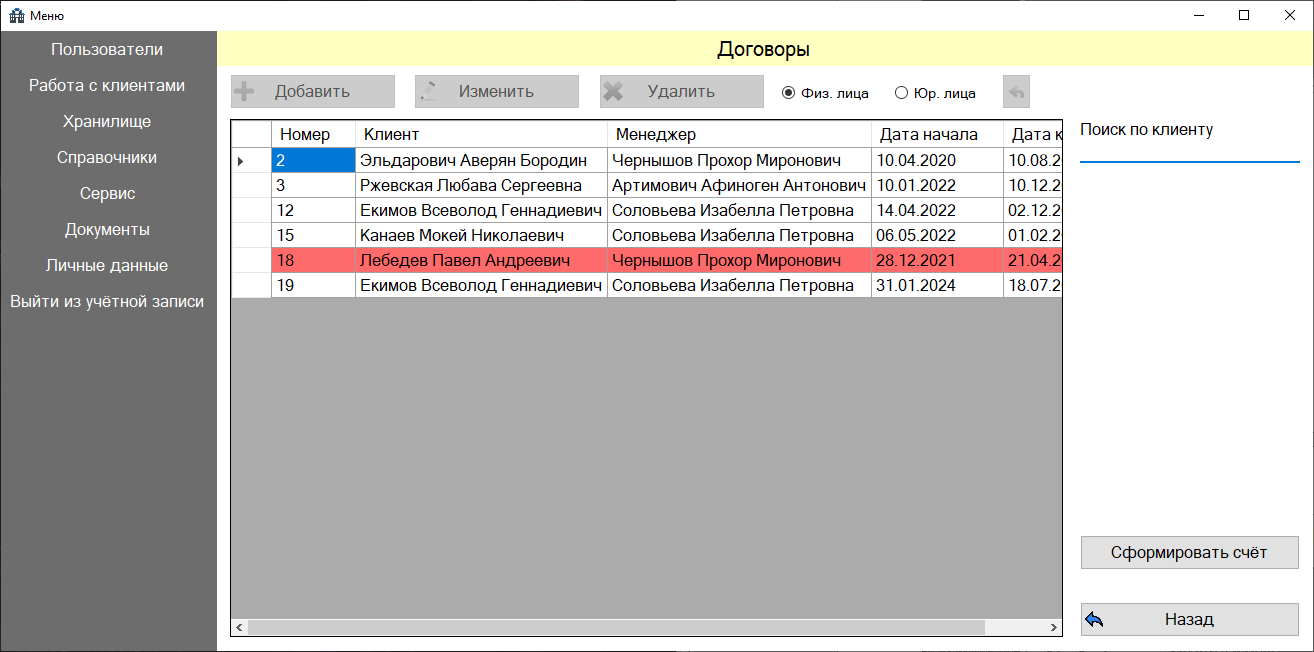


Рисунок В.6 ‒ Список договоров

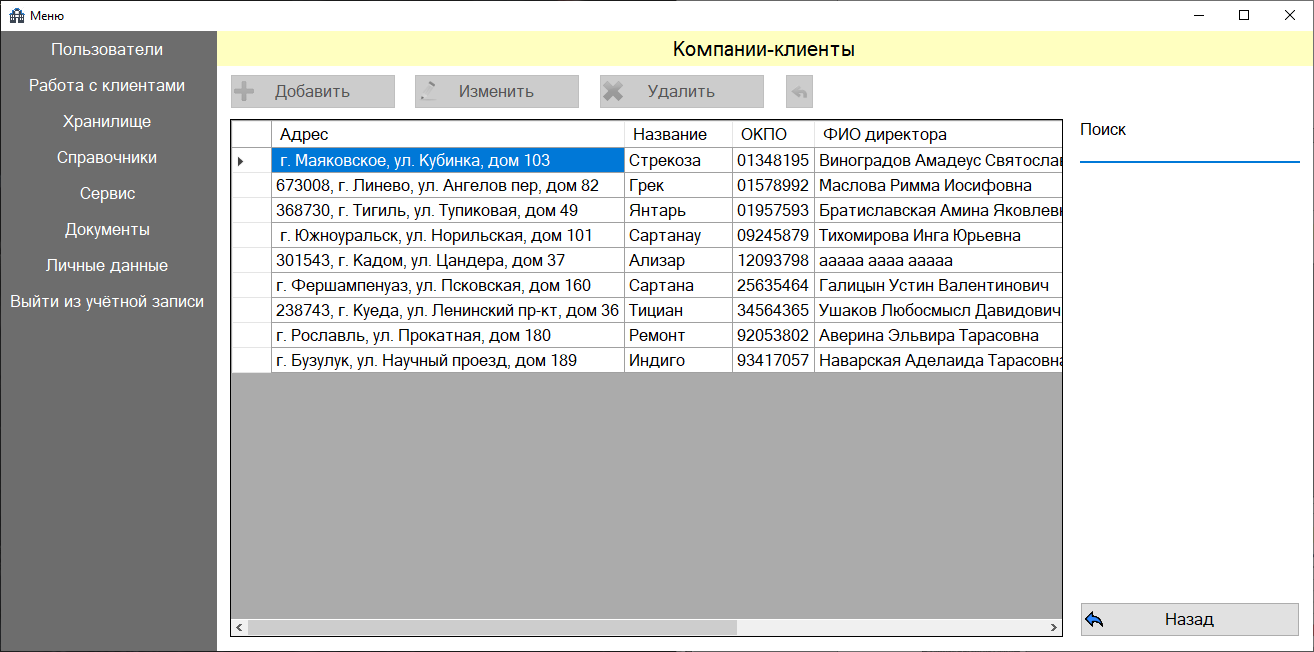


Рисунок В.7 ‒ Список клиентов юридических лиц

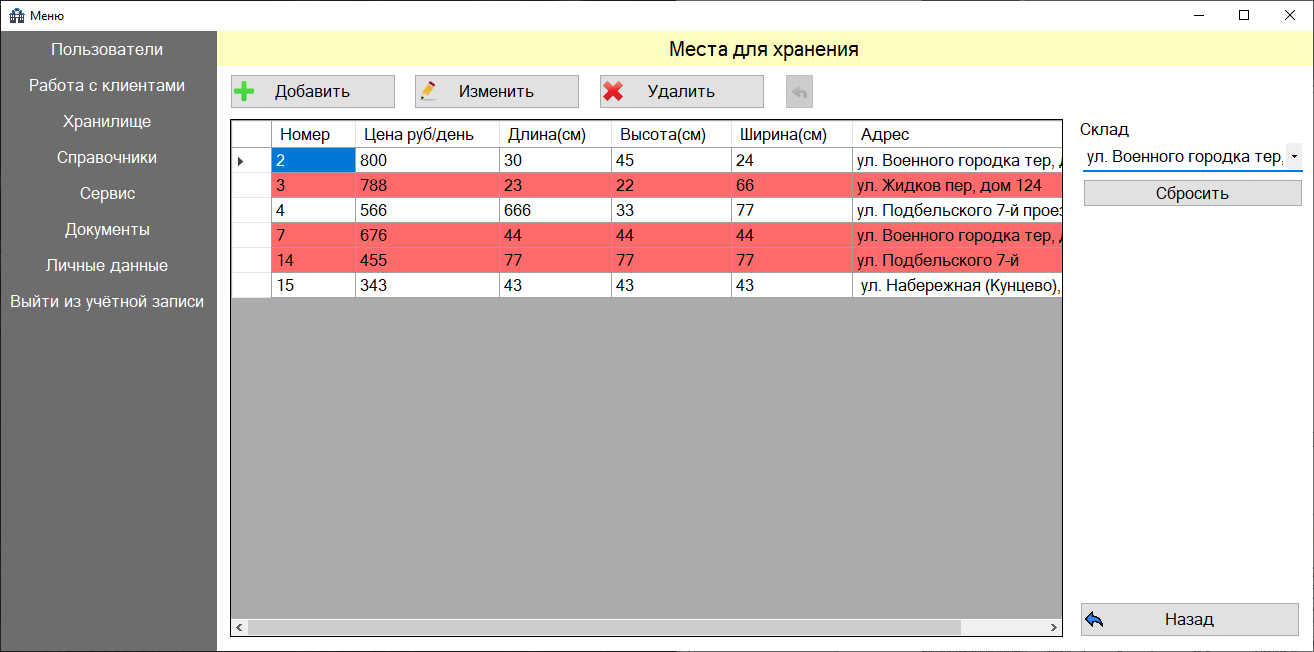


Рисунок В.8 ‒ Список мест для хранения

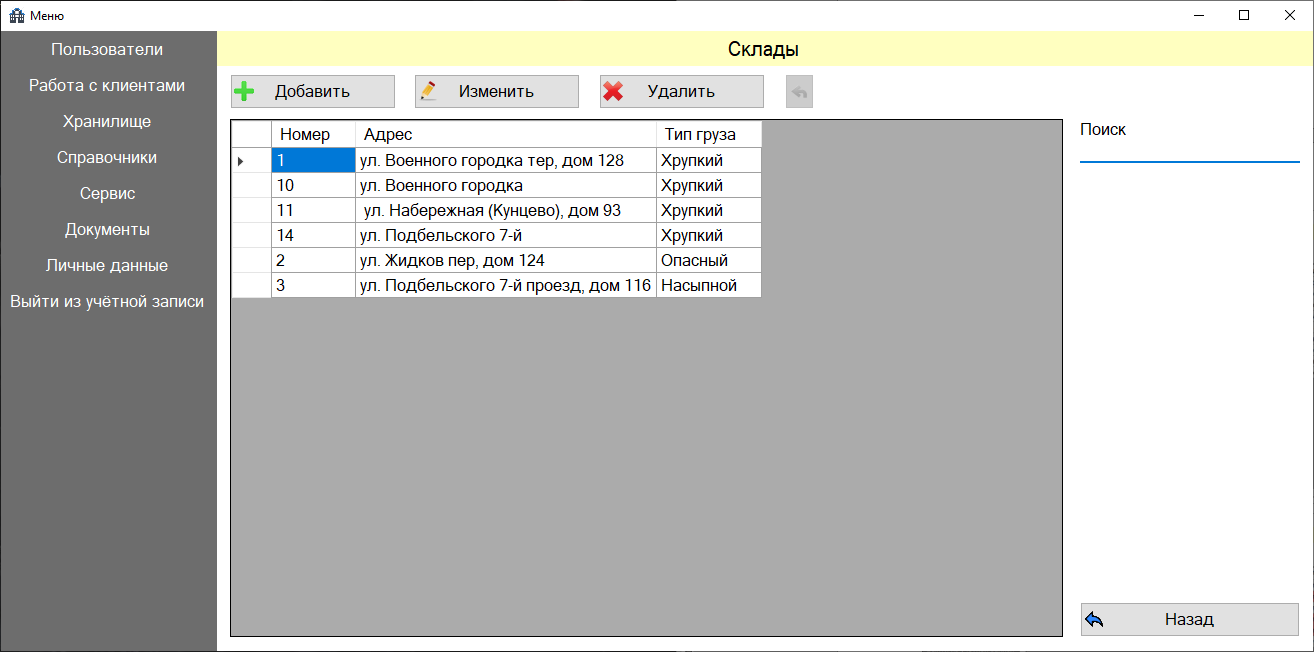


Рисунок В.9 ‒ Список складов

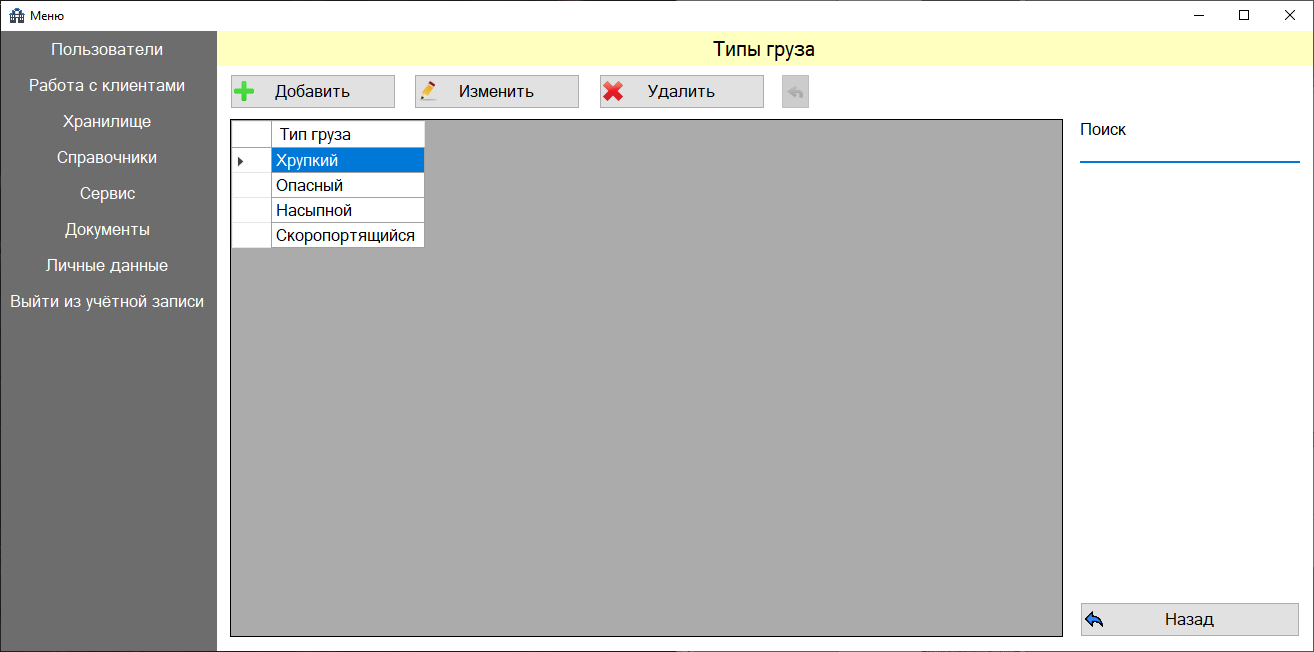


Рисунок В.10 ‒ Список типов грузов

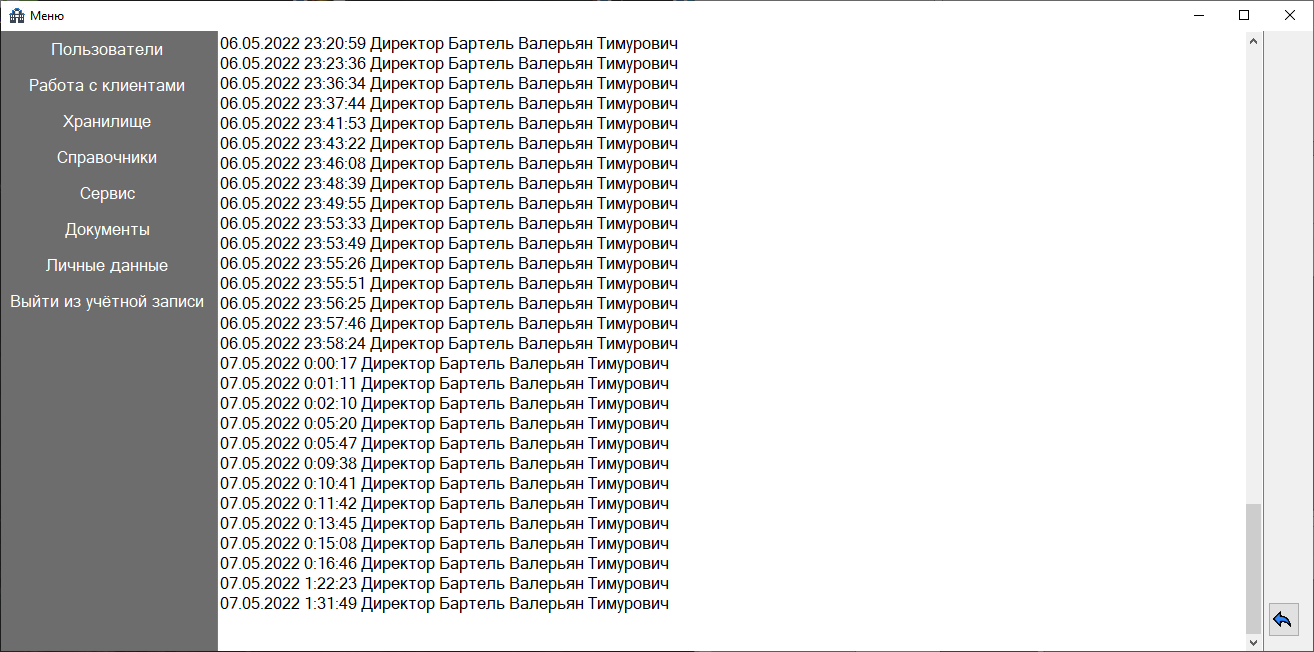


Рисунок В.11 ‒ Журнал активности

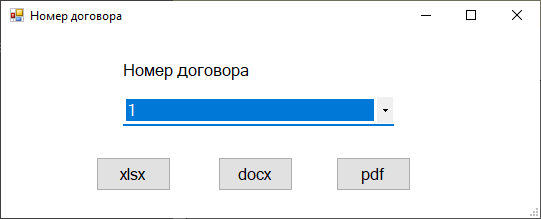


Рисунок В.12 ‒ Форма создания выходного документа

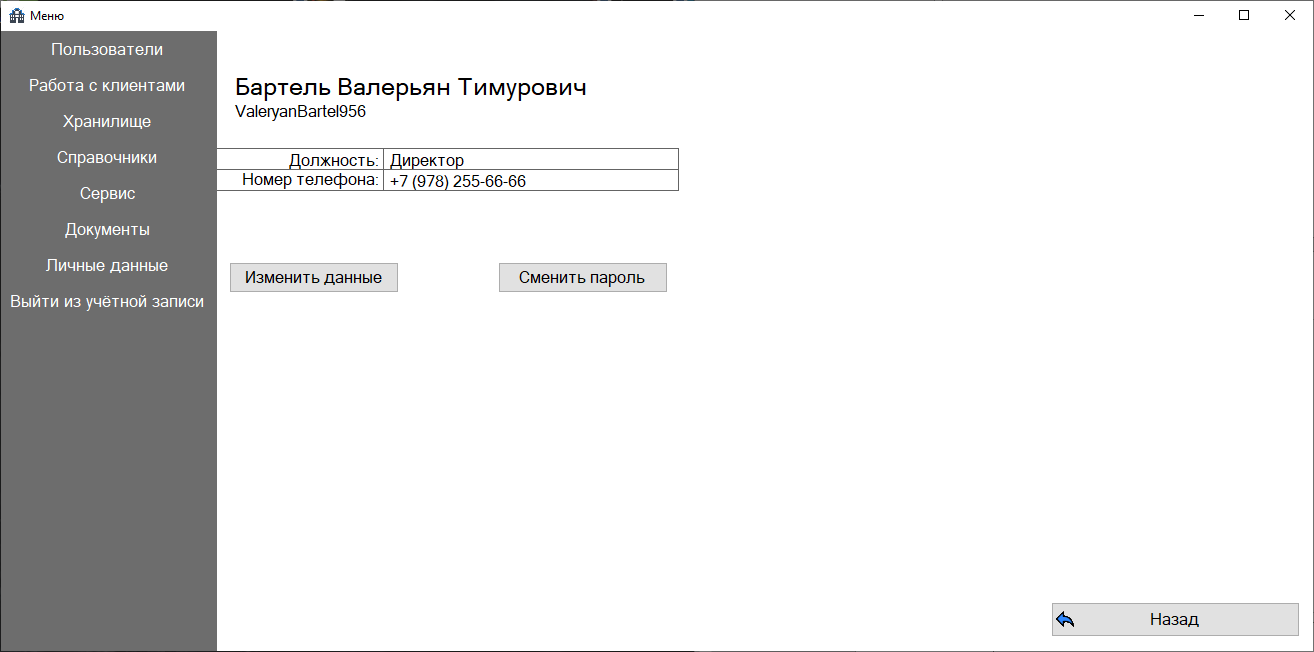


Рисунок В.13 ‒ Личные данные

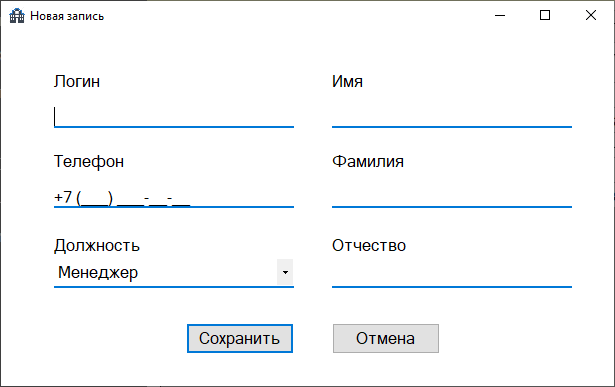


Рисунок В.14 ‒ Форма добавления пользователя

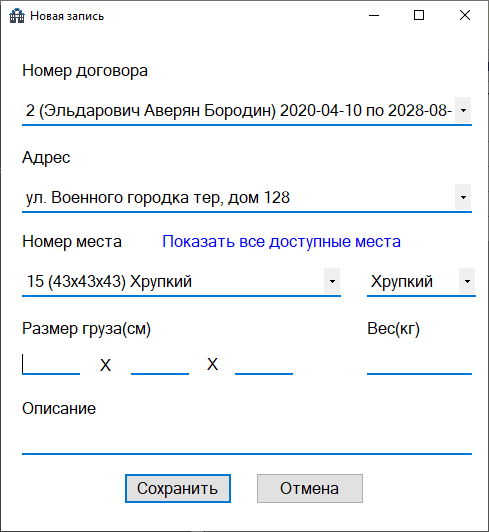


Рисунок В.15 ‒ Форма добавления хранения

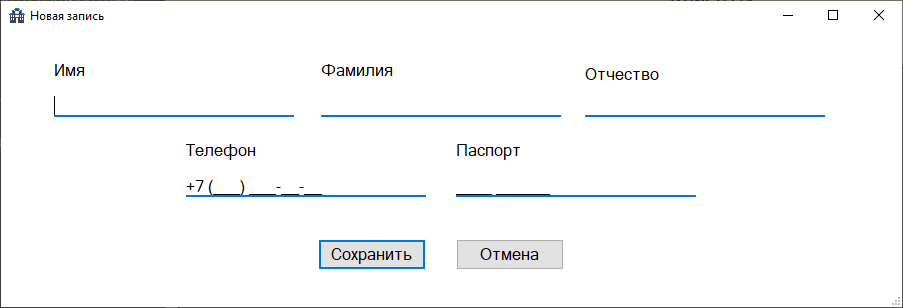


Рисунок В.16 ‒ Форма добавления клиента физ. лица

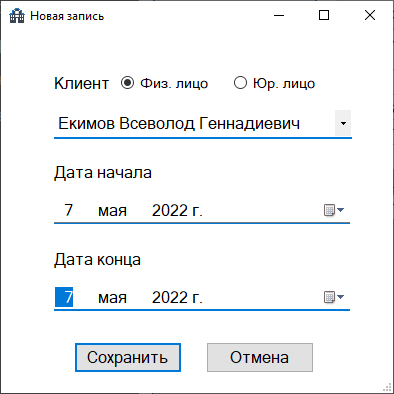


Рисунок В.17 ‒ Форма добавления договора

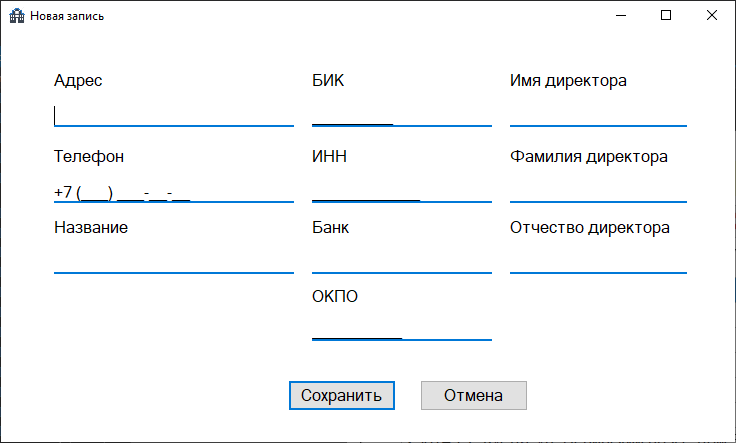


Рисунок В.18 ‒ Форма добавления клиента юр. лица

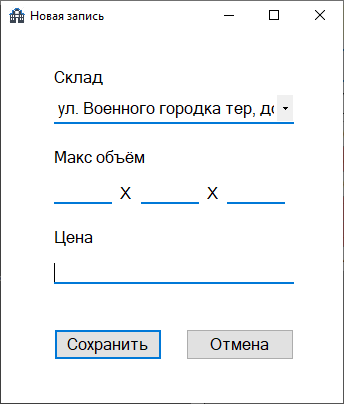


Рисунок В.19 ‒ Форма добавления места на складе

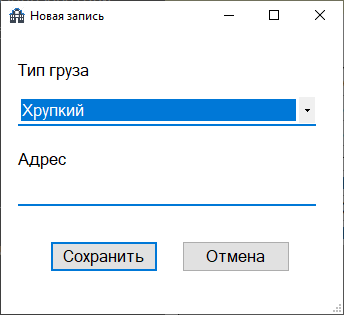


Рисунок В.20 ‒ Форма добавления склада

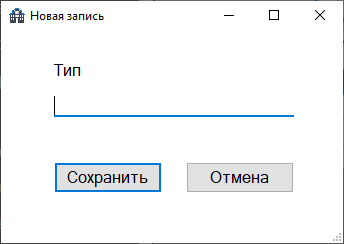
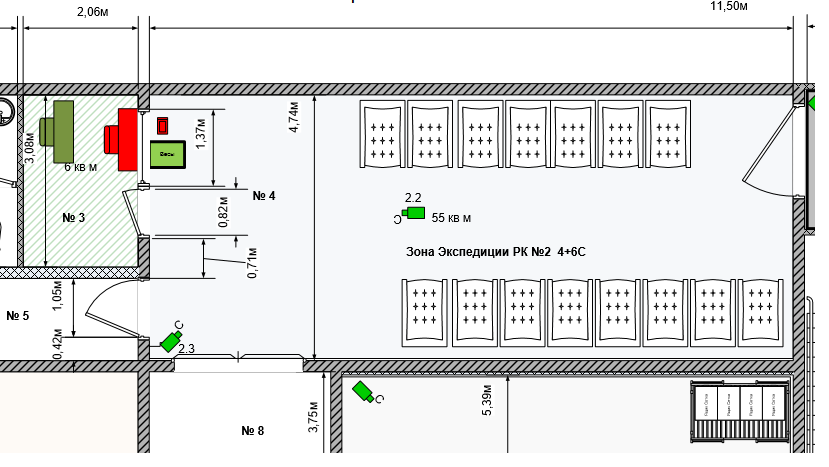
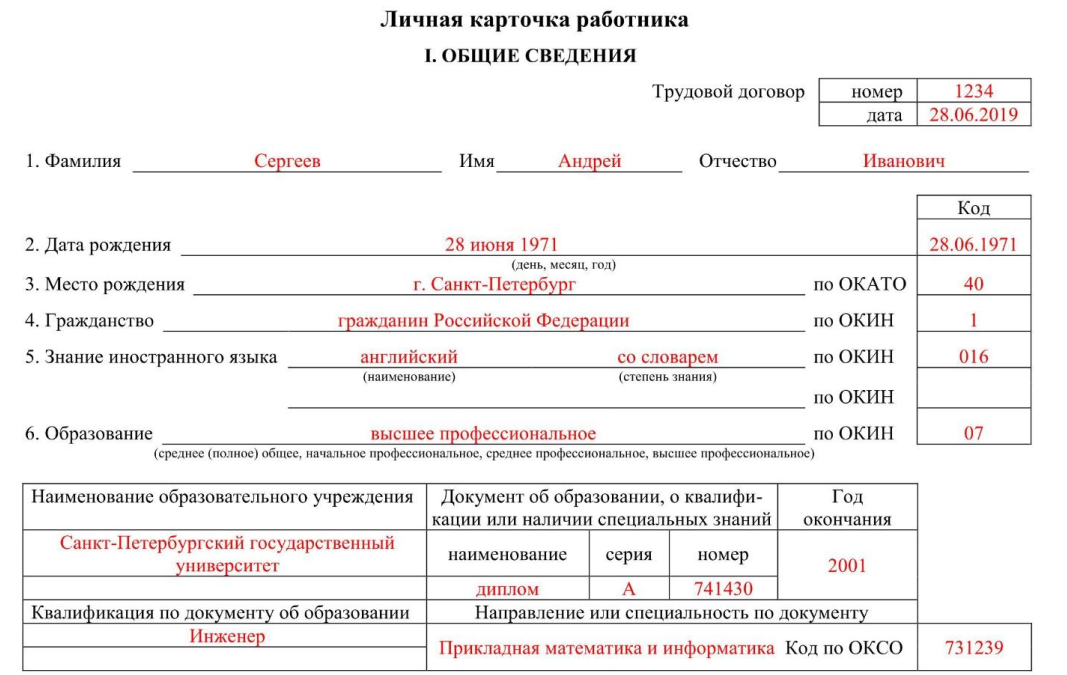


Рисунок В.21 ‒ Форма добавления типа груза

# Приложение Г

**Образцы входных документов**

****



Анкета юридического клиента АО «Кипрей»

|  |  |
| --- | --- |
| **Юридический адрес** | г. Маяковское, ул.  Кубинка, дом 103 |
| **Телефон** | +7 (939) 020 82 63 |
| **Полное название** | Стрекоза |
| **БИК** | 238875679 |
| **ИНН** | 023857239570 |
| **Банк** | Сбербанк |
| **ОКПО** | 01348195 |
| **Имя директора** | Амадеус |
| **Фамилия директора** | Виноградов |
| **Отчество директора** | Святославович |

# Приложение Д

**Содержимое таблиц базы данных**

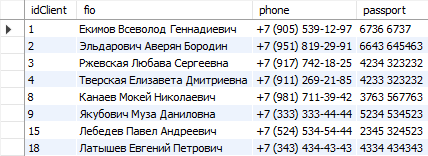
****

Рисунок Д.1 ‒ Содержимое таблицы «clients»

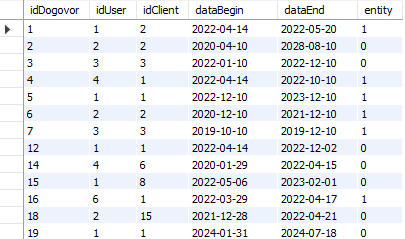
****

Рисунок Д.2 ‒ Содержимое таблицы «dogovori»

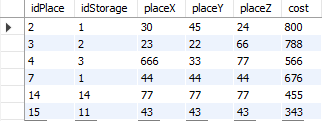
****

Рисунок Д.3 ‒ Содержимое таблицы «places»

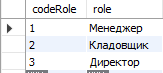
****

Рисунок Д.4 ‒ Содержимое таблицы «s\_roles»

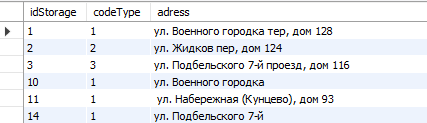
****

Рисунок Д.5 ‒ Содержимое таблицы «storages»

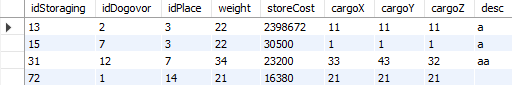
****

Рисунок Д.6 ‒ Содержимое таблицы «storaging»

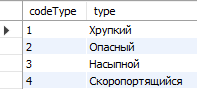
****

Рисунок Д.7 ‒ Содержимое таблицы «types»

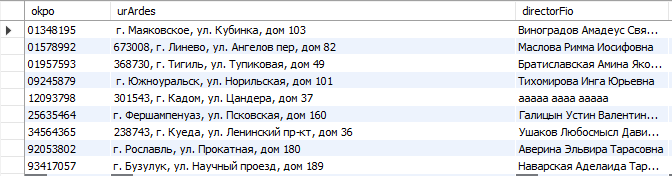
****

Рисунок Д.8 ‒ Содержимое таблицы «uridclients», часть 1

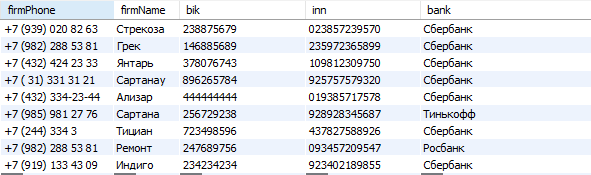
****

Рисунок Д.9 ‒ Содержимое таблицы «uridclients», часть 2

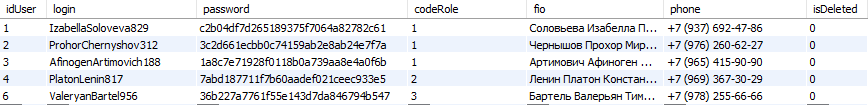
****

Рисунок Д.10 ‒ Содержимое таблицы «users»

# Приложение Е

**Листинг фрагмента программного кода**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Data;

using System.IO;

using System.Linq;

using System.Windows.Forms;

namespace KipreySklad.Classes

{

static class DataBase

{

static MySqlConnection myConnect;

static public MySqlCommand msCommand;

static public DataTable auth = new DataTable();

static public string fio;

static public MySqlDataAdapter msDataAdapter;

static public int userId, role;

static private string [,] buf = { { "clients",

"dogovori", "places", "storages", "storaging",

"types", "uridclients", "users" } };

/// <summary>

/// Соединение с БД

/// </summary>

/// <returns></returns>

static public bool Connect()

{

try

{

myConnect = new MySqlConnection(@"Database " +

@"= sklad; DataSource = localhost; " +

@"User = root; Password = 1111; Charset = utf8;");

myConnect.Open();

msCommand = new MySqlCommand

{

Connection = myConnect

};

msDataAdapter = new MySqlDataAdapter(msCommand);

return true;

}

catch

{

MessageBox.Show("Ошибка при подключении к БД", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

/// <summary>

/// Авторизация

/// </summary>

/// <param name="Login"></param>

/// <param name="Password"></param>

/// <returns></returns>

static public bool Authorization(string Login, string Password)

{

try

{

auth.Clear();

StreamWriter sw = new StreamWriter("Log.txt", true);

msCommand.CommandText = @"SELECT s\_roles.role, users.fio,

users.codeRole,users.idUser

FROM users, s\_roles

WHERE Login = '"

+ Login + "' AND Password = MD5('" + Password + "') " +

"AND users.codeRole = s\_roles.codeRole and isDeleted = '0';";

Object result = msCommand.ExecuteScalar();

if (result != null)

{

msDataAdapter.Fill(auth);

sw.WriteLine($"{DateTime.Now} " +

$"{auth.Rows [0].Field<string>("role")} " +

$"{auth.Rows [0].Field<string>("fio")}");

sw.Close();

role = auth.Rows [0].

Field<int>("codeRole");

fio = auth.Rows [0].Field<string>("fio");

userId = auth.Rows [0].Field<int>("idUser");

return true;

}

//if ((Login == "admin") && (Password == "admin"))

//{

// role = 0;

// sw.Close();

// fio = "admin";

// return true;

//}

else

{

MessageBox.Show("Неверный пароль или логин," +

" повторите попытку", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

sw.Close();

return false;

}

}

catch

{

role = 0;

MessageBox.Show("Ошибка при авторизации...", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

/// <summary>

/// Экспорт БД

/// </summary>

/// <returns></returns>

static public bool Export()

{

try

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.DefaultExt = "sql";

sfd.Filter = "SQL Text File (.sql)|\*.sql";

MySqlBackup msBackup = new MySqlBackup(msCommand);

if (sfd.ShowDialog() != DialogResult.OK)

return false;

File.Create(sfd.FileName).Close();

msBackup.ExportToFile(sfd.FileName);

MessageBox.Show("Резервная копия создана", "",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return true;

}

catch

{

MessageBox.Show("Ошибка во время экспортирования", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

/// <summary>

/// Импорт БД

/// </summary>

/// <returns></returns>

static public bool Import()

{

try

{

MySqlBackup msBackup = new MySqlBackup(msCommand);

OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = "SQL Text File (.sql)|\*.sql";

if (ofd.ShowDialog() == DialogResult.OK)

{

if (!File.ReadLines(ofd.FileName).

Any(line => line.Contains("pricehistory")) ||

!File.ReadLines(ofd.FileName).

Any(line => line.Contains("uridclients")) ||

!File.ReadLines(ofd.FileName).

Any(line => line.Contains("storaging")))

{

MessageBox.Show("Выбран неверный файл", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

msBackup.ImportFromFile(ofd.FileName);

}

else

{

return false;

}

MessageBox.Show("Резервная копия загружена", "",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return true;

}

catch

{

MessageBox.Show("Ошибка во время импортирования", "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

/// <summary>

/// Сохранение таблицы в буфер

/// </summary>

/// <param name="table"></param>

/// <returns></returns>

static public bool Import(string table)

{

try

{

MySqlBackup msBackup = new MySqlBackup(msCommand);

msBackup.ImportFromString(buf [CoordinatesOf(buf, table), 1]);

return true;

}

catch

{

MessageBox.Show("Ошибка во время импортирования", "",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

/// <summary>

/// Поиск элемента в двумерной матрице

/// </summary>

/// <typeparam name="T"></typeparam>

/// <param name="matrix"></param>

/// <param name="value"></param>

/// <returns></returns>

public static int CoordinatesOf<T>(this T [,] matrix, T value)

{

int w = matrix.GetLength(0);

int h = matrix.GetLength(1);

for (int x = 0; x < w; ++x)

{

for (int y = 0; y < h; ++y)

{

if (matrix [x, y].Equals(value))

return x;

}

}

return -1;

}

/// <summary>

/// Выгрузка таблицы из буфера

/// </summary>

/// <param name="table"></param>

/// <returns></returns>

static public bool Export(string table)

{

try

{

MySqlBackup msBackup = new MySqlBackup(msCommand);

msBackup.ExportInfo.TablesToBeExportedList =

new List<string> { table };

buf [CoordinatesOf(buf, table), 1] = msBackup.ExportToString();

return true;

}

catch

{

MessageBox.Show("Ошибка во время экспортирования", "",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

}

}

}

# Приложение Ж

**USB-накопитель**

Селегененко Ю.А., учёт хранения грузов в АО “Кипрей”: Дипломный проект. – Мурманск, 2022. – 72 с.

Научный руководитель: Кузнецова Ксения Валерьевна\_\_\_\_\_

Рецензент: Кабанов Сергей Павлович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дипломный проект выполнен мною совершенно самостоятельно. Все использованные в работе материалы из опубликованной литературы и других источников имеют ссылки на них.

«\_\_\_\_» июня 2022 г.

Селегененко Юрий Алексеевич\_\_\_\_\_\_

(ФИО выпускника, (подпись)

Выпускная квалификационная работа представлена к защите

«\_\_\_» июня 2022 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись начальника учебного отдела)

Защищена «\_\_\_» июня 2022 г.

Протокол № \_\_\_\_ от «\_\_\_» июня 2022 г.

Оценка: «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»