Министерство образования и молодежной политики Свердловской области

государственное автономное профессиональное

образовательное учреждение Свердловской области

«Уральский радиотехнический колледж им. А.С. Попова»

|  |  |
| --- | --- |
|  |  |

Проектирование и разработка программного обеспечения для логистической компании «Быстрее света»

Пояснительная записка к курсовому проекту

по ПМ.03 Участие в интеграции программных модулей

по МДК 03.01 Технология разработки программного обеспечения

РК 09.02.03.332 24 ПЗ

|  |  |  |
| --- | --- | --- |
| Нормоконтролер  \_\_\_\_\_\_\_\_\_\_\_\_ А. В. Беляева  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2023 г. |  | Руководитель  \_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Беляева  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |
|  |  | Разработчик  \_\_\_\_\_\_\_\_\_\_\_\_\_С. Ю. Хромцов  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |
|  |  |  |

Екатеринбург

2023

УТВЕРЖДЕНО

цикловой методической комиссией

информационных технологий

от «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

«\_ »\_ Н.Н Шутова

**ЗАДАНИЕ**

для курсового проектирования по ПМ.03 Участие в интеграции программных модулей по МДК 03.01 Технология разработки программного обеспечения студенту(ке) «4» курса группы «\_\_\_\_\_\_\_\_»

ГАПОУ СО «Уральский радиотехнический колледж им А.С. Попова»

Тема задания

Курсовой проект на указанную тему выполняется студентом в следующем объеме:

* 1. Пояснительная записка состоит из следующих разделов:

Введение

1. Постановка задачи
2. Системный проект
   1. Описание предметной области
   2. Диаграммы вариантов использования
   3. Сценарии вариантов использования
   4. Описание данных
   5. Логическая структура базы данных

3 Технический проект

* 1. Выбор состава технических и программных средств
  2. Физическая структура базы данных
  3. Физическая структура программы

1. Экспериментальная часть
   1. Тестирование приложения
   2. Руководство пользователя.

Заключение

Список использованных источников

Дата выдачи « » 20 г

Срок окончания « » 20 г

Преподаватель А.В. Беляева

Содержание

[Введение 4](#_Toc138080342)

[1 Постановка задачи 5](#_Toc138080343)

[2 Системный проект 7](#_Toc138080344)

[2.1 Описание предметной области 7](#_Toc138080345)

[2.2 Описание данных 9](#_Toc138080346)

[2.3 Логическая структура базы данных 15](#_Toc138080347)

[3 Технический проект 19](#_Toc138080348)

[3.1 Выбор состава технических и программных средств 19](#_Toc138080349)

[3.2 Физическая структура базы данных 20](#_Toc138080350)

[Заключение 24](#_Toc138080351)

[Список использованных источников 26](#_Toc138080352)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

РК 09.02.03 332 24 ПЗ

Разраб.

Хромцов С. Ю.

Провер.

Беляева А. В.

Н. Контр.

Беляева А. В.

Проектирование и разработка программного обеспечения для логистической компании «Быстрее света»

Пояснительная записка

Лит.

Листов

26

ГАПОУ CO

УРТК им. А.С. Попова

Введение

Темой данной курсовой работы является проектирование и разработка программного обеспечения для логистической компании «Быстрее света». Так как современный мир, с его высокой степенью глобализации и увеличивающимся объемом товарооборота, ставит перед логистическими компаниями огромные вызовы, в преодолении которых помогает программное обеспечение. В условиях быстро меняющейся бизнес-среды и растущих ожиданий клиентов, эффективное управление логистическими процессами становится ключевым фактором конкурентоспособности. В данном контексте, разработка программного продукта для логистической компании приобретает особую актуальность.

Логистические компании сегодня сталкиваются с рядом сложных проблем. Во-первых, важно учитывать глобальные изменения в торговле, такие как рост электронной коммерции, увеличение объемов грузоперевозок и стремительное развитие международной логистики. Во-вторых, клиенты становятся все более требовательными, ожидая не только скоростной доставки, но и информацию в режиме реального времени о состоянии своих грузов. В-третьих, необходимость оптимизации маршрутов, управления запасами и сокращения издержек становятся более актуальными, чем когда-либо.

В рамках данного курсового проекта планируется создание базовой версии программного обеспечения, охватывающей основные функциональные элементы логистической компании. Продукт будет разработан с соблюдением современных стандартов разработки программного обеспечения и будет обеспечивать удобный и интуитивно понятный интерфейс для пользователей.

Актуальность данной темы обусловлена ростом объемов грузоперевозок, увеличением ожиданий клиентов и необходимостью оптимизации процессов в логистических компаниях.

Основной целью данного курсового проекта является разработка и реализация программного продукта, который облегчит и улучшит управление логистическими процессами для логистической компании. Для достижения этой цели ставятся следующие задачи:

* проектирование базы данных и архитектуры приложения;
* разработка функционала для отслеживания грузов, сбора статистики, хранения информации о клиентах;
* создание пользовательского интерфейса для клиентов и логистических менеджеров. Данный курсовой проект призван продемонстрировать практическое применение современных технологий в логистической индустрии и подчеркнуть важность развития информационных решений для решения актуальных задач в этой области.

1. Постановка задачи

Целью данной курсовой работы является создание программного обеспечения для логистической компании «Быстрее света», которая позволит вести учёт:

* посылок;
* статусов посылок;
* пунктов выдачи;
* складов;
* машин.

Формировать отчеты и статистику:

* клиентов;
* машин;
* сотрудников.

Даст возможность клиентам самостоятельно ознакомиться с услугами и рассчитать стоимость доставки на сайте. А также исключает ошибки, допускаемые при ручной обработке данных.

Для достижения поставленной цели необходимо решить следующие задачи:

* проанализировать предметную область, изучить сферу деятельности логистической компании, для их автоматизации с помощью базы данных;
* написать варианты использования программного обеспечения;
* проанализировать данные, которые будут храниться в базе данных;
* описать логическую структуру базы данных, которая будет описывать как происходит обмен информацией в базе данных;
* выбрать технические и программные средства для реализации проекта;
* описать физическую структуру программы. Описать диаграмму классов разрабатываемого программного обеспечения;
* разработать базу данных для логистической компании «Быстрее света»;
* разработать программное обеспечение для логистической компании «Быстрее света»;
* протестировать разработанное приложение;
* составить руководство пользователя, в котором будут расписаны все основные функциональные возможности разработанного программного обеспечения

В первом разделе «постановка задачи» описывается цель курсового проекта и задачи, которые необходимо выполнить для достижения цели.

Во втором разделе «системный проект» описана предметная область, и логическая структура базы данных.

В третьем разделе «технический проект» происходит выбор программных и технических средств, а также описывается физическая структура базы данных.

В последнем разделе «экспериментальная часть» описан процесс тестирования приложения и его результаты, методики, которые были использованы при тестировании и приведено подробное руководство для пользователя, с описанием всех возможностей разработанного приложения

1. Системный проект
   1. Описание предметной области

Компания «Быстрее света» занимается доставкой посылок. Клиент обладает возможностью отправить посылку при обращении в пункт выдачи. Для отправки посылки необходимо оформить документы у сотрудника пункта. Документы включают в себя акт приема и накладную на товар, для заполнения данных документов клиент должен предоставить паспортные данные, а также указать пункт выдачи, в который необходимо доставить товар.

Сотрудник на основании этих данных должен заполнить акт приема, указав в нем свой личный номер и номер пункта, где был принят товар. Сотрудник обязан проверить, что товар соответствует регламентам перевозки, в противном случае сотрудник сообщает клиенту причину, по которой товар не может быть принят.

При оформлении посылки высчитывается стоимость доставки с учетом веса и категории груза. Далее в акт приема вписываются характеристики посылки: вес, габариты и тип упаковки, порядковый номер. После клиент оплачивает, ставит подпись в акте приема и получает его копию.

Принятые посылки передаются на перевозку. Перевозка осуществляется при помощи автомобилей компании. После принятия посылок работник пункта выдачи заносит в акт приема со склада полученные посылки, в акт отправки на склад переданные на склад посылки.

При совершении перевозки, на складе в машину загружаются посылки для отправки на определенный склад или в пункт выдачи. После загрузки водитель получает сопроводительные документы, также ему сообщается конечная точка маршрута. Если перевозка осуществлялась в другой регион тогда, после того как водитель доставил посылки, они сортируются для развоза по пунктам выдачи.

При доставке посылки в пункт выдачи клиенту сообщается срок, до которого он должен забрать посылку. В случае если клиент не явился в пункт до озвученного срока, посылка отправляется на утилизацию.

Клиент приходит за заказом. Проверяет его целостность, ставит подпись и забирает товар.

* 1. Диаграммы вариантов использования

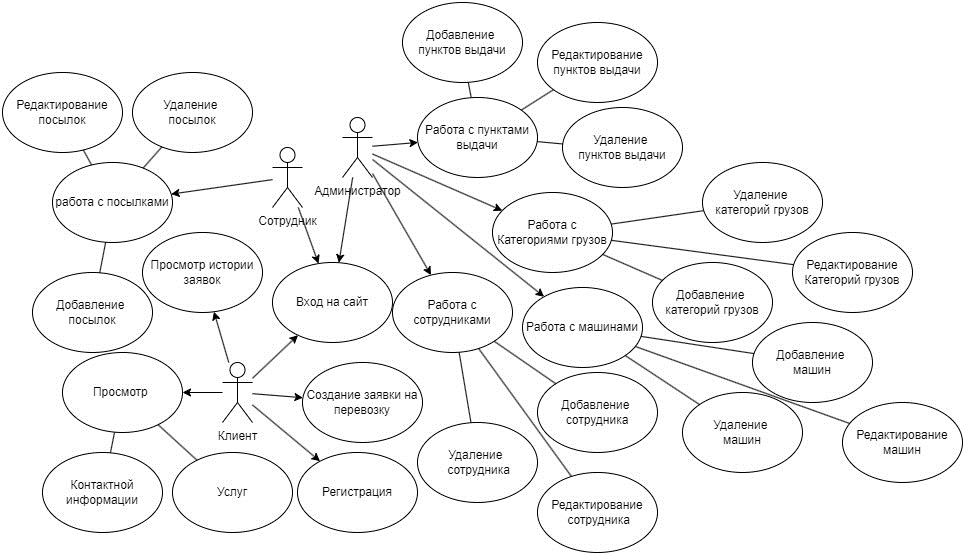


Рисунок 2.2.1 Диаграмма вариантов использования

Возможные действия клиента:

* создание заявки на перевозку;
* вход на сайт;
* просмотр истории заявок;
* просмотр контактной информации;
* просмотр услуг;
* регистрация на сайте.

Доступные действия сотрудника:

* вход на сайт;
* добавление посылок;
* редактирование посылок;
* удаление посылок.

Доступные действия администратора:

* вход на сайт;
* добавление пунктов выдачи;
* редактирование пунктов выдачи;
* удаление пунктов выдачи;
* удаление категорий грузов;
* редактирование категорий грузов;
* добавление категорий грузов;
* добавление машин;
* редактирование машин;
* удаление машин;
* добавление сотрудника;
* редактирование сотрудника;
* удаление сотрудника.
  1. Сценарии вариантов использования

Для логистической агентства могут существовать различные сценарии использования сайта:

1. регистрация на сайте – клиент хочет зарегистрироваться на сайте.

Основной поток событий:

* клиент нажимает на кнопку «Регистрация» на главной странице;
* система открывает форму регистрации;
* клиент заполняет поля: ФИО, город, регион, дату рождения, паспортные данные, логин, пароль, почта, телефон, и нажимает на кнопку «Сохранить»;
* система обрабатывает запрос на создание новой учётной записи в базе данных и выводит информацию о подтверждении регистрации на главную страницу.

Альтернативные потоки:

* при неверном заполнении данных пользователем, выводится сообщение о необходимости исправления тех или иных ошибок.

1. просмотр собственных заявок – покупатель хочет посмотреть историю своих заявок.

Основной поток событий:

* покупатель открывает историю своих заявок, нажав на кнопку «История заявок»;
* система выводит данные о покупателе и об его заявках из базы данных, где указаны текущие статусу заявок и полная информация о них;
* покупатель выбирает из списка автомобиль, нажав на кнопку «Просмотр»;
* система обрабатывает запрос и выводит информацию о заявке из базы данных и выводит информацию о её подтверждении на главную страницу.

Предусловия:

* для просмотра истории пользователь должен быть авторизированным.

1. добавление информации о сотруднике – администратор хочет добавить нового сотрудника.

Основной поток событий:

* администратор нажимает на кнопку «Добавить сотрудника» на странице просмотра персонала;
* система загружает форму для добавления информации о сотруднике;
* администратор заполняет поля: ФИО, паспортные данные, почта, телефон, логин, пароль, должность и нажимает на кнопку «Сохранить»;
* система обрабатывает запрос и обновляет введённую информацию в базе данных и выводит информацию о подтверждении операции на главную страницу.

Альтернативный поток:

* администратор не верно указывает формат данных для заполнения полей. Программное обеспечение выводит сообщение о неверном заполнение данных, а также предложит повторный ввод и укажет на ошибки.

Предусловия:

* для добавления информации сотруднике, администратор обязательно должен быть авторизирован.

1. создание заявки на перевозку – клиент хочет создать заявку на перевозку.

Основной поток событий:

* клиент нажимает на кнопку «Создать заявку на перевозку»;
* система загружает форму для создания заявки на перевозку;
* клиент заполняет все поля и нажимает на кнопку «Отправить заявку»;
* система обрабатывает запрос и обновляет введённую информацию в базе данных.

Альтернативный поток:

* клиент не верно указывает формат данных для заполнения полей. Программное обеспечение выводит сообщение о неверном заполнение данных, а также предложит повторный ввод и укажет на ошибки.

Предусловия:

* для создания заявки на перевозку, администратор обязательно должен быть авторизирован.

1. вход на сайт – пользователь входит в свой аккаунт для доступа к персонализированным услугам и информации.

Основной поток событий:

* пользователь нажимает кнопку «Вход»;
* система загружает страницу входа;
* пользователь вводит логин и пароль для входа в аккаунт;
* система делает запрос в базу данных и сверяет логин и пароль;
* пользователь авторизируется;
* система загружает главную страницу.

Альтернативный поток:

* при неверном заполнении, выводится сообщение о необходимости исправления тех или иных ошибок;
* если пользователь забыл пароль, он может воспользоваться функцией сброса пароля, которая отправит ему инструкции на восстановление пароля на указанный адрес электронной почты.

Предусловия:

* для входа в аккаунт, пользователь должен быть зарегистрирован.

1. просмотр контактной информации – пользователь просматривает контактную информацию сайта.

Основной поток событий:

* пользователь заходит на сайт;
* система загружает главную страницу сайта;
* пользователь нажимает кнопку «Контакты»;
* система загружает страницу контактной информации;
* пользователь смотрит контактную информацию.

Предусловия:

* для просмотра контактной информации нужно зайти на сайт.

1. просмотр услуг – пользователь просматривает услуги компании.

Основной поток событий:

* пользователь заходит на сайт;
* система загружает главную страницу сайта;
* пользователь нажимает кнопку «Услуги»;
* система загружает страницу услуг компании;
* пользователь смотрит услуги компании.

Предусловия:

* для просмотра услуг нужно зайти на сайт.

1. добавление посылки – сотрудник добавляет информацию о посылке клиента в базу данных.

Основной поток событий:

* сотрудник нажимает кнопку «Добавление посылки»;
* система загружает страницу добавления посылки;
* сотрудник заполняет в поля всю информацию о посылке и нажимает «Сохранить»;
* система делает запрос в базу данных и добавляет информацию о посылке.

Альтернативный поток:

* при неверном заполнении, выводится сообщение о необходимости исправления тех или иных ошибок.

Предусловия:

* для добавления посылки, пользователь должен быть авторизирован.

1. редактирование посылки – сотрудник изменяет информацию о посылке клиента.

Основной поток событий:

* сотрудник переходит в раздел администрирование и нажимает кнопку «Посылки»;
* система делает запрос в базу данных и загружает страницу посылок;
* сотрудник выбирает конкретную посылку и нажимает на кнопку «Редактирование посылки»;
* система делает запрос в базу данных и загружает страницу редактирования посылки;
* сотрудник изменяет информацию в полях о посылке и нажимает «Сохранить»;
* система делает запрос в базу данных и обновляет информацию о посылке.

Альтернативный поток:

* при неверном заполнении, выводится сообщение о необходимости исправления тех или иных ошибок.

Предусловия:

* для редактирования посылки, пользователь должен быть авторизирован.

1. удаление посылки – сотрудник удаляет посылку клиента.

Основной поток событий:

* сотрудник переходит в раздел администрирование и нажимает кнопку «Посылки»;
* система делает запрос в базу данных и загружает страницу посылок;
* сотрудник выбирает конкретную посылку и нажимает на кнопку «Удалить посылку»;
* сотрудник удаляет посылку;
* система удаляет выбранную запись из базы данных.

Предусловия:

* для удаления пункта выдачи пользователь обязательно должен быть авторизирован.

1. добавление пункта выдачи – администратор добавляет информацию о пункте выдачи в базу данных.

Основной поток событий:

* администратор нажимает кнопку «Добавление пункта выдачи»;
* система загружает страницу добавления пункта выдачи;
* администратор заполняет в поля всю информацию о посылке и нажимает «Сохранить»;
* система делает запрос в базу данных и добавляет информацию о пункте выдачи.

Альтернативный поток:

* при неверном заполнении, выводится сообщение о необходимости исправления тех или иных ошибок.

Предусловия:

* для добавления пункта выдачи, пользователь должен быть авторизирован.

1. редактирование пункта выдачи – администратор изменяет информацию о пункте выдачи.

Основной поток событий:

* администратор переходит в раздел администрирование и нажимает кнопку «Пункты выдачи»;
* система делает запрос в базу данных и загружает страницу пунктов выдачи;
* администратор выбирает конкретный пункт выдачи и нажимает на кнопку «Редактирование пункта выдачи»;
* система делает запрос в базу данных и загружает страницу редактирования пункта выдачи;
* администратор изменяет информацию в полях о пункте выдачи и нажимает «Сохранить»;
* система делает запрос в базу данных и обновляет информацию о пункте выдачи.

Альтернативный поток:

* при неверном заполнении, выводится сообщение о необходимости исправления тех или иных ошибок.

Предусловия:

* для редактирования пункта выдачи, пользователь должен быть авторизирован.

1. удаление пункта выдачи– администратор удаляет пункт выдачи.

Основной поток событий:

* администратор переходит в раздел администрирование и нажимает кнопку «Пункты выдачи»;
* система делает запрос в базу данных и загружает страницу пунктов выдачи;
* администратор выбирает конкретный пункт выдачи и нажимает на кнопку «Удалить пункт выдачи»;
* администратор удаляет пункт выдачи;
* система удаляет выбранную запись из базы данных.

Предусловия:

* для удаления пункта выдачи пользователь обязательно должен быть авторизирован.

1. добавление категорий груза – администратор добавляет информацию о категории груза в базу данных.

Основной поток событий:

* администратор нажимает кнопку «Добавление категории груза»;
* система загружает страницу добавления категории груза;
* администратор заполняет в поля всю информацию о категории груза и нажимает «Сохранить»;
* система делает запрос в базу данных и добавляет информацию о категории груза.

Альтернативный поток:

* при неверном заполнении, выводится сообщение о необходимости исправления тех или иных ошибок.

Предусловия:

* для добавления категории груза, пользователь должен быть авторизирован.

1. редактирование пункта выдачи – администратор изменяет информацию о пункте выдачи.

Основной поток событий:

* администратор переходит в раздел администрирование и нажимает кнопку «Категории груза»;
* система делает запрос в базу данных и загружает страницу категорий груза;
* администратор выбирает конкретную категорию груза и нажимает на кнопку «Редактирование категории груза»;
* система делает запрос в базу данных и загружает страницу редактирования категории груза;
* администратор изменяет информацию в полях о категории груза и нажимает «Сохранить»;
* система делает запрос в базу данных и обновляет информацию о категории груза.

Альтернативный поток:

* при неверном заполнении, выводится сообщение о необходимости исправления тех или иных ошибок.

Предусловия:

* для редактирования категорим груза, пользователь должен быть авторизирован.

1. удаление пункта выдачи– администратор удаляет пункт выдачи.

Основной поток событий:

* администратор переходит в раздел администрирование и нажимает кнопку «Категории груза»;
* система делает запрос в базу данных и загружает страницу категорий груза;
* администратор выбирает конкретную категорию груза и нажимает на кнопку «Удалить категорию груза»;
* администратор удаляет категорию груза;
* система удаляет выбранную запись из базы данных.

Предусловия:

* для удаления категории груза пользователь обязательно должен быть авторизирован.

1. добавление машины – администратор добавляет информацию о машине в базу данных.

Основной поток событий:

* администратор нажимает кнопку «Добавление машины»;
* система загружает страницу добавления машины;
* администратор заполняет в поля всю информацию о машине и нажимает «Сохранить»;
* система делает запрос в базу данных и добавляет информацию о машине.

Альтернативный поток:

* при неверном заполнении, выводится сообщение о необходимости исправления тех или иных ошибок.

Предусловия:

* для добавления машины, пользователь должен быть авторизирован.

1. редактирование машины – администратор изменяет информацию о машине.

Основной поток событий:

* администратор переходит в раздел администрирование и нажимает кнопку «Машины»;
* система делает запрос в базу данных и загружает страницу машин;
* администратор выбирает конкретную машину и нажимает на кнопку «Редактирование машины»;
* система делает запрос в базу данных и загружает страницу редактирования машины;
* администратор изменяет информацию в полях о машине и нажимает «Сохранить»;
* система делает запрос в базу данных и обновляет информацию о машине.

Альтернативный поток:

* при неверном заполнении, выводится сообщение о необходимости исправления тех или иных ошибок.

Предусловия:

* для редактирования машины, пользователь должен быть авторизирован.

1. удаление машины– администратор удаляет машину.

Основной поток событий:

* администратор переходит в раздел администрирование и нажимает кнопку «Машины»;
* система делает запрос в базу данных и загружает страницу машин;
* администратор выбирает конкретную машину и нажимает на кнопку «Удалить машину»;
* администратор удаляет машину;
* система удаляет выбранную запись из базы данных.

Предусловия:

* для удаления машину пользователь обязательно должен быть авторизирован.

1. редактирование сотрудника – администратор изменяет информацию о сотруднике.

Основной поток событий:

* администратор переходит в раздел администрирование и нажимает кнопку «Сотрудники»;
* система делает запрос в базу данных и загружает страницу сотрудников;
* администратор выбирает конкретного сотрудника и нажимает на кнопку «Редактирование сотрудника»;
* система делает запрос в базу данных и загружает страницу редактирования сотрудника;
* администратор изменяет информацию в полях о сотруднике и нажимает «Сохранить»;
* система делает запрос в базу данных и обновляет информацию о сотруднике.

Альтернативный поток:

* при неверном заполнении, выводится сообщение о необходимости исправления тех или иных ошибок.

Предусловия:

* для редактирования сотрудника, пользователь должен быть авторизирован.

1. удаление сотрудника– администратор удаляет сотрудника.

Основной поток событий:

* администратор переходит в раздел администрирование и нажимает кнопку «Сотрудники»;
* система делает запрос в базу данных и загружает страницу сотрудников;
* администратор выбирает конкретного сотрудника и нажимает на кнопку «Удалить сотрудника»;
* администратор удаляет сотрудника;
* система удаляет выбранную запись из базы данных.

Предусловия:

* для удаления сотрудника пользователь обязательно должен быть авторизирован.

### Описание данных

На основе анализа работы логистической компании, будут сформированы следующие сущности:

* пункт\_Приема\_Выдачи;
* склад;
* посылка;
* категория\_груза;
* машина;
* пользователь.

Описание сущности «Пункт\_Приема\_Выдачи» представлено в

таблице 2.4.1.

Таблица 2.4.1 – Описание данных сущности «Пункт\_Приема\_Выдачи»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Примечание |
| 1 | 2 | 3 | 4 |
| ID\_пункта | Числовой | Id пункта выдачи | Первичный ключ |
| Регион | Текстовый | Регион, в котором находится пункт выдачи | Максимальный размер 200 |
| Город | Текстовый | Город, в котором находится пункт выдачи | Максимальный размер 200 |
| Улица | Текстовый | Улица, на которой находится пункт выдачи | Максимальный размер 200 |
| Дом | Текстовый | Номер дома | Максимальный размер 10 |
| Корпус | Текстовый | Корпус дома | Максимальный размер 10 |

Описание сущности «Склад» представлено в таблице 2.4.2.

Таблица 2.4.2 – Описание данных сущности «Склад»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Примечание |
| 1 | 2 | 3 | 4 |
| ID\_склада | Числовой | Id Склада | Первичный ключ |
| Регион | Текстовый | Регион, в котором находится склад | Максимальный размер 200 |
| Город | Текстовый | Город, в котором находится склад | Максимальный размер 200 |
| Улица | Текстовый | Улица, на которой находится склад | Максимальный размер 200 |
| Дом | Текстовый | Номер склада | Максимальный размер 10 |
| Корпус | Текстовый | Корпус склада | Максимальный размер 10 |

Описание сущности «Посылка» представлено в таблице 2.4.3.

Таблица 2.4.3 – Описание данных сущности «Посылка»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Примечание |
| 1 | 2 | 3 | 4 |
| ID\_посылки | Числовой | Id посылки | Первичный ключ |
| Комментарии | Текстовый | Комментарии к заказу | Необязательный, максимальный размер 500 |
| Вес | Числовой | Вес | Неотрицательный, дробный указывается в килограммах |
| Дата\_принятия | Дата | Дата принятия | Устанавливается дата, когда посылка принята от клиента |
| Дата\_доставки\_в\_ пункт\_выдачи | Дата | Дата доставки в пункт выдачи | Устанавливается дата доставки в пункт выдачи |
| Дата \_выдачи | Дата | Дата выдачи | Не может быть меньше даты принятия |
| Длина | Текстовый | Габариты | Длина указывается в метрах |
| Высота | Текстовый | Габариты | Высота указывается в метрах |
| Ширина | Текстовый | Габариты | Ширина указывается в метрах |
| Стоимость\_доставки | Числовой | Стоимость доставки | * Неотрицательная, указывается в рублях, дробный тип. |
| Статус | Текстовый | Статус | Может иметь следующие значения:   * принят от клиента; * отправлен на склад; * отправлен на склад в город назначения; * принят на складе в городе назначения; * отправлен в пункт выдачи; * принят в пункте выдачи; * утерян; * заявка на перевозку; * утилизирован; * выдан. |

Описание сущности «Категория\_груза» представлено в таблице 2.4.4.

Таблица 2.4.4 – Описание данных сущности «Категория\_груза»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Примечание |
| 1 | 2 | 3 | 4 |
| ID\_категории | Числовой | Id категории | Первичный ключ |
| Наименование | Текстовый | Наименование | Максимальный размер 200 |
| Коэфициент\_перевозки | Числовой | Коэффициент перевозки | Больше 0, дробный тип с точностью 2. |
| Комментарии | Текстовый | Комментарии | Максимальный размер 500, необязательный |

Описание сущности «Машина» представлено в таблице 2.4.5.

Таблица 2.4.5 – Описание данных сущности «Машина»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Примечание |
| 1 | 2 | 3 | 4 |
| ID\_машины | Числовой | Id машины | Первичный ключ |
| Идентификационный\_номер | Текстовый | Идентификационный номер машины (VIN) | Уникальное значение, фиксированный размер 17 символов |
| Гос\_номер | Текстовый | Гос. номер машины | Состоит из 8 или 9 символов |
| Марка | Текстовый | Марка машины | Максимальный размер 200 |
| Модель | Текстовый | Модель машины | Максимальный размер 200 |
| Статус | Текстовый | Статус | Может иметь следующие значения:   * на базе; * назначен водитель; * ремонт; * списана. |

Описание сущности «Пользователь» представлено в таблице 2.4.6

Таблица 2.4.6 – Описание данных сущности «Пользователь»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Примечание |
| 1 | 2 | 3 | 4 |
| ID\_пользователя | Числовой | Id пользователя | Первичный ключ |
| Логин | Текстовый | Логин  Пользователя | Максимальный размер 150 |
| Пароль | Текстовый | Пароль для входа в систему | Максимальный размер 128 |
| Дата\_рождения | Дата | Дата рождения пользователя |  |
| E-mail | Текстовый | E-mail | Максимальный размер 254, уникальное значение |
| Имя | Текстовый | Имя | Максимальный размер 50 |
| Фамилия | Текстовый | Фамилия | Максимальный размер 50 |
| Отчество | Текстовый | Отчество | Необязательный, Максимальный размер 50 |
| Серия\_и\_номер\_ паспорта | Текстовый | Серия и номер паспорта | Уникальное значение, фиксированный размер 10 |
| Телефон | Текстовый | Номер телефона | Максимальный размер 11 |
| Роль | Текстовый | Роль Пользователя | Может иметь следующие значения:   * клиент; * сотрудник; * водитель; * администратор. |

## 2.5 Логическая структура базы данных

Логическая структура базы данных — это схема, которая демонстрирует логические ограничения, которые распространяются на хранимые данные. [В ней отражаются, представления](https://www.lucidchart.com/pages/ru/%D1%81%D1%85%D0%B5%D0%BC%D1%8B%D0%B1%D0%B0%D0%B7-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и условия целостности .

[Под логической структурой понимается совокупность сущностей, содержащихся в них полей и связей между этими сущностями, с которой взаимодействует пользовательская программа, обрабатывающая базу данных](https://it.wikireading.ru/41761).

Логическая структура базы данных логистической компании изображена на рисунке 2.5.1.

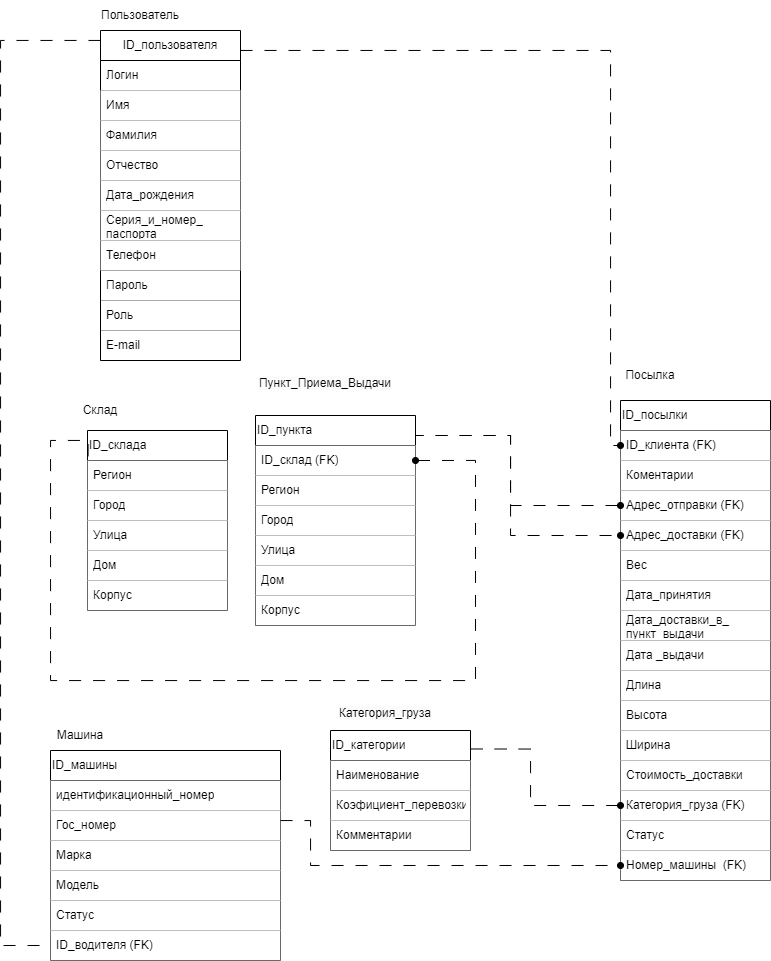


Рисунок 2.3.1 – логическая структура базы данных

Рисунок 2.3.1 – логическая структура базы данных

Сущность «Пункт\_Приема\_Выдачи» содержит информацию о пунктах, где клиенты могут получить или отдать посылку на перевозку.

Сущность «Склад» содержит информацию о складах, на которых хранятся посылки для дальнейшей перевозки.

Сущность «Посылка» содержит данные посылок, которые перевозит компания.

Сущность «Категория\_груза» содержит категории грузов, которые может перевозить компания

Сущность «Машина» хранит информацию о машинах логистической компании.

Сущность «Пользователь» содержит информацию о пользователях сайта, клиентах и сотрудниках.

Связь между сущностями «Пользователь» и «Посылка» выбрана один ко многим, так как один клиент может отправлять несколько посылок.

Связь между сущностями «Машина» и «Посылка» выбрана один ко многим, так как на одной машине может перевозится множество посылок, связь, не идентифицирующая так как у посылка может находиться не машине, а на складе.

Связь между сущностями «Пункт\_Приема\_Выдачи» и «Посылка» выбрана один ко многим, из-за того, что на в одном пункте храниться множество посылок.

Связь между сущностями «Машина» и «Пользователь» выбрана один к одному, так как за одной машиной может быть закреплен только один сотрудник, связь, не идентифицирующая, так как за машиной не обязательно закреплен сотрудник.

Связь между сущностями «Склад» и «Пункт\_Приема\_Выдачи» выбрана один ко многим, так как на один склад привозятся посылки с нескольких пунктов выдачи.

Связь между сущностями «Пользователь» и «Посылка» выбрана один ко многим, так как сотрудник может принять множество посылок.

Связь между сущностями «Категория\_груза» и «Посылка» выбрана один ко многим, так как разные посылки могут иметь одну и туже категорию.

3 Технический проект

3.1 Выбор состава технических и программных средств

Для разработки программного обеспечения и базы данных для логистической компании были выбраны: среда разработки Pycharm, в которой был использован язык программирования Python, фреймворк Django, система управления реляционными базами данных SQLite, также HTML и CSS для размещения на веб-странице различных элементов.

Django – это фреймворк для веб-разработки, который предоставляет инструменты и структуру для создания веб-приложений на языке программирования Python. Django был разработан с целью упростить и ускорить процесс создания веб-приложений, предоставляя готовые решения для многих распространенных задач, связанных с веб-разработкой.

Преимущества Django:

* административная панель: Django поставляется с встроенной административной панелью, которую можно легко настроить для управления данными и администрирования приложения без необходимости создавать пользовательский интерфейс;
* шаблоны и статика: Django позволяет организовать структуру шаблонов и обработку статических файлов, что упрощает разработку пользовательского интерфейса.;
* безопасность: Django включает в себя множество встроенных механизмов безопасности, таких как защита от атак инъекции SQL и других типов атак.

PyCharm – это интегрированная среда разработки (IDE) для языка программирования Python, разработанная компанией JetBrains. Она предоставляет разработчикам множество инструментов и функций, которые упрощают и улучшают процесс создания и поддержки приложений на Python. PyCharm имеет следующие преимущества:

* интеграция с Git: PyCharm интегрирован с системой контроля версий Git, что позволяет управлять исходным кодом и коммитить изменения непосредственно из IDE;
* мощный редактор кода: PyCharm предоставляет богатый редактор кода с автодополнением, подсветкой синтаксиса, проверкой ошибок и другими инструментами, которые упрощают написание и редактирование Python-кода.

Python – это высокоуровневый интерпретируемый язык программирования, который был создан в конце 1980-х годов и развивается до сих пор. Python отличается чистым и читаемым синтаксисом, что делает его одним из наиболее популярных языков программирования в мире. Он имеет следующие преимущества:

* простота и читаемость: Синтаксис Python разработан так, чтобы быть легко читаемым и понятным. Это делает язык доступным для начинающих разработчиков и упрощает поддержку и сопровождение кода.
* кроссплатформенность: Python поддерживается на различных операционных системах, включая Windows, macOS и различные дистрибутивы Linux, что делает его переносимым между платформами;
* динамическая типизация: Python предоставляет динамическую типизацию, что упрощает разработку и уменьшает количество кода, необходимого для решения задач.

SQLite – это компактная и встроенная система управления реляционными базами данных, которая позволяет хранить, управлять и извлекать данные в локальных или встроенных в приложение базах данных. SQLite имеет следующие преимущества: простота в использовании и компактность.

HTML – это язык разметки, используемый для создания структуры и содержания веб-страниц. С помощью HTML разработчики определяют, какие элементы должны присутствовать на веб-странице и как они должны взаимодействовать друг с другом.

CSS – это язык стилей, который используется для определения внешнего вида и макета веб-страниц. CSS позволяет разработчикам задавать цвета, шрифты, размеры, отступы, расположение элементов и другие стилевые атрибуты, которые делают веб-страницу более привлекательной и удобной для пользователей.

3.2 Физическая структура базы данных

Физическая структура базы данных включает в себя описание файлов и групп файлов данных, журнала транзакций, первоначальный размер файлов, шаг прироста базы данных, ее максимальный размер, параметры конфигурации и другие физические характеристики.

Физическая структура базы данных логистической компании изображена на рисунке 3.2.1.

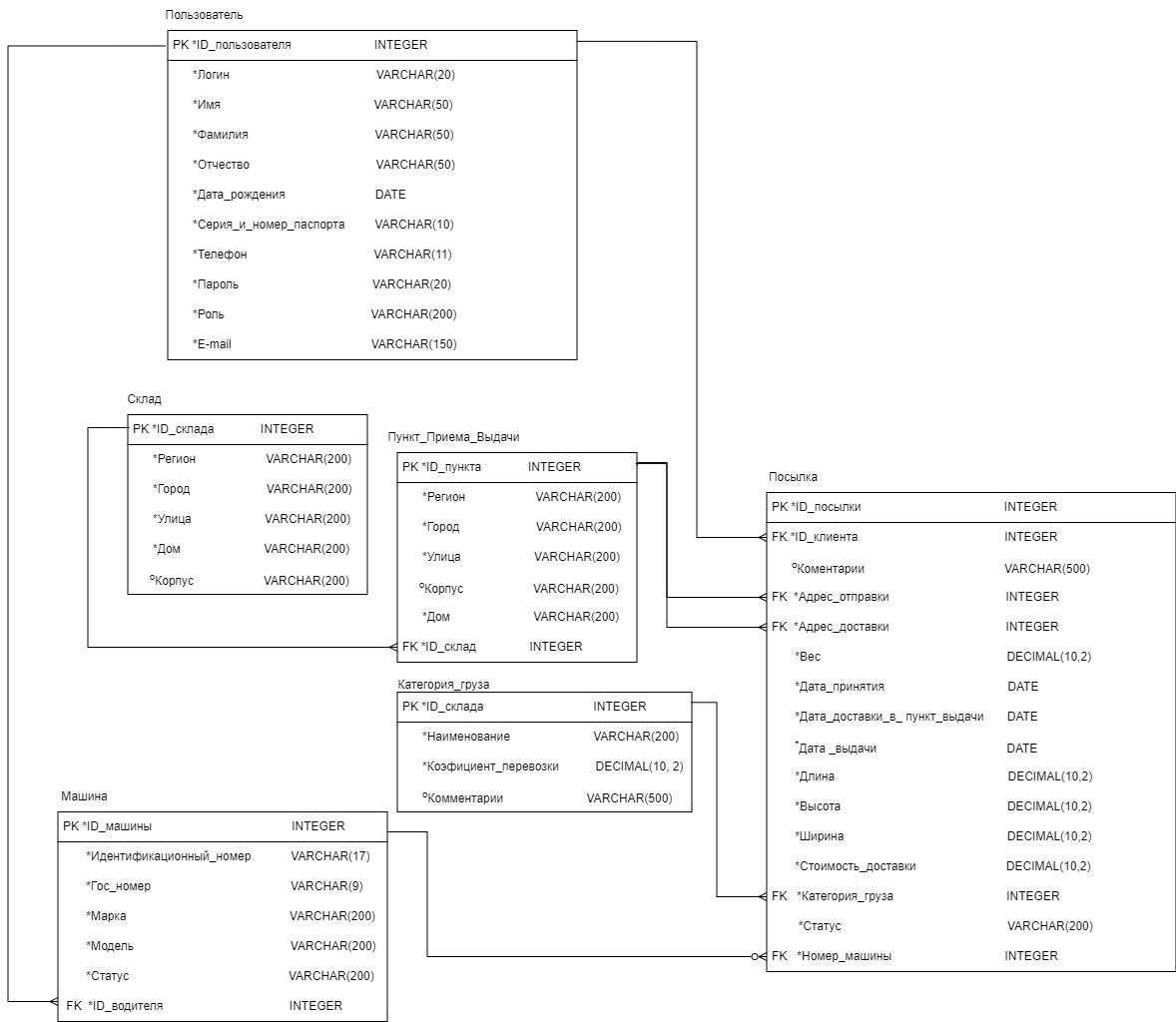


Рисунок 3.2.1 Физическая структура базы данных

В PostgreSQL, отсутствую связи «один к одному» между таблицами. Поэтому между таблицами «Employee» и «Car» используется связь «один ко многим».

Связь между таблицами «Employee» и «Driveridentification» настроена с использованием опции «ON DELETE CASCADE». Таким образом, при удалении сотрудника, удалиться водительское удостоверение. Это обеспечивает целостность данных.

Ключевое слово «UNIQUE» используется в таблицах Employee, Client, DriverIdentification, Car для определения уникальности значений в полях: Email, SeriaNumberPassport, DriverLicense, VIN. Оно гарантирует, что каждое значение в указанных столбцах будет уникальным, то есть не будет дублироваться в пределах таблицы.

Для всех полей, кроме Patronymic, CarID, DriverID в таблицах Client, Employee, Package, Car указывается ключевое слово «NOT NULL» для определения ограничения на поля, которое требует, чтобы в этом поле всегда присутствовало непустое значение. Определение поля с ограничением «NOT NULL» гарантирует, что каждая запись в таблице будет содержать непустое значение в указанном поле. Это используется для обеспечения целостности данных.

Примеры запросов в СУБД PostgreSQL:

* SELECT "DriverIdentification", "B", "BE", "C", "CE", "DateReceipt", "TerminationDate", (SELECT "EmployeeID" FROM "Employee" WHERE "Employee"."EmployeeID" = "DriverIdentification"."EmployeeID") AS "EmployeeID" FROM "DriverIdentification" WHERE "B" IS TRUE;

Данный запрос выдает записи и таблицы DriverIdentification где есть категория «B». Выборка осуществляется с помощью условного оператора «WHERE»;

* SELECT "Name", "Surname", "Patronymic", "Birthdate", "SeriaNumberPassport", "PhoneNumber", "Email" FROM "Client" WHERE "Name" LIKE ('А%');

Данный запрос выдает записи из таблицы Client, где имя начинается на букву «А». Выборка осуществляется с помощью условного оператора «WHERE» и «LIKE»;

* SELECT "Address", "NumberSeats", "Region", "Director" FROM "LocationBase" ORDER BY "NumberSeats";

Данный запрос выдает записи из таблицы LocationBase в порядке возрастания количества мест для автомобилей. Выборка осуществляется с помощью условного оператора «ORDER BY».

* SELECT W."Address", W."Region", E."Surname" FROM "Warehouse" AS W INNER JOIN "Employee" AS E ON W."Director" = E."EmployeeID";

Данный запрос выводит склады с директорами. Выборка осуществляется с помощью оператора объединения «INNER JOIN».

Заключение

В результате разработки базы данных и программного обеспечения для логистической компании «Быстрее света» была достигнута поставленная цель создание базы данных для логистической компании «Быстрее света», которая позволяет вести учёт:

* посылок;
* статусов посылок;
* пунктов выдачи;
* складов;
* машин.

Формировать отчеты и статистику:

* клиентов;
* машин;
* сотрудников;

А также для исключения ошибок, допускаемых при ручной обработке

данных.

В процессе разработки были решены поставленные задачи, включающие: анализ предметной области, определение структуры базы данных, выбор технических и программных средств, а также разработку приложения. Логическая структура базы данных была описана, в которой были учтены бизнес-процессы компании «Быстрее света», а также определены роли пользователей с соответствующими правами доступа.

Преимущества использования разработанной системы включают: обеспечение безопасности базы данных путем разграничения прав доступа, возможность обработки больших объемов данных и обеспечение целостности и непротиворечивости информации. Приложение позволяет быстро находить, изменять и добавлять информацию, формировать отчеты, что значительно упрощает учет и использование данных, повышает производительность работы предприятия и способствует его эффективному функционированию.

Таким образом, разработка базы данных и программного обеспечения для логистической компании «Быстрее света» достигается значительный прогресс в автоматизации и оптимизации ее деятельности, обеспечивая эффективное управление данными, повышение производительности и обеспечение безопасности информации.

Список использованных источников

1. Никитенкова, С. П. Разработка WPF-приложений на основе баз данных : учебно-методическое пособие / С. П. Никитенкова. — Нижний Новгород : ННГУ им. Н. И. Лобачевского, 2019. — 46 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/144994>;

2. Карпова Т.С. Базы данных: модели, разработка, реализация. [Электронный ресурс]. – Режим доступа: [http://www.intuit.ru/department/database](http://www.intuit.ru/department/database/dbmdi/);

3. Кузнецов С.Д. Введение в реляционные базы данных [Электронный ресурс]. – Режим доступа: <https://intuit.ru/studies/courses/74/74/lecture/27919>;

4. Прокушев, Я. Е. Базы данных : учебное пособие / Я. Е. Прокушев. — 2-е изд., доп. — Санкт-Петербург : Интермедия, 2022. — 264 с. — ISBN 978-5-4383-0250-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/217925>;

5. Кривоносова, Н. В. Технология WPF. Разработка модулей программного обеспечения: практикум : учебное пособие / Н. В. Кривоносова. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2021. — 132 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/279719>;

6. Ильин, И. В. Базы данных : учебное пособие / И. В. Ильин, О. Ю. Ильяшенко. — Санкт-Петербург : СПбГПУ, 2020. — 96 с. — ISBN 978-5-7422-7101-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/192881>;

7. Грошев А.С. Основы работы с базами данных [Электронный ресурс]. – Режим доступа: https://studfile.net/preview/2983990/page:13.