

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ  
И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Техническое Задание  
“Курьерская служба”

Выполнили: студенты гр. 953501

Чижик Д.А.  
Дубровский К.Л.

Проверила:  
Тушинская Е.В.  
Гриценко Н.Ю.

Минск 2022

## Содержание

1. Предназначения и формулировка целей .....	3
1.1 Целевая аудитория.....	3
1.2 Цели документа.....	3
1.3 Риски .....	3
1.4 Анализ существующих аналогов .....	3
2. Терминология .....	5
3. Функциональные требования .....	6
3.1 Общие требования .....	6
3.2 Функциональная карта .....	8
3.3 Варианты использования программного средства.....	9
3.4 База данных .....	10
3.5 Data Flow diagram .....	12
3.6 Интерфейс .....	12
4. Нефункциональные требования .....	17
4.1 Системные характеристики: .....	17
4.2 Атрибуты качества .....	17
4.3 Детальные спецификации.....	17
5. Выбор инструментов разработки .....	18
5.1 .NET.....	18
5.2 ASP.NET .....	20
5.3 MS SQL Server .....	21
5.4 MS Visual Studio.....	23
5.5 MS SQL Server Management Studio .....	24
6. План.....	25

# **1. Предназначения и формулировка целей**

## **1.1 Целевая аудитория**

Целевой аудиторией являются:

- Пользователи, которым необходима быстрая доставка интересующего их предмета из точки А в точку Б (до четырех последовательных адресов);
- Пользователи, планирующие переезд в новую квартиру/дом в пределах страны, планирующие ремонт, и им необходима доставка стройматериалов;
- Пользователи, которые приобрели крупный товар в интернет-магазине, и им нужна доставка товара из пункта выдачи.

## **1.2 Цели документа**

Целями данного документа являются:

- Установка рабочего плана
- Распределение ролей и задач в команде
- Постановка задач

## **1.3 Риски**

Создание данного продукта может быть связано со следующими рисками:

- Риск произвести неправильную оценку сложности проекта и не уложиться в поставленные сроки;
- Риск наличия неквалифицированных сотрудников;
- Риск не учесть некоторые аспекты предметной области;
- Риск создать не конкурентоспособный продукт.

## **1.4 Анализ существующих аналогов**

В качестве аналогичных программных продуктов рассматривались другие курьерские службы доставки, такие как:

- [www.dostavka24.by](http://www.dostavka24.by);
- [www.cdek.by](http://www.cdek.by);
- [www.deliver.by](http://www.deliver.by).

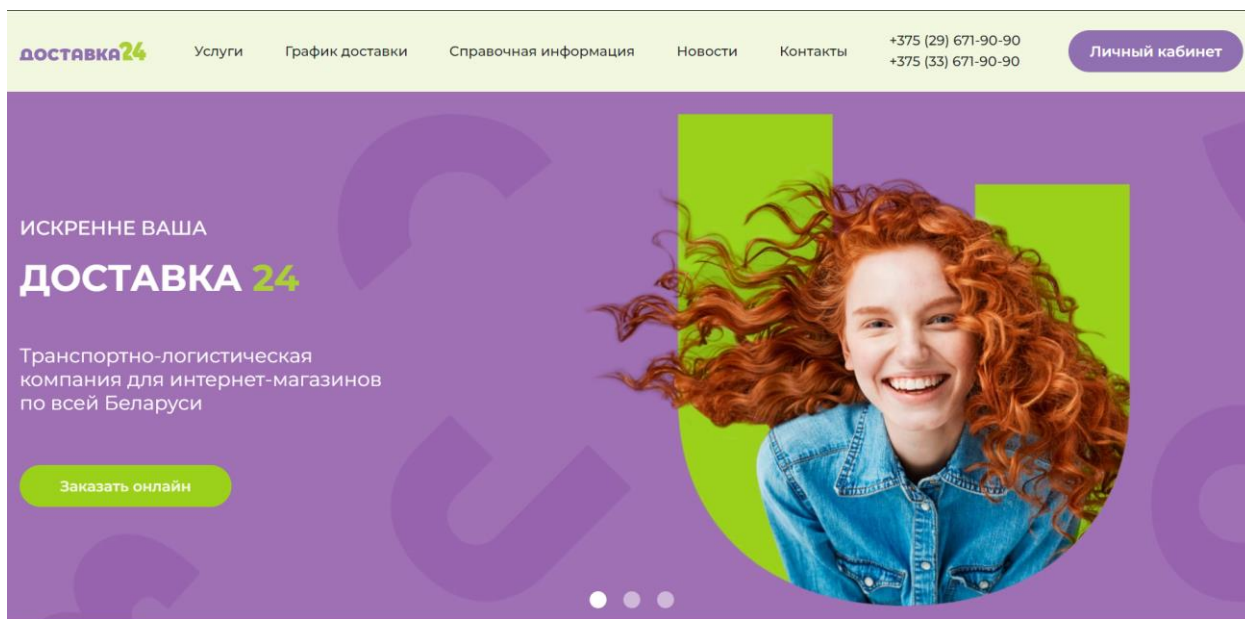


Рисунок 1.1 – Сайт курьерской службы dostavka24.by

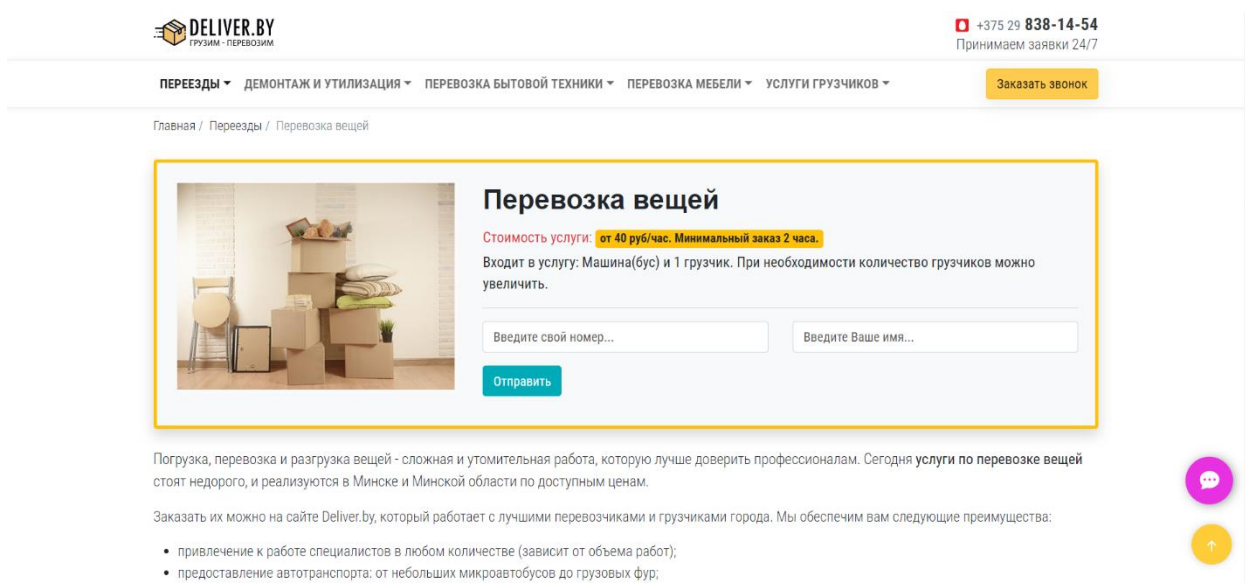


Рисунок 1.2 – Сайт курьерской службы deliver.by

## 2. Терминология

- **Клиент** – это потенциальный клиент курьерской службы, использующий сервис в личных целях;
- **Заказ** – сформированная клиентом заявка, содержащая всю необходимую информацию для доставки; доставляемый курьерской службой груз клиента от точки А в точку Б;
- **Администратор** – привилегированный пользователь, обладающий расширенными правами для управления всеми поступающими с сайта заказами и распределения их между водителями грузовых автомобилей, а также решения проблемных ситуаций;
- **Водитель** – работник курьерской службы, осуществляющий доставку заказа клиента;
- **Грузчик** - работник курьерской службы, сопровождающий водителя для осуществления погрузки-разгрузки заказа;
- **Транзакция** – операция, состоящая в переводе денежных средств с одного счёта на другой, сделка купли-продажи;
- **Денежный перевод** – перемещение денег от одного лица к другому через платёжную систему без непосредственной передачи наличных денежных знаков из рук в руки;
- **Учетная запись** – хранимая в компьютерной системе совокупность данных о пользователе, необходимая для его опознания (аутентификации) и предоставления доступа к его личным данным и настройкам;
- **Аутентификация** – процедура проверки подлинности;
- **Личный кабинет** – особый раздел приложения водителя, который позволяет пользователю получить доступ к данным о состоянии и статистической информации заказов.

### **3. Функциональные требования**

#### **3.1 Общие требования**

Сервис включает в себя:

- А.** Сайт для формирования заказа клиентом;
- В.** Приложение, реализующее необходимый функционал для работы администратора;
- С.** Приложение, реализующее необходимый функционал для работы водителя.

**ОТ А-1:** Стартовая страница сайта должна содержать следующую информацию:

- информацию об услугах курьерской службы доставки
- информацию о тарифах и способе оплаты заказа
- информацию о связи с администратором

**ОТ А-2:** Пользователь должен иметь возможность сформировать заказ на доставку при помощи формы. Форма регистрации заказа пользователя-клиента должна содержать следующие поля:

- имя клиента
- номер телефона клиента
- пункты А и Б доставки груза
- информацию о грузе

Поля для заполнения информации о грузе могут быть дополнительно добавлены в связку «пункт А + пункт Б» маршрута, если клиент планирует перевести груз, состоящий из нескольких позиций.

Пользователь может добавить новую связку «пункт Б + пункт В» маршрута (до 4-х пунктов), если пользователю необходима доставка груза в разные места.

**ОТ А-3:** После формирования заказа пользователь должен получить информацию о стоимости заказа и выбрать способ оплаты. В случае оплаты онлайн, пользователь перенаправляется на страницу с оплатой.

**ОТ В-1:** Администратор должен иметь возможность просматривать, редактировать, удалять записи о заказах, водителях, грузчиках, тарифах и автомобилях в таблицах.

**ОТ В-2:** Администратор должен иметь возможность добавлять новые записи о водителях, грузчиках, автомобилях заполнив соответствующие формы.

**ОТ В-3:** Администратор должен иметь возможность просматривать записи в таблицах архива, но не должен иметь возможности их изменять.

**ОТ В-4:** Администратор должен иметь возможность закрепить заказ за водителем, изменив соответствующее поле записи в таблице заказа.

**ОТ С-1:** Водитель должен иметь установленным собственное приложение для водителей, вход в которое должен осуществляться по аутентификационным данным аккаунта, предоставленного ему администратором.

**ОТ С-2:** Водитель должен иметь возможность просматривать записи о заказах, которые ему назначил администратор.

**ОТ С-3:** Водитель должен иметь возможность выбрать на закрепленный за ним заказ грузчика и автомобиль от компании.

**ОТ С-4:** Водитель должен иметь возможность просмотреть указанный в заказе маршрут по интернет-карте.

### 3.2 Функциональная карта

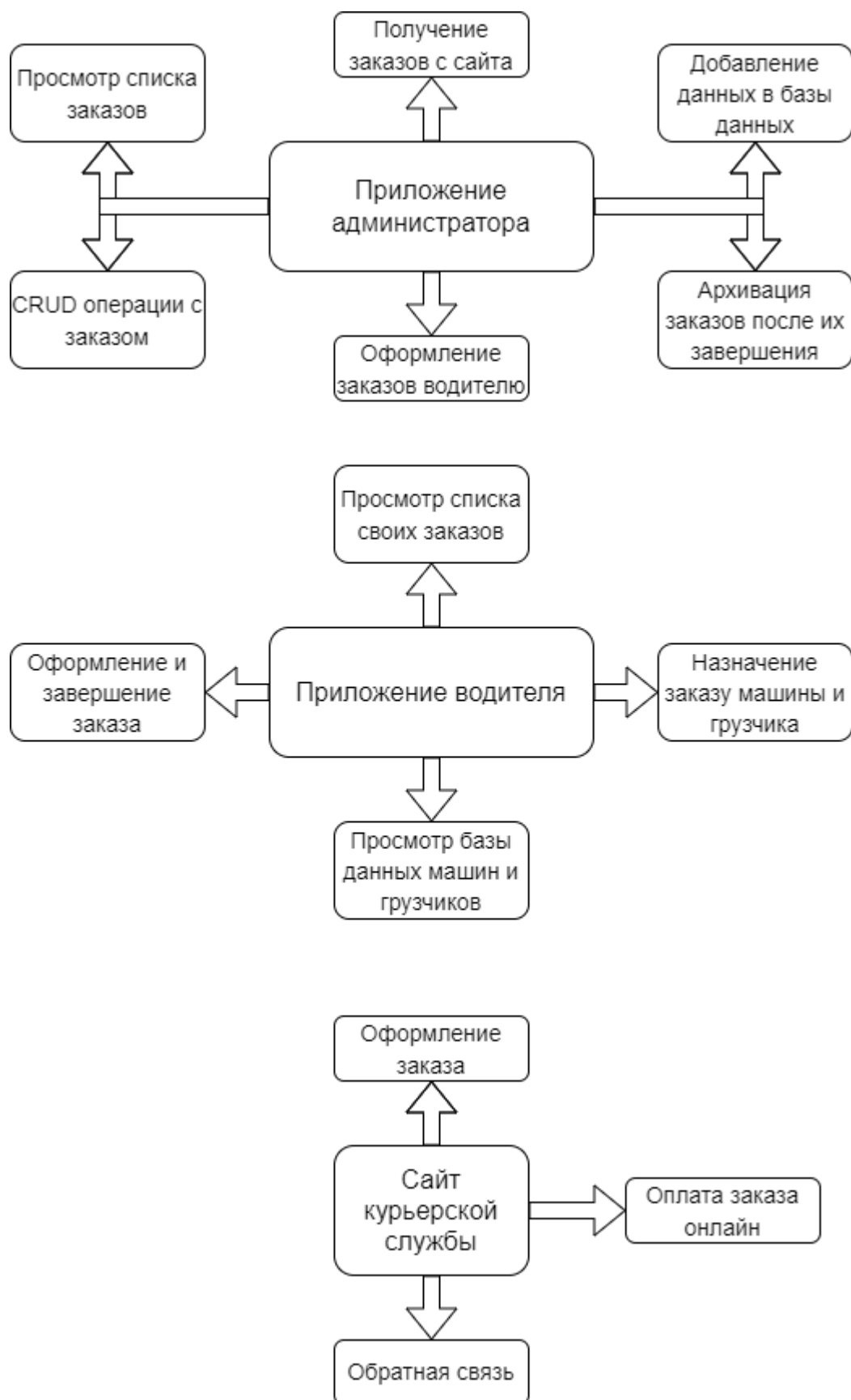


Рисунок 3.1 – Функциональная карта ПС



### 3.3 Варианты использования программного средства

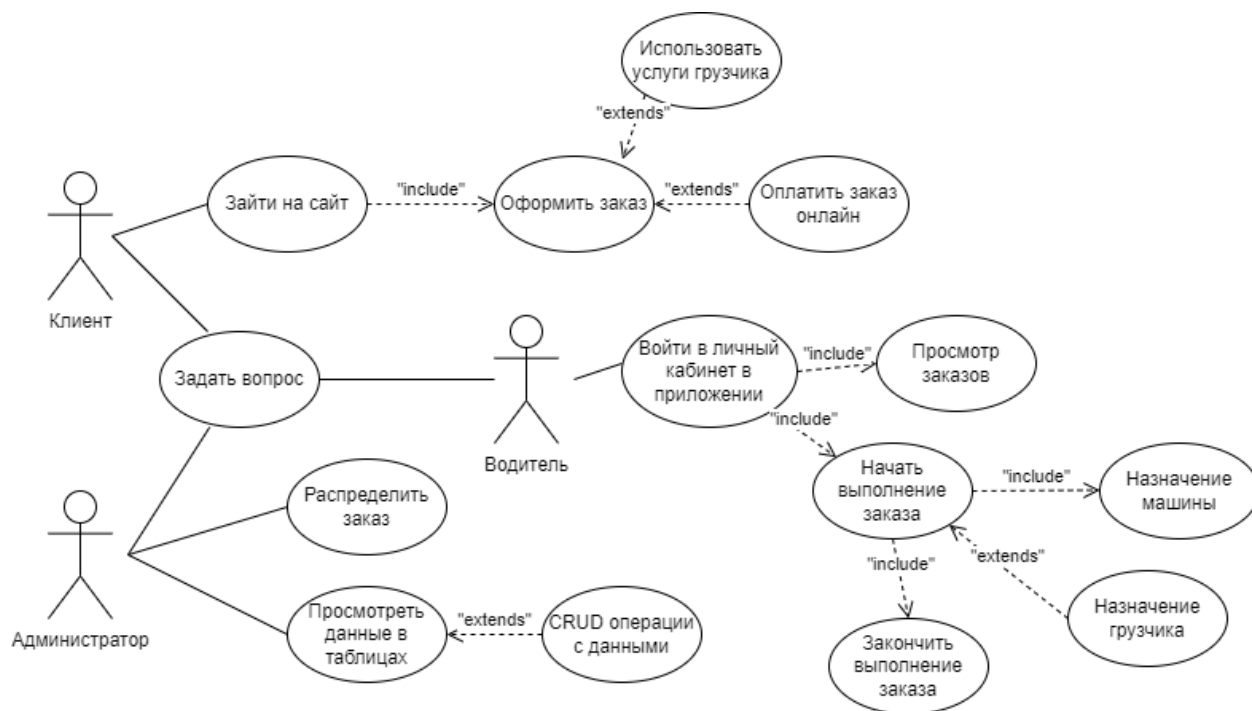


Рисунок 3.2 – Диаграмма вариантов использования ПК (Use case)

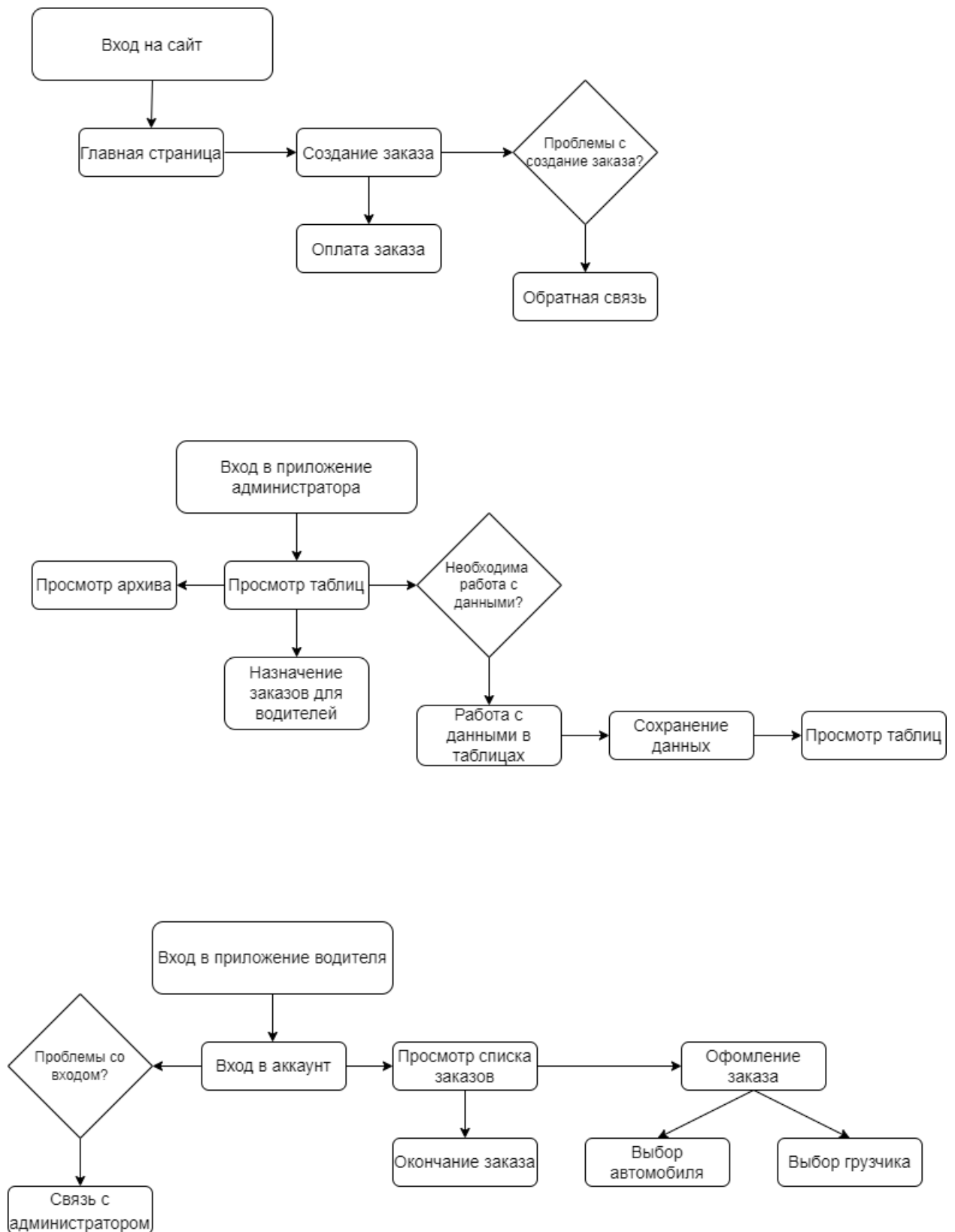


Рисунок 3.3 – Диаграмма вариантов использования ПК (User flow)

### 3.4 База данных

Для работы сервиса необходима спроектировать базу данных.

База данных будет храниться на интернет-сервере, к ней будут иметь доступ сайт (добавление записей о заказе), приложение администратора

(просмотр, добавление, удаление, редактирование всех записей), приложение водителя (просмотр записей о заказе, добавление записей о водителе). Модель базы данных представлена на рисунке 3.3.

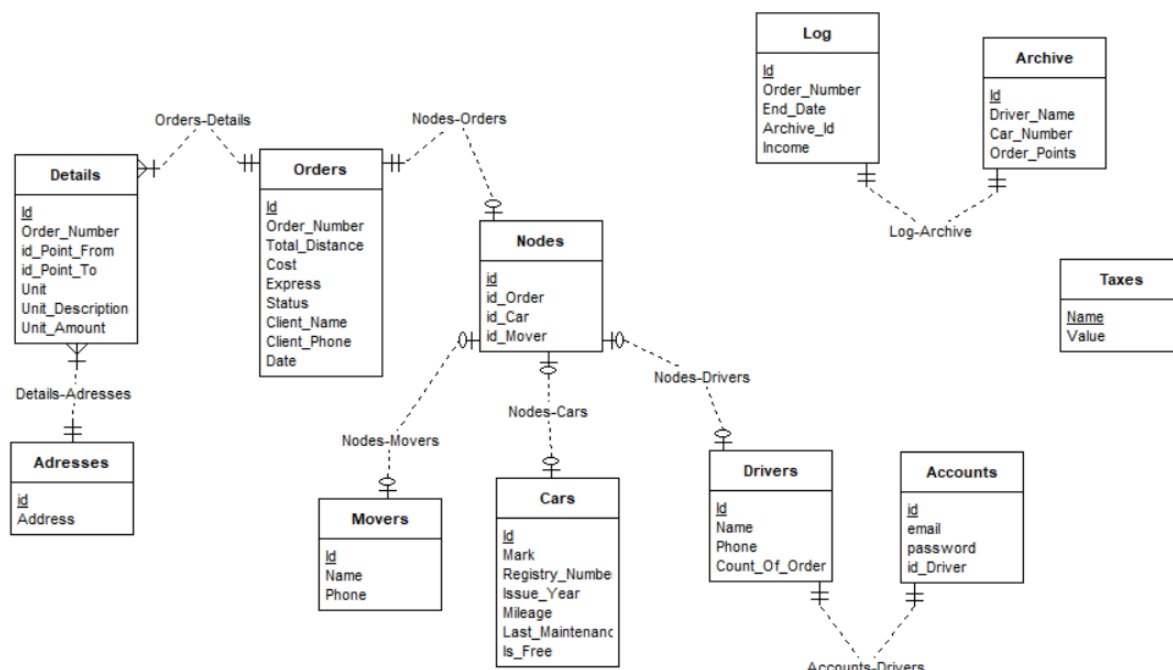


Рисунок 3.4 – ER диаграмма базы данных

База данных состоит из 11 таблиц. Таблицы Drivers и Accounts хранят информацию о водителях. Таблицы Movers и Cars хранят данные о грузчиках и автомобилях компании соответственно. За добавление и удаление записей в этих таблицах отвечает администратор.

Таблицы Orders, Details Addresses содержит записи о заказе, причём одной записи в таблице Orders может соответствовать несколько записей из Details. Также адресам (записям) из таблицы Addresses могут соответствовать несколько записей из Details. В этих таблицах записи добавляются после ввода клиентом данных на сайте. Записи в таблице Nodes формируются, когда администратор соотносит заказ водителю, автомобилю и грузчику.

После того, как заказ выполнен, данные о нем удаляются из предыдущих таблиц и попадают в таблицы Archive и Log, которые представляют собой архив записей заказов.

Таблица Taxes содержит некоторые значения, которые могут быть полезны при добавлении новых заказов и могут изменяться (например, зависимость стоимости доставки от расстояния, максимальное расстояние, минимальная стоимость и т.д.).

База данных находится в третьей нормальной форме, т.к.:

- все атрибуты простые (1-я НФ),
- каждый не ключевой атрибут неприводимо зависит от первичного ключа, т.е. в составе потенциального ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость (2-я НФ),
- каждый не ключевой атрибут нетранзитивно зависит от первичного ключа, т.е. все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы, вынесены в отдельные таблицы (3-я НФ).

### 3.5 Data Flow diagram

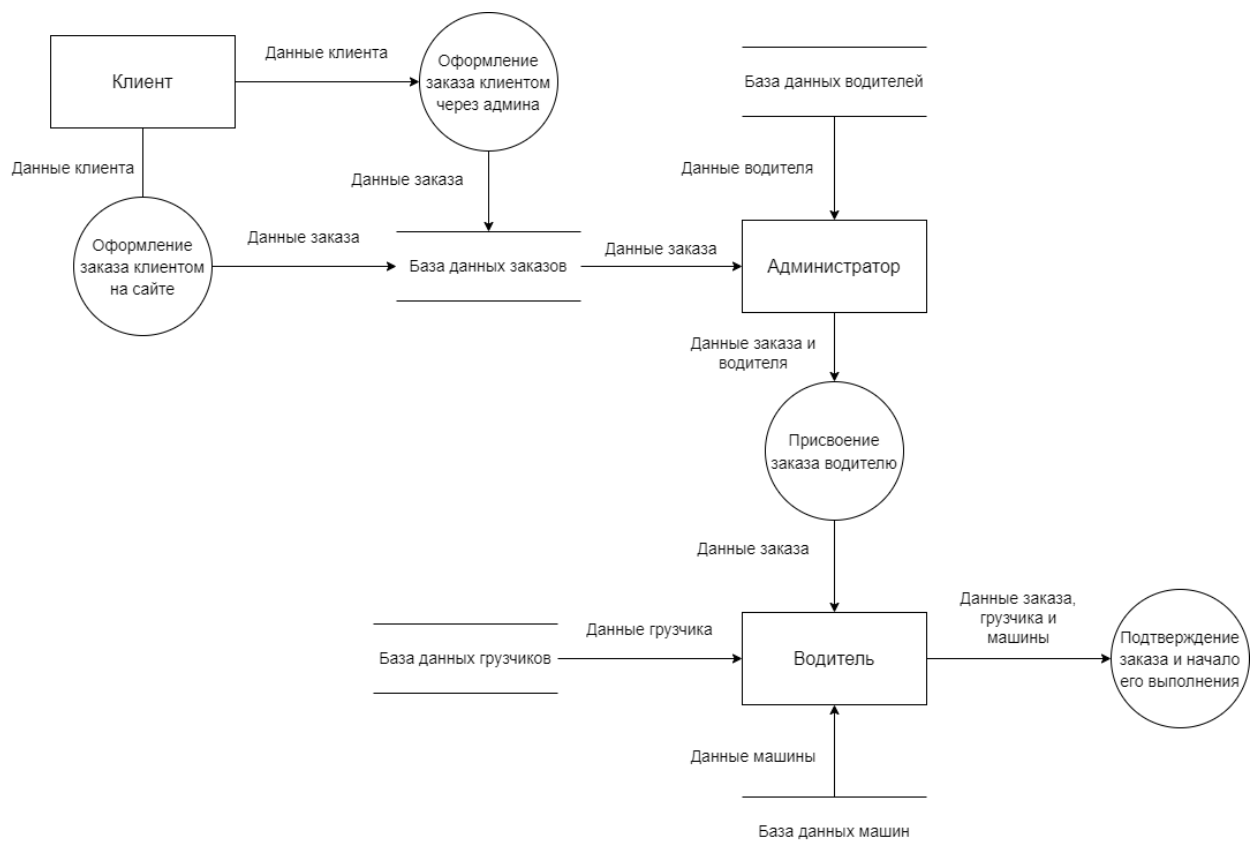


Рисунок 3.5 – Диаграмма потока данных (Data flow)

### 3.6 Интерфейс

Программные средства должны использовать следующий графический интерфейс:

## Client Order Site Page

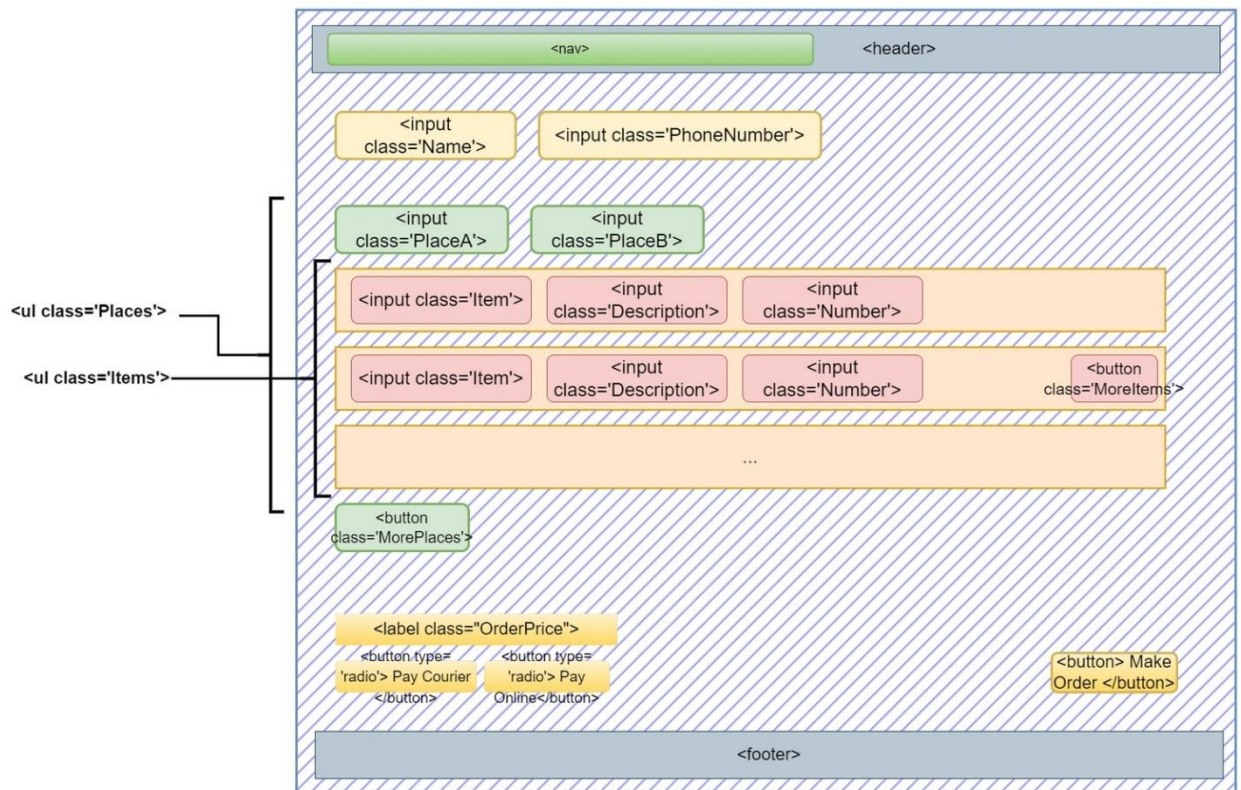


Рисунок 3.6 – Интерфейс сайта заказа (ПК версия)

## Client Order Site Page (mobile)

The diagram illustrates the layout of a mobile client order site page. It features a header with navigation and a footer. The main content area includes input fields for Name, Phone Number, Place A, Place B, Item, Description, and Number. There are also buttons for More Places, More Items, Pay Courier, Pay Online, and Make Order. A label for Order Price is also present.

<nav> <header>

<input class='Name'> <input class='PhoneNumber'>

<input class='PlaceA'>

<input class='PlaceB'>

<input class='Item'>

<input class='Description'>

<input class='Number'> <button class='MoreItems'>

<button class='MorePlaces'>

<label class="OrderPrice">

<button type='radio'> Pay Courier </button>

<button type='radio'> Pay Online</button>

<button> Make Order </button>

<footer>

Рисунок 3.7 – Интерфейс сайта заказа (мобильная версия)



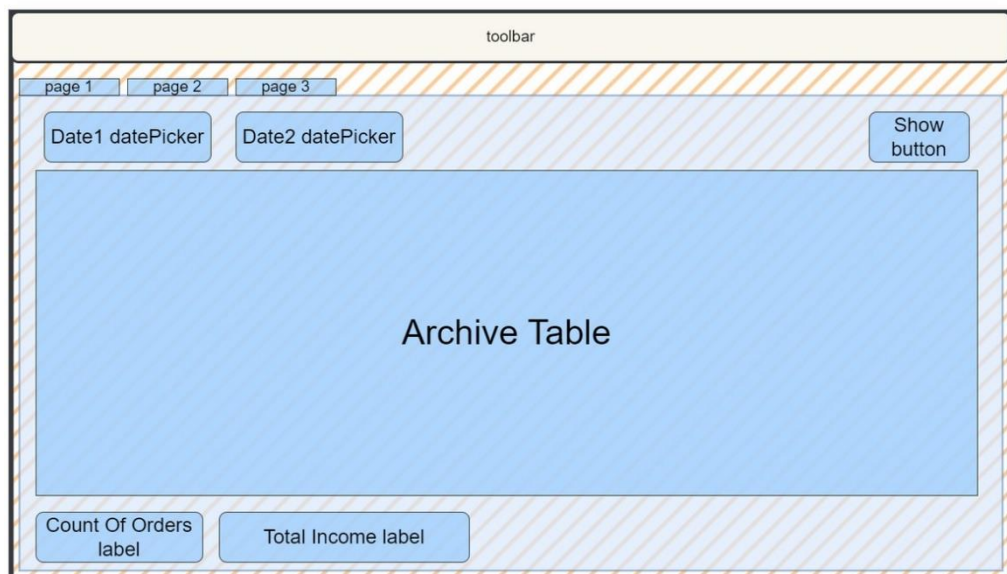
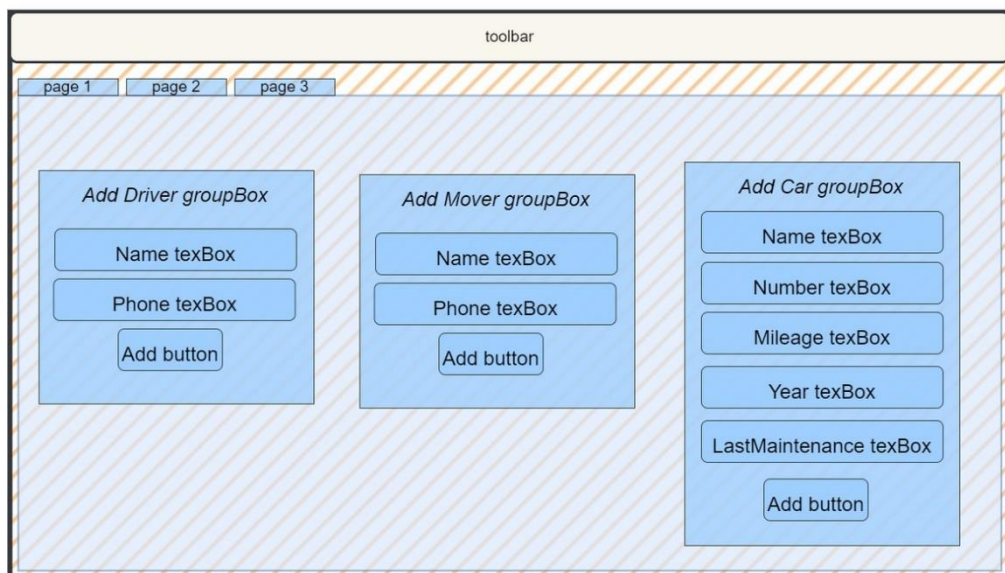
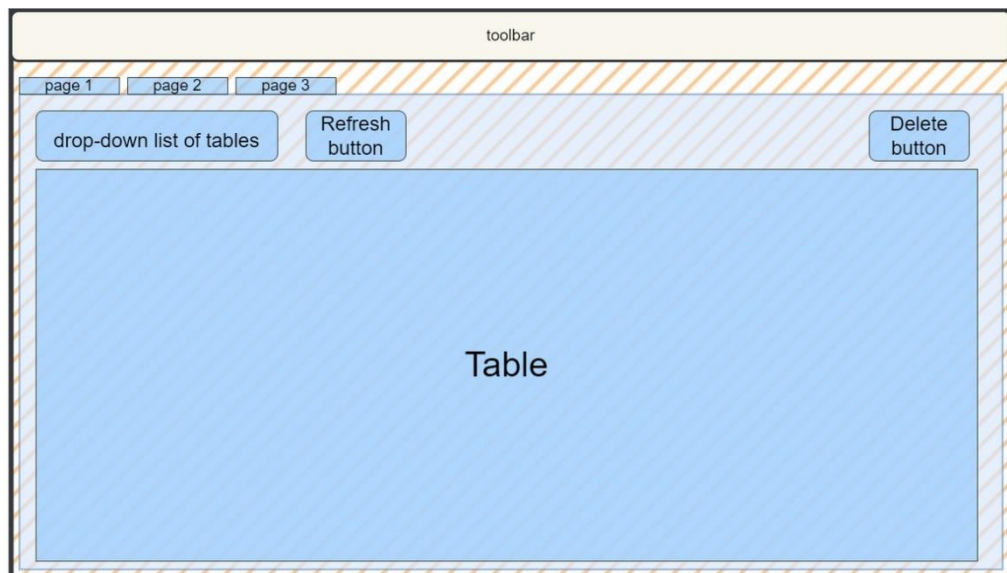


Рисунок 3.8 – Интерфейс приложения администратора (3 страницы)

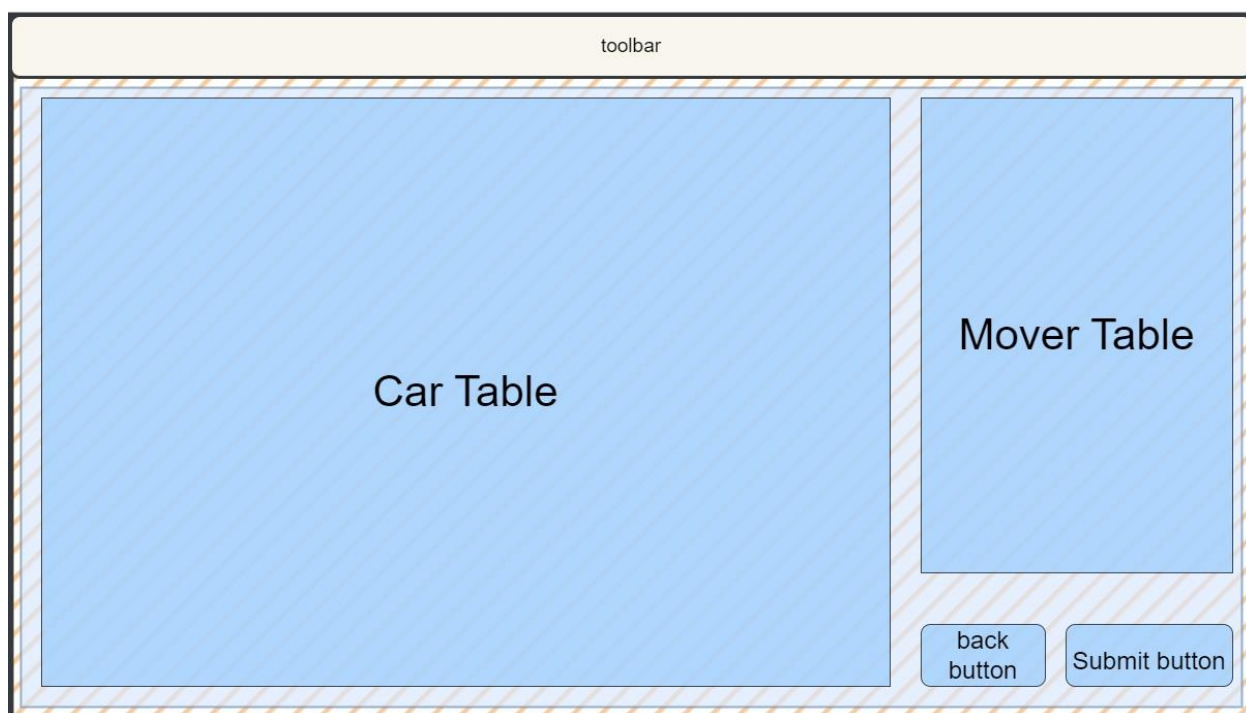


Рисунок 3.9 – Интерфейс приложения водителя (2 страницы)



## **4. Нефункциональные требования**

### **4.1 Системные характеристики:**

- CX-1: Сервис должен использовать базу данных.
- CX-2: При написании приложений должны быть использованы технологии .NET Core, .NET Framework, ASP.NET и SQL Server.
- CX-3: Рекомендуемой системой для работы сервиса является ОС Windows.

### **4.2 Атрибуты качества**

- АК-1: Производительность
  - АК-1.1: Приложение должно обеспечивать скорость обработки данных 5 МБ/сек со следующим (или эквивалентным оборудованием): CPU i7, RAM 4 GB, средняя скорость чтения/записи диска 30 МБ/сек.
- АК-2: Устойчивость к входным данным
  - АК-2.1: Приложение должно обрабатывать вводимые данные на русском и английском языках.
  - АК-2.2: Приложение должно корректно вести себя при неправильно введенных входных данных.

### **4.3 Детальные спецификации**

- ДС-1: Журнал ошибок
  - ДС-1.1: Формат журнала ошибок в лог-файл: YYYY-MM-DD HH:MM:SS код\_ошибки параметры\_ошибки стек\_вызова.
  - ДС-1.2: В случае если лог-файл отсутствует, должен быть создан новый пустой лог-файл.
  - ДС-1.3: В случае если лог-файл уже существует, должно происходить добавление новых записей в его конец.

## 5. Выбор инструментов разработки

На основании выдвинутых к программному средству основных функциональных и нефункциональных требований, обзора существующих аналогов, было принято решение о проектировании сети приложений, использующих одну базу данных в сети Интернет, разделенных функциональными требованиями на три части: веб-сайт для клиента, приложение для администратора и приложение для водителя.

### 5.1 .NET

Microsoft.NET - программная технология, предназначенная для создания как обычных программ, так и веб-приложений (в качестве платформы для разработок впервые предложена корпорацией Microsoft).

Одной из основных идей Microsoft.NET является совместимость различных служб, написанных на разных языках. Например, служба, написанная на C++ для Microsoft.NET, может обратиться к методу класса из библиотеки, написанной на Delphi; на C# можно написать класс, наследованный от класса, написанного на Visual Basic.NET, а исключение, созданное методом, написанным на C#, может быть перехвачено и обработано в Delphi. Каждая библиотека (сборка) в.NET имеет сведения о своей версии, что позволяет устранить возможные конфликты между разными версиями сборок.

.NET является патентованной технологией корпорации Microsoft. Тем не менее, после заключения договоренности с компанией Novell, была признана технология Mono как реализация.NET на Unix-подобных системах (GNU/Linux, Mac OS X). Однако договоренность касается Novell и клиентов Novell, также технологии ASP.NET, ADO.NET и Windows.Forms не были стандартизированы ECMA/ISO и использование их в Mono находится под угрозой претензий со стороны Microsoft. Mono предоставляет реализацию ASP.NET, ADO.NET и Windows.Forms, но в то же время рекомендует обходить эти API.

#### **Среды разработки.NET-приложений:**

- 1) Microsoft Visual Studio (C#, Visual Basic.NET, Managed C++).
- 2) SharpDevelop.
- 3) MonoDevelop.
- 4) Eclipse.
- 5) Borland Developer Studio (Delphi for.NET, C#).
- 6) PascalABC.NET и т. д.

Приложения также можно разрабатывать в текстовом редакторе и использовать консольный компилятор.

Так же как и технология Java, среда разработки.NET создаёт байт-код, предназначенный для исполнения виртуальной машиной. Входной язык этой

машины в.NET называется MSIL (Microsoft Intermediate Language), или CIL (Common Intermediate Language, более поздний вариант), или просто IL. Применение байт-кода позволяет получить кроссплатформенность на уровне скомпилированного проекта (в терминах.NET: сборка), а не только на уровне исходного текста, как, например, в C. Перед запуском сборки в среде исполнения CLR байт-код преобразуется встроенным в среду JIT-компилятором (just in time, компиляция на лету) в машинные коды целевого процессора. Также существует возможность скомпилировать сборку в родной (native) код для выбранной платформы с помощью поставляемой вместе с .NET Framework утилиты NGen.exe.

### **Преимущества.Net:**

- 1) Полные возможности взаимодействия с существующим кодом. Существующие двоичные компоненты COM отлично работают вместе с двоичными файлами.NET.
- 2) Полное и абсолютное межъязыковое взаимодействие. В отличие от классического COM, в.NET поддерживаются межъязыковое наследование, межъязыковая обработка исключений и межъязыковая отладка.
- 3) Общая среда выполнения для любых приложений.NET, вне зависимости от того, на каких языках они были созданы. Один из важных моментов при этом - то, что для всех языков используется один и тот же набор встроенных типов данных.
- 4) Библиотека базовых классов, которая обеспечивает сокрытие всех сложностей, связанных с непосредственным использованием вызовов API, и предлагает целостную объектную модель для всех языков программирования, поддерживающих.NET.
- 5) Отсутствие сложности, присущей COM. IClassFactory, IUnknown, код IDL и VARIANT-совместимые типы данных (BSTR, SAFEARRAY и остальные) не используются в коде программ.NET.
- 6) Действительное упрощение процесса развертывания приложения. В.NET нет необходимости регистрировать двойные типы в системном реестре. Более того,.NET позволяет разным версиям одного и того же модуля DLL мирно сосуществовать на одном компьютере.

### **Строительные блоки.NET (CLR, CTS и CLS)**

Технологии CLR, CTS и CLS очень важны для понимания смысла платформы.NET. С точки зрения программиста.NET вполне можно рассматривать просто как новую среду выполнения и новую библиотеку базовых классов. Среда выполнения.NET как раз и обеспечивается с помощью Common Language Runtime (CLR, стандартная среда выполнения для языков). Главная роль CLR заключается в том, чтобы обнаруживать и загружать типы.NET и производить управление ими в соответствии с вашими

командами. CLR берет на себя всю низкоуровневую работу - например, автоматическое управление памятью, межъязыковым взаимодействием, развертывание (с отслеживанием версий) различных двоичных библиотек.

Еще один строительный блок платформы.NET - это Common Type System (CTS, стандартная система типов). CTS полностью описывает все типы данных, поддерживаемые средой выполнения, определяет, как одни типы данных могут взаимодействовать с другими и как они будут представлены в формате метаданных.NET.

Важно понимать, что не во всех языках программирования.NET обязательно должны поддерживаться все типы данных, которые определены в CTS. Common Language Specification (CLS) - это набор правил, определяющих подмножество общих типов данных, в отношении которых гарантируется, что они безопасны при использовании во всех языках.NET.

### **Библиотека базовых классов.NET**

Помимо спецификаций CLR и CTS/CLS платформа.NET предоставляет в распоряжение программиста также и библиотеку базовых классов, доступную из любого языка программирования.NET. Библиотека базовых классов не только прячет обычные низкоуровневые операции, такие как файловый ввод-вывод, обработка графики и взаимодействие с оборудованием компьютера, но и обеспечивает поддержку большого количества служб, используемых в современных приложениях.

## **5.2 ASP.NET**

ASP.NET – это новая технология для создания мощных сценариев, которые выполняются на сервере. Она предоставляет разработчикам службы, необходимые для создания .NET-приложений. Компания Microsoft первоначально планировала назвать свой продукт ASP+ -- как усовершенствование ASP. Но после создания платформы .NET ASP+ была переименована в ASP.NET и вошла в состав пакета среды разработки приложений Visual Studio.NET.

Непосредственно взаимодействуя с операционной системой, среда .NET Framework предоставляет интерфейс ASP-приложениям. Новая технология ASP.NET позволяет создавать приложения на нескольких языках программирования, например, на VisualBasic .NET, C# и JScript .NET. Благодаря этому приложениям предоставляются возможности .NET, такие как работа в среде CLR, безопасность типов и наследование. Наиболее важными усовершенствованиями, добавленными в ASP.NET, являются серверные элементы управления (ServerControls), новые возможности работы в Web, кэширование Web-страниц и новая объектная модель.

Технология ASP.NET является новой средой разработки Web-приложений. Технология ASP базировалась на использовании языков

сценариев. В основу ASP.NET положена работа в среде CLR, что позволяет создавать Web-приложения на любом языке, поддерживаемом платформой .NET. Независимо от языка программирования, использованного при создании приложения ASP, его код компилируется в код на промежуточном языке IL. Это немаловажное преимущество, так как теперь возможности одного языка могут использоваться в другом языке без необходимости написания дополнительного кода. Таким образом достигается высокая степень повторного использования кода.

Файлы страниц, создаваемых в рамках технологии ASP.NET, могут иметь различные расширения. Файл стандартной ASP.NET-страницы имеет расширение .aspx. Файл Web-службы имеет расширение .asmx, а файл пользовательского элемента управления -- расширение .ascx. Поддержка различных форматов файлов позволяет одновременно использовать ASP.NET-и ASP-страницы. В зависимости от расширения файла сервер IIS вызывает соответствующий ISAPI-фильтр для управления выполнением задачи. Архитектура ASP.NET позволяет различать управляемый и неуправляемый код. На коде, управляемом средой CLR, написаны .NET-приложения, что позволяет использовать возможности .NET Framework. Например, с помощью функции отсоединенного доступа технология ASP.NET поддерживает работу с серверами IIS 4.0, IIS 5.0 или Internet Explorer 5.5.

ASP.NET поддерживает две модели программирования: Web-формы и Web-службы. Web-формы позволяют создавать Web-страницы с помощью форм. Элементы управления Web-форм доступны на панели инструментов General (Стандартная) и могут использоваться для создания элементов пользовательского интерфейса Web-форм. Элементы управления Web-форм могут использоваться многократно, что значительно упрощает задачу написания кода для Web-страниц.

### **5.3 MS SQL Server**

В качестве СУБД для хранения данных системы была выбрана СУБД MS SQL Server.

SQL Server -- это комплексная платформа управления данными и бизнес-аналитики. Она обладает первоклассной масштабируемостью, возможностью создавать хранилища данных, продвинутыми средствами анализа и достаточной безопасностью, что позволяет использовать ее как основу для критически важных бизнес-приложений. Эта редакция позволяет консолидировать серверы и выполнять крупномасштабные OLTP-операции и создание отчетности.

#### **Достоинства SQL Server:**

- **Высокий уровень доступности.** Технологии, которые защищают данные от дорогостоящих человеческих ошибок и максимально

сокращают сроки аварийного восстановления, помогут обеспечить непрерывность ведения бизнеса.

- **Производительность и масштабируемость.** Инфраструктура, поставившая официальный рекорд в обработке больших объемов данных и пиковых нагрузок.
- **Безопасность.** Встроенные средства защиты от несанкционированного доступа позволяют решить вопрос конфиденциальности и соответствия нормативным требованиям.
- **Управляемость.** Автоматические диагностика, калибровка и настройка инфраструктуры позволяют снизить издержки на управление, сократить потребность в обслуживании и притом управлять огромными объемами данных.
- **Бизнес-аналитика.** Большие объемы данных из хранилищ или киосков легко запрашиваются и анализируются, преобразуясь в практически значимый результат, ведущий к принятию верного решения.

### Преимущества SQL Server:

- ✓ Управление большими и постоянно растущими таблицами станет более эффективным благодаря прозрачному разбиению таблиц на управляемые блоки данных по технологии параллелизма в секционированных таблицах.
- ✓ Улучшения в создании отказоустойчивых кластеров в ОС WindowsServer.
- ✓ Поврежденные страницы данных можно восстановить с зеркального сервера благодаря улучшенному зеркалированию баз данных.
- ✓ Новые узлы в одноранговую репликацию можно добавлять во время работы, не отключая репликацию.
- ✓ Регулятор ресурсов позволяет осуществлять упреждающий контроль приоритетности рабочей нагрузки и выделения ресурсов.
- ✓ Сжатие резервных копий позволяет сократить время, требуемое на восстановление, а также уменьшить количество занимаемого резервными копиями пространства.
- ✓ Возможность горячей замены процессора снижает время простоев из-за обслуживания оборудования.
- ✓ Средство сбора данных о производительности дает возможность осуществлять тонкую настройку экземпляров SQL Server по всему предприятию, а также устранять неполадки и производить мониторинг.
- ✓ Общекорпоративные средства шифрования становятся возможными благодаря расширенному управлению ключами и аппаратным модулям безопасности.
- ✓ Прозрачное шифрование данных позволяет шифровать информацию без внесения каких-либо изменений в приложения.
- ✓ Все предпринятые действия могут подвергаться проверке благодаря новому объекту аудита.

- ✓ Сжатие данных позволяет хранить информацию более эффективно и снизить требования к хранилищу.
- ✓ Использование оптимизации запросов с объединением «звезда» сокращает время отклика благодаря распознаванию схем объединения данных хранилища.
- ✓ В хранилище всегда будет самая актуальная информация за счет отслеживания изменений в данных.
- ✓ Улучшения, внесенные в масштабируемость и производительность служб аналитики SQL Server, позволяют создавать высокопроизводительные решения аналитики под практически любое количество пользователей.
- ✓ Продвинутое ядро интеллектуального анализа данных
- ✓ Перепроектированное ядро служб отчетов SQL Server позволяет осуществлять обработку по запросу и достигать лучшей производительности.
- ✓ Масштабируемые задачи просмотра повышают производительность извлечения, преобразования и загрузки в службах интеграции SQL Server.

## **5.4 MS Visual Studio**

Среда разработки Microsoft Visual Studio – это набор инструментов и средств, предназначенных для помощи разработчикам программ любого уровня квалификации в решении сложных задач и создания новаторских решений. Разработчикам программного обеспечения часто приходится решать ряд проблем, чтобы создавать удачные программы. Роль Visual Studio заключается в том, чтобы улучшить процесс разработки и упростить разработку высокоэффективных программ.

Как в Visual Studio улучшен процесс разработки:

### **1) Производительность**

Средства Visual Studio позволяют разработчикам работать с большей отдачей и затрачивать меньше усилий на повторяющиеся задачи. Следует отметить высокопроизводительные редакторы кода, поддержку технологии IntelliSense, мастеров и различных языков кодирования в одной интегрированной среде разработки (IDE), а также продукты управления жизненным циклом приложений (ALM) в Microsoft Visual Studio Team System. В новых версиях Visual Studio постоянно появляются новые средства, позволяющие разработчикам сосредоточиться на решении основных проблем, а не на рутинной работе.

### **2) Интеграция**

Разработчики, применяющие Visual Studio, получают в свое распоряжение интегрированный продукт, включающий инструменты,

серверы и службы. Продукты Visual Studio отлично работают вместе - не только один с другим, но и с прочими программами Майкрософт, включая серверные продукты и приложения Microsoft Office.

### **3) Комплексность**

В Visual Studio содержатся инструменты для всех этапов разработки программного обеспечения (разработка, тестирование, развертывание, интеграция и управления) и для разработчиков любого уровня квалификации, от новичков до опытных специалистов. Visual Studio поддерживает разработку для различных типов устройств - ПК, серверов, сетевых и мобильных устройств.

### **4) Надежность**

Visual Studio разрабатывается таким образом, чтобы обеспечить высокую надежность и совместимость. Visual Studio обладает удачным сочетанием безопасности, масштабируемости и взаимодействия. В Visual Studio всегда поддерживаются новейшие технологии, но везде, где это возможно, обеспечивается обратная совместимость.

## **5.5 MS SQL Server Management Studio**

Так как имеющаяся база данных была в формате MS SQL Server, поэтому для развертывания использовался продукт Microsoft SQL Server Management Studio. Среда Microsoft SQL Server Management Studio (SSMS) — это интегрированная среда для доступа, настройки, администрирования, разработки всех компонентов SQL Server и управления ими. Служба SSMS сочетает в себе обширную группу графических инструментов с рядом отличных редакторов сценариев для обеспечения доступа к службе SQL Server для разработчиков и администраторов всех профессиональных уровней.



## **6. План**

01.10.2022 – Разработать техническое задание

31.10.2022 – Написать базовый функционал

24.11.2022 – Реализовать все основные требования проекта

10.12.2022 – Тестирование, исправление дефектов

17.12.2022 – Защита проекта