

## СОДЕРЖАНИЕ

Список использованных обозначений, сокращений и терминов .....	9
Введение .....	10
1 Техническое задание на создание системы .....	13
1.1 Назначение и цели создания системы .....	13
1.2 Характеристика объекта автоматизации .....	13
1.2.1 Общее описание .....	13
1.2.2 Состав и порядок функционирования .....	13
1.3 Общие требования к системе.....	14
1.3.1 Требования к структуре и функционированию системы .....	14
1.3.2 Дополнительные требования.....	15
1.4 Требования к функциям, выполняемым системой.....	17
1.4.1 Регистрация пользователя в системе .....	17
1.4.2 Авторизация и аутентификация в системе.....	17
1.4.3 Движение заказа.....	17
1.4.4 Печать отчета о совершенном заказе.....	18
1.4.5 Подача заявки на доставку.....	18
1.4.6 Распределение заказов между курьерами .....	18
1.4.7 Рейтинг заказчиков и курьеров .....	19
1.5 Требования к видам обеспечения.....	19
1.5.1 Информационное обеспечение.....	19
1.5.2 Требования к математическому обеспечению .....	20

					ВКР — УлГТУ — 09.03.02 — 16/114 — 2020 ПЗ					
Изм.	Лист	№ докум.	Подп.	Дата	Автоматизированная система курьерской службы			Лит.	Лист	Листов
Разраб.		Иванова А.П.							5	109
Пров.		Кандаулов В.М.								
Консульт.		Рыбкина М.В.						ИСТдд — 41		
Нконтр.		Шукина В.Е.								
Утв.										

1.5.3 Требования к программному обеспечению .....	21
1.6 Анализ аналогичных разработок.....	22
1.6.1 «Ярославская служба доставки» .....	22
1.6.2 «БайкалСервис» .....	23
1.6.3 «До точки».....	23
1.6.4 Сравнение аналогичных разработок.....	23
2 Информационное обеспечение системы .....	25
2.1 Выбор средств управления данными.....	25
2.2 Проектирование базы данных .....	26
2.3 Проектирование файлов данных.....	30
2.4 Организация сбора, передачи, обработки и выдачи информации.....	30
3 Алгоритмическое обеспечение системы.....	32
3.1 Алгоритм авторизации пользователя .....	32
3.2 Алгоритм подачи заявки на доставку .....	32
3.3 Алгоритм получения печатной формы заказа .....	35
3.4 Алгоритм подсчета рейтинга.....	35
4 Программное обеспечение системы .....	37
4.1 Структура программного обеспечения и функции его компонентов .....	37
4.2 Выбор компонентов программного обеспечения.....	37
4.2.1 Операционная система .....	37
4.2.2 Инструментально средство разработки и язык программирования..	38
4.2.3 Вспомогательное программное обеспечение .....	39
4.3 Разработка прикладного программного обеспечения.....	39
4.3.1 Структура прикладного программного обеспечения .....	39

4.4 Особенности реализации, эксплуатации и сопровождения системы.....	44
4.5 Руководство пользователя .....	45
4.5.1 Требования к условиям эксплуатации.....	45
4.5.2 Инсталляция и настройка.....	45
4.5.3 Порядок и особенности работы.....	45
4.5.4 Исключительные ситуации и их обработка .....	63
5 Тестирование системы .....	67
5.1 Условия и порядок тестирования.....	67
5.2 Исходные данные для контрольных примеров.....	67
5.2.1 Описание ситуаций для элементов системы.....	67
5.3 Результаты тестирования .....	67
6 Экономический раздел.....	68
6.1 Определение трудоемкости разработки системы.....	68
6.2 Затраты на разработку системы .....	69
6.2.1 Затраты на материальные ресурсы .....	69
6.2.2 Затраты на электроэнергию .....	69
6.2.3 Затраты на оплату труда .....	70
6.2.4 Затраты на обязательные взносы .....	71
6.2.5 Затраты на амортизационные отчисления .....	71
6.2.6 Расчет прочих затрат .....	72
6.3 Себестоимость проекта .....	73
6.4 Определение экономической эффективности проекта.....	73
6.5 Выводы по технико-экономическому анализу и обоснованию проекта разработки.....	75
Заключение.....	76

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подп.	Дата		

Список использованных источников.....77

Приложение А.....79

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подп.	Дата		

## СПИСОК ИСПОЛЬЗОВАННЫХ ОБОЗНАЧЕНИЙ, СОКРАЩЕНИЙ И ТЕРМИНОВ

CSS – формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

HTML – стандартизированный язык разметки документов в сети интернет.

HTTP – протокол прикладного уровня передачи данных.

IDEF1X – одна из методологий семейства IDEF, применяемая для построения информационной модели.

АС – автоматизированная система.

БД – база данных.

ИС – информационная система.

Пользователь – человек, использующий разрабатываемую АС.

Хэширование – преобразование массива входных данных произвольной длины в выходную битовую строку установленной длины по определенному алгоритму.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подп.	Дата		

## ВВЕДЕНИЕ

Жизнь в современном мире можно очень кратко охарактеризовать одним словом – «спешка». Мы все время спешим, но все равно нам не хватает времени на жизнь. Поэтому появляются различные сервисы, которые ускоряют процесс получения услуг и товаров.

Спрос экспресс-доставки растет с каждым годом. В настоящее время это самый быстрый, удобный и простой способ доставки небольших грузов и документов. Чтобы отправить товар, нужно просто оставить заявку на сайте, указав точный адрес отправителя и получателя.

Курьерская доставка имеет много преимуществ. Это лучшее решение, когда время доставки является приоритетным. Преимущества курьерских служб, в отличие от Почты России:

- никаких очередей,
- отправка-получение товара по принципу «с рук в руки»;
- гибкая система ценообразования и специальные тарифные планы для постоянных клиентов;
- минимум времени на оформление отправления.

Наличие курьерских служб доставки играет важную роль в развитии малого бизнеса. Передача документов, транспортировка и сопровождение товаров нередко осуществляется именно службой доставки.

Целью данной работы является спроектировать и написать систему для автоматизации курьерской службы, позволяющую с легкостью подать заявку на доставку груза и получить его уже на следующий день. Причем распределение курьеров будет контролироваться администратором. Требуется сделать автоматически формируемые отчеты совершенных заказов. Предполагается реализовать систему в виде веб-сайта. Таким образом, она будет иметь простой интерфейс, понятный для пользователя. Заказчику понадобится минимум усилий и времени для подачи заявки на доставку.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подп.	Дата		

В отличие от других подобных систем, в данном случае предполагается система рейтинга курьеров и заказчиков, что повысит качество работы курьеров и прилежность заказчиков.

Данная особенность реализована в виде модального окна в личном кабинете заказчика и курьера. Рейтинг представляет собой пять звезд, каждая из которых соответствует оценке от одного до пяти. Для выставления оценки необходимо будет лишь выбрать одну из звезд.

Характеристики проекта более подробно рассмотрены в основной части пояснительной записки. Техническое задание содержит требования к функционалу и структуре системы. Использование информационных ресурсов рассматривается в информационном обеспечении. Принципы и алгоритмы работы механизма авторизации, оценки курьеров и заказчиков приведены в алгоритмическом обеспечении. Вопросы, связанные с технической стороной проекта, необходимое оборудование для работы системы, рассматриваются в разделе технического обеспечения. Раздел «Программное обеспечение» подробно описывает состав программного обеспечения.

Источник [9] рассказывает об особенностях языка C#, его синтаксисе. Содержащаяся в нем информация является очень полезной при написании большей части кода для создания системы.

В источнике [10] рассматривается несколько решений подсчета рейтинга и предлагается наиболее правильное – на основе доверительного интервала Вильсона. Данное решение и было реализовано в системе.

Источник [1] содержит руководство по разработке веб-приложений на платформе ASP.NET с применением паттерна MVC, основы Entity Framework, использование JavaScript.

Для работы с API Яндекс.Карт использовался источник [11]. В данном источнике подробно описываются классы и методы API, приводятся примеры использования тех или иных объектов.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		11

В источнике [12] рассматривается обзор библиотек для работы с файлами .pdf. Мною была выбрана библиотека iTextSharp, пример работы с которой описывается в источнике [13].

Для работы с .xls файлами использовался источник [14]. В данном источнике подробно рассказывается как создать документ, как сохранить его в файл. Описываются методы, свойства, события, которые позволяют манипулировать xml-документом.

Для реализации генерации QR-кода использовался источник [15]. В нем описывается работа с QR code API, содержится документация со списком параметров и примерами использования.

В источнике [16] рассказывается как сделать запрос GET с помощью C#. Приводится конкретный порядок действий и пример кода.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подп.	Дата		



# 1 ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА СОЗДАНИЕ СИСТЕМЫ

## 1.1 Назначение и цели создания системы

Разрабатываемая система предназначена для автоматизации работы курьерской службы, специализированная на грузоперевозку, а именно для автоматизации процесса отправки и получения заказа.

Данная система может использоваться в организациях, где администратор будет просматривать таблицы с заказами, с возможностью удалять заказы и получать отчет по всем заказам. Курьер будет проводить движение заказа, меняя его статус. Клиенты могут заказать доставку груза на сайте компании и получить печатную форму заказа.

Цели создания системы:

- уменьшить сроки доставки грузов;
- упростить взаимодействие заказчика и курьера.

## 1.2 Характеристика объекта автоматизации

### 1.2.1 Общее описание

Объектом автоматизации информационной системы (далее ИС) является деятельность курьерской службы, нацеленная на повышение эффективности их работы благодаря использованию веб-сайта – «Ульяновская служба доставки», где осуществлена возможность заказать доставку. Сайт имеет простой интерфейс, и не возникнет сложностей для совершения заказа. Данный сайт значительно упрощает отправку и получение посылок, освобождая грузовладельца от всех забот по организации перевозки, выбору вида транспорта и схемы доставки груза.

### 1.2.2 Состав и порядок функционирования

Курьерская служба — это выгодная и полезная услуга, которой можно воспользоваться как в пределах деловой, так и повседневной жизни. Основные

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		13

функции: оформление заказа, подбор автомобиля, возможность отследить статус заказа, доставка груза.

Основные операции по сбору и обработке информации:

- заполнение бланка заказа;
- поиск свободных курьеров с помощью телефонных звонков;
- распределение заказов между курьерами;
- заполнение накладной после доставки;
- уведомление о выполнении заказа;
- ведение отчета о заказах.

Все операции курьерской службы будут автоматизированы путем создания веб-сайта, обеспечивая доступ с любого устройства.

### **1.3 Общие требования к системе**

#### **1.3.1 Требования к структуре и функционированию системы**

Система должна представлять собой интернет-портал, на котором клиенты (зарегистрированные и незарегистрированные), курьер и администратор работают в трех разных ролях, разделенных механизмами авторизации.

В состав системы должны входить следующие подсистемы:

- подсистема авторизации, регистрации и разграничения доступа; авторизация пользователей под индивидуальными почтами и паролями; за каждой учетной записью закреплена своя роль;
- подсистема рейтинга заказчиков и курьеров;
- подсистема распределения заказов между курьерами;
- подсистема формирования отчета о заказах за определенный период;
- подсистема учета заказчиков;
- подсистема учета курьеров;

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подп.	Дата		

- подсистема расчета стоимости доставки груза и совершения заказа предназначена для расчета стоимости доставки пользователем, а затем для предоставления печатной формы сделанного заказа;
- подсистема движения заказа, предназначенная для отслеживания заказа и установления его статуса курьером.

Система должна состоять из двух частей: сервер управления веб-интерфейсом и клиентская часть в виде веб-сайта, которое используется обычными пользователями.

Перспективы развития системы предполагают добавление к вышеописанной системе функционала отслеживания груза.

### **1.3.2 Дополнительные требования**

#### *1.3.2.1 Требования к персоналу*

Численность пользователей может быть неограниченная.

Администратор должен владеть навыками пользования ПК на базовом уровне: важно уметь авторизовываться, заполнять формы, также иметь навыки работы с \*.xls файлами. Квалификация других сотрудников не имеет значения.

Режим работы персонала не влияет на использование системы.

Порядок подготовки администратора системы – знакомство и подготовка к работе с базой данных системы и ее основным функционалом, доступным для администратора.

#### *1.3.2.2 Требования к надежности*

Система должна сохранять работоспособность и обеспечивать восстановление своих функций при возникновении следующих внештатных ситуаций:

- при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы должно происходить после перезапуска ОС и запуска исполняемого файла системы;

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						15
Изм.	Лист	№ докум.	Подп.	Дата		

- при ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функции системы возлагается на ОС;
- при ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

Для защиты аппаратуры от бросков напряжения и коммутационных помех должны применяться сетевые фильтры.

В системе должна быть обеспечена возможность восстановления данных с внешнего накопителя после восстановления активного накопителя. БД системы необходимо резервировать минимум 1 раз в месяц.

Специальные дополнительные требования по составу и количественным значениям показателей надежности для подсистем модернизируемых модулей и, соответственно, к создаваемой системе в целом не предъявляются.

#### *1.3.2.3 Требования к патентной чистоте*

Система должна использовать только лицензионное программное обеспечение. Установка системы в целом, как и установка отдельных частей системы не должна предъявлять дополнительных требований к покупке лицензий на программное обеспечение сторонних производителей.

#### *1.3.2.4 Требования по стандартизации, унификации и тиражированию*

Разработка системы должна проводиться с соблюдением требований действующих государственных стандартов в соответствии с областью их распространения, одним из которых является использование современных технологий, протоколов и стандартов для взаимодействия с внешними системами.

Унификации подлежат все составляющие пользовательского интерфейса: система отчетности, формы отображения информации на экране, типы и формы сообщений и запросов пользователю.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						16
Изм.	Лист	№ докум.	Подп.	Дата		

## **1.4 Требования к функциям, выполняемым системой**

Реализация функций каждой из задач должна быть закончена в указанные сроки. Функции учета эффективности следует реализовать после реализации остальных задач.

### **1.4.1 Регистрация пользователя в системе**

Данная функция осуществляет регистрацию с разделением уровней доступа разрешений к управлению системы.

Входные данные: Почта, Пароль.

После успешной авторизации пользователю системы должно отобразиться окно работы.

Необходимо учесть, что в окно регистрации можно перейти от окна авторизации. В БД почта служит в качестве логина. Пароль должен хэшироваться для большей защиты.

Для регистрации нового пользователя требуется заполнить поля: почта, пароль и повторить пароль. Почта должна быть не менее 7 символов. Пароль должен быть не менее 3х символов.

### **1.4.2 Авторизация и аутентификация в системе**

Данная функция позволяет авторизоваться пользователю под своими учетными данными с разделением уровней доступа разрешений к управлению системой.

Входные данные: Почта, Пароль.

При успешности переход осуществляется аналогично регистрации.

### **1.4.3 Движение заказа**

Данная функция позволяет определять статус заказа и его стоимость.

Входные данные: статус и информация о грузе передаются из БД.

Смена статуса курьером и окончательная стоимость доставки отображается в интерфейсе пользователя.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						17
Изм.	Лист	№ докум.	Подп.	Дата		

#### **1.4.4 Печать отчета о совершенном заказе**

Функция позволяет заказчику получить печатную форму отчета о совершенном заказе.

Входные данные: информация о грузе (ФИО, номер телефона, адрес, дата, время, доступность отчета, статус, вес, длина, высота, ширина, оплата отправителем, цена) передаются из БД.

После смены статуса заказа курьером появляется возможность совершить печать отчета о заказе.

Необходимо учесть, что шаблон формы не зависит от данных и постоянен. Динамически изменяются только данные о пользователе и информация о заказе.

#### **1.4.5 Подача заявки на доставку**

Функция позволяет заказчику подать заявку на доставку груза.

Входные данные: информация о грузе (ФИО, номер телефона, адрес, дата, время, вес, длина, высота, ширина, оплата отправителем).

После успешной подачи заявки совершенный заказ отображается в личном кабинете заказчика.

#### **1.4.6 Распределение заказов между курьерами**

Функция позволяет распределять заказы между курьерами.

Входные данные: информация о грузе (адрес, район), и информация о курьере (район).

При условии, что к конкретному району прикреплен один курьер, заказ автоматически распределяется данному курьеру. В противном случае, администратор вручную распределяет заказы по курьерам. Если же груз является негабаритным, он распределяется на курьера с газелью, вне зависимости от района.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		18

#### **1.4.7 Рейтинг заказчиков и курьеров**

Функция позволяет заказчику оценивать работу курьера, а также курьеру оценивать заказчика.

Входные данные: оценка (от 1 до 5).

После доставки груза и заказчик, и курьер могут оценить услуги и добросовестность соответственно. Для заказчиков рейтинг также влияет на окончательную стоимость последующих заказов.

### **1.5 Требования к видам обеспечения**

#### **1.5.1 Информационное обеспечение**

Информационное обеспечение системы должно базироваться на базах данных предприятия «Ульяновская служба доставки». Для функционирования системы необходимо следующее информационное обеспечение:

- каталог заказов;
- справочник курьеров;
- каталог заказчиков.

Информационное обеспечение системы должно являться совокупностью информационных массивов в БД всей системы, используемы системой управления базой данных (СУБД) MS SQL.

Обеспечить контроль ввода в системе. Процедура не предусматривает придание юридической силы электронным документам.

В системе должен быть реализован многопользовательский режим доступа к данным. При этом должна осуществляться проверка на доступ к данным в информационной системе.

Время генерации страницы не должно превышать 5 секунд. Время загрузки страницы должно совпадать с временем генерации.

Пользователям запрещается получать данные с помощью языка запросов без использования прикладного программного обеспечения.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						19
Изм.	Лист	№ докум.	Подп.	Дата		

Необходимо использовать резервное копирование базы данных раз в день. При этом должны сохраняться резервные копии за предыдущие два дня, а также каждая копия, созданная в воскресенье, в течение последнего месяца.

### **1.5.2 Требования к математическому обеспечению**

Необходимо разработать алгоритмы, реализующие следующие возможности со стороны администратора:

- управление пользователями: удаление, просмотр деталей;
- формирование отчетов;
- авторизация;
- управление заказами: удаление, просмотр деталей;
- распределение заказов между курьерами.

Со стороны курьера:

- управление заказами: удаление, изменение, просмотр деталей;
- смена статуса заказа;
- расчет стоимости доставки;
- получение отчета;
- авторизация и регистрация;
- оценивание заказчиков.

Со стороны заказчика:

- подача заявки на доставку;
- печать информации о заказе;
- предварительный расчет стоимости доставки;
- оценивание курьеров;
- авторизация и регистрация.

Система должна содержать математические методы расчета:

- стоимости доставки исходя из параметров груза и расстояния отправки;

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						20
Изм.	Лист	№ докум.	Подп.	Дата		



- рейтинга курьеров и заказчиков, исходя из суммы и количества всех голосов;
- расстояния от города отправки до города получения.

При разработке информационной системы необходимо использовать возможности языка и платформы для реализации алгоритмов. При возможности использовать библиотеки, разрабатываемые по открытой лицензии. При отсутствии такой возможности необходимо самостоятельно реализовать функциональность.

### 1.5.3 Требования к программному обеспечению

Требования к используемому программному обеспечению системы определяются на стадии разработки технического проекта. При выборе программного обеспечения необходимо учитывать требования к патентной чистоте. В состав программных средств должны входить:

- серверная платформа, включающая БД и программная среда для веб-разработчиков;
- любой редактор электронных таблиц, работающий с расширением xls;
- браузер Yandex.

Клиентская часть разрабатываемой информационной системы должна корректно отображаться и функционировать на данном браузере.

Серверная часть информационной системы должна корректно разворачиваться на платформе, выполняя все принятые ограничения в БД.

Экранные формы должны соответствовать правилам стилизации вебориентированных информационных систем, принятым на момент разработки технического задания (далее ТЗ).

К обеспечению качества ПС предъявляются следующие требования:

- функциональность должна обеспечиваться выполнением подсистемами всех их функций;

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						21
Изм.	Лист	№ докум.	Подп.	Дата		

- надежность должна обеспечиваться за счет предупреждения ошибок — не допущения ошибок в готовых ПС;
- легкость применения должна обеспечиваться за счет применения покупных программных средств;
- эффективность должна обеспечиваться за счет принятия подходящих, верных решений на разных этапах разработки ПС и системы в целом;
- сопровождаемость должна обеспечиваться за счет высокого качества документации по сопровождению, а также за счет использования в программном тексте описания объектов и комментариев; использованием осмысленных и устойчиво различимых имен объектов; размещением не больше одного оператора в строке текста программы; избеганием создания фрагментов текстов программ с неочевидным смыслом;
- также на каждом этапе в разработке ПС должна проводится проверка правильности принятых решений по разработке и применению готовых ПС.

## **1.6 Анализ аналогичных разработок**

### **1.6.1 «Ярославская служба доставки»**

Служба обеспечивает услуги экспресс-доставки в любую точку земного шара, адресную доставку корреспонденции по Ярославской области в любых объёмах, транспортно-экспедиционные услуги. Программа работает только на платформе веб-приложение. К преимуществам данной системы относятся отслеживание отправления. К недостаткам, неавтоматичная система подачи заявок.

По сравнению с моей системой здесь отсутствует авторизация пользователя, то есть пользователь не имеет своего личного кабинета, в котором он может посмотреть свою историю заказов.

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						22
Изм.	Лист	№ докум.	Подп.	Дата		

### 1.6.2 «БайкалСервис»

В отличие от предыдущей курьерской службы, подача заявок полностью автоматическая. В функционал данного сайта также входит отслеживание отправок и расчет стоимости грузоперевозки. Есть возможность получить печатную форму заказа. Также пользователю доступны дополнительные услуги: доставка груза ко времени, система уведомлений, где помимо обычных способов смс и e-mail можно выбрать мессенджеры с чат-ботом.

Также данная система имеет как мобильную версию сайта, так и мобильное приложение, что не реализовано в моей системе. Но по сравнению с разрабатываемой ИС, на сайте «БайкалСервис» отсутствует рейтинг заказчиков и курьеров.

### 1.6.3 «До точки»

Еще один веб-сайт службы доставки. По сравнению с предыдущим сайтом, но имеет более простой и удобный интерфейс сайта. Интуитивно понятный личный кабинет. К основным задачам, решаемых в данной системе является ведение базы клиентов, расчет стоимости доставки, печать накладной и оформление доставки.

К основным отличительным функциям относятся: статистика совершенных заказов; оповещения заказчиков. В отличие от моей разработки здесь отсутствует система рейтинга заказчиков и курьеров, а также курьер сам выбирает заказ, следовательно, может возникнуть ситуация, когда заказ может быть не принят.

### 1.6.4 Сравнение аналогичных разработок

Итоговое сравнение расширений описанных аналогов представлено в таблице 1.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						23
Изм.	Лист	№ докум.	Подп.	Дата		

Таблица 1 – Сравнение аналогичных разработок

Критерий	Ярославская служба доставки	БайкалСервис	Экспресс Точка Ру	Разрабатываемая ИС
Отчеты	-	+	+	+
Рейтинг	-	-	-	+
Личный кабинет	-	+	+	+
Отслеживание	+	+	-	-
Понятный функционал	+	-	+	+
Мобильное приложение	-	+	+	-
Распределение заказов	системой	системой	курьером	системой

Исходя из сравнения систем, представленных в таблице 1, разрабатываемая информационная система имеет ряд преимуществ перед своими конкурентами, но и имеет недостатки в плане количества и проработанности функционала.

## 2 ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

### 2.1 Выбор средств управления данными

Согласно п. 1.3.2 технического задания необходимо использовать современные бесплатные технологии по управлению данными. Согласно п. 1.5.1 технического задания в качестве хранилища данных должна выступать реляционная база данных, система управления которой поддерживает транзакции и целостность данных.

Для выбора подходит множество современных реляционных баз данных, однако основываясь на компетенциях, полученных при обучении в бакалавриате, выбор был остановлен на следующих технологиях:

- MySQL4;
- PostgreSQL;
- Microsoft SQL Server.

Для выявления наиболее подходящей технологии были подобраны критерии (нумерация в списке соответствует нумерации в сравнительной таблице ниже):

1. Наличие удобной среды исполнения запросов и отображения данных.
2. Наличие удобной среды отладки и диагностики запросов к данным.
3. Простота установки и настройки.
4. Поддержка современных ORM-систем.
5. Наличие документации с примерами.
6. Мультиплатформенность.
7. Изучение технологии разработчиком.

Сводка по сравнению технологий представлена в таблице 2.

Таблица 2 – Сводка по сравнению технологий доступа к данным

Критерий	PostgreSQL	MySQL	MS SQL Server
1	2	3	4
K1	Нет	Есть, phpMyAdmin	Есть, SSMS
K2	Нет	Нет	Есть, SSMS

## Продолжение таблицы 2

1	2	3	4
K3	Нет	Да	Да
K4	Есть	Частично	Есть
K5	Частично	Частично	Есть
K6	Да	Да	Частично
K7	Нет	Да	Да

Согласно выбранным критериям было принято решение выбрать в качестве доступа к данным технологию Microsoft SQL Server. Однако система разрабатывалась с учетом быстрой смены подсистемы хранения данных. Это достигается путем использования подхода к проектированию информационной системы Code-First.

## 2.2 Проектирование базы данных

На рисунке 1 представлена концептуальная схема данных в нотации IDEF1X.

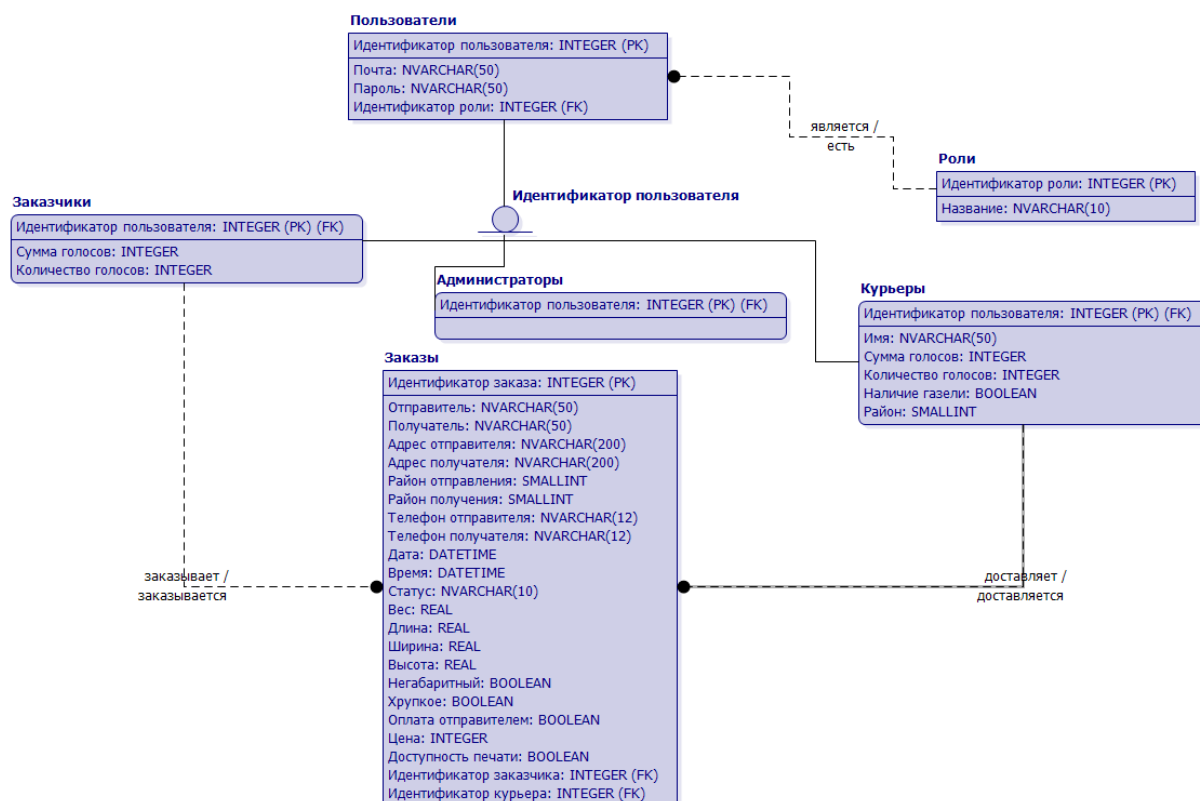


Рисунок 1 – Модель данных

Описание сущностей приведено в таблице 3.

Таблица 3 – Сущности концептуальной схемы

Название сущности	Описание
Пользователи	Сведения о пользователях, которые зарегистрированы в системе
Роли	Сведения о ролях Пользователей системы
Заказы	Сведения о заказах
Заказчики	Сведения о заказчиках
Курьеры	Сведения о курьерах
Администраторы	Сведения об администраторах

Далее (таблицы 4 – 11) приведено описание атрибутов сущностей. Названия таблиц (в квадратных скобках) и полей (в фигурных скобках) базы данных сопоставлены с соответствующими сущностями и атрибутами.

Атрибуты сущности «Пользователи» представлены в таблице 11.

Таблица 4 – Атрибуты сущности «Пользователи» (поля таблицы [Users])

Название атрибута	Тип и размер	Описание
Идентификатор пользователя {Id}	int	Уникальный номер пользователя
Почта {Email}	nvarchar(50)	Почта пользователя
Пароль {Password}	nvarchar(50)	Пароль пользователя
Идентификатор роли {RoleId}	int	Уникальный номер роли

Атрибуты сущности «Роли» представлены в таблице 5.

Таблица 5 – Атрибуты сущности «Роли» (поля таблицы [Roles])

Название атрибута	Тип и размер	Описание
Идентификатор роли {Id}	int	Уникальный номер роли
Название {Name}	nvarchar(10)	Название роли

Атрибуты сущности «Заказы» представлены в таблице 6.

Таблица 6 – Атрибуты сущности «Заказы» (поля таблицы [Orders])

Название атрибута	Тип и размер	Описание
<b>1</b>	<b>2</b>	<b>3</b>
Идентификатор заказа {OrderId}	int	Уникальный номер заказа

Продолжение таблицы 6

1	2	3
Отправитель {Sender}	nvarchar(100)	ФИО отправителя
Получатель {Receiver}	nvarchar(100)	ФИО получателя
Адрес отправителя {Address_Sender}	nvarchar(200)	Адрес принятия посылки
Адрес получателя {Address_Receiver}	nvarchar(200)	Адрес доставки посылки
Район отправления {Area_Sender}	tinyint	Район отправления посылки
Район получения {Area_Receiver}	tinyint	Район получения посылки
Телефон отправителя {Address_Sender}	nvarchar(12)	Номер телефона отправителя
Телефон получателя {Address_Receiver}	nvarchar(12)	Номер телефона получателя
Дата {Date}	datetime	Дата желаемой отправки
Время {Time}	datetime	Время желаемой отправки
Статус {Status}	nvarchar(10)	Статус заказа
Вес {Weight}	real	Вес груза
Длина {Length}	real	Длина груза
Ширина {Width}	real	Ширина груза
Высота {Height}	real	Высота груза
Негабаритный {Big}	boolean	Отметка, является ли груз негабаритным
Хрупкое {Fragile}	boolean	Отметка, является ли груз хрупким
Оплата отправителем {Who_pay}	boolean	Отметка оплачивает заказ отправитель или получатель
Цена {Price}	int	Цена заказа
Доступность печати {Print}	boolean	Отметка о доступности печати заявки
Идентификатор заказчика {CustomerId}	int	Уникальный номер заказчика
Идентификатор курьера {CourierId}	int	Уникальный номер курьера



Атрибуты сущности «Заказчики» представлены в таблице 7.

Таблица 7 – Атрибуты сущности «Заказчики» (поля таблицы [Customers])

Название атрибута	Тип и размер	Описание
Идентификатор пользователя {Id}	int	Уникальный номер пользователя
Сумма голосов {sumVotes}	nvarchar(100)	Сумма всех голосов данному заказчику
Количество голосов {totalVotes}	nvarchar(100)	Количество голосов данному заказчику

Атрибуты сущности «Курьеры» представлены в таблице 8.

Таблица 8 – Атрибуты сущности «Курьеры» (поля таблицы [Couriers])

Название атрибута	Тип и размер	Описание
Идентификатор пользователя {Id}	int	Уникальный номер пользователя
Сумма голосов {sumVotes}	nvarchar(100)	Сумма всех голосов данному курьеру
Количество голосов {totalVotes}	nvarchar(100)	Количество голосов данному курьеру
Наличие газели {time}	boolean	Отметка о вождении газели
Район {Area}	tinyint	Район, к которому привязан курьер

Атрибуты сущности «Администраторы» представлены в таблице 9.

Таблица 9 – Атрибуты сущности «Администраторы» (поля таблицы [Admins])

Название атрибута	Тип и размер	Описание
Идентификатор пользователя {Id}	int	Уникальный номер пользователя

В таблице 10 приведено описание индексов созданных в БД таблиц.

Таблица 10 – Описание индексов таблиц

№	Название индекса	Название таблицы	Поля индекса	Уникальность
1	2	3	4	5
1	IX_RoleId	Users	RoleId	нет
2	IX_Id	Customers	Id	нет
3	IX_Id	Couriers	Id	нет

## Продолжение таблицы 10

1	2	3	4	5
4	IX_Id	Admins	Id	нет
5	IX_CourierId	Orders	CourierId	нет
6	IX_CustomerId	Orders	CustomerId	нет
7	OrderId	Orders	OrderId	да
8	Id	Users, Roles, Customers, Couriers, Admins	Id	да

### 2.3 Проектирование файлов данных

Результатом работы экспорта отображаемых на веб-странице таблиц должен быть файл отчёта, в формате \*.xls. Получается, что во время работы системы создается дополнительный файл, который сохраняется в оперативную память.

Печатную форму заказа в виде таблицы с данными заказа было решено представить в виде PDF документа.

### 2.4 Организация сбора, передачи, обработки и выдачи информации

Информация о необходимых данных заносится посредством интерфейса ИС, данные берутся из БД, обрабатываются специальными алгоритмами ИС и передаются в печатной форме посредством сети «Интернет».

Для осуществления сбора информации от пользователя используется веб-страница. Передача информации происходит через интернет, реализуемая с помощью протокола HTTP, причем используемая архитектура предполагается клиент-серверной.

Достоверность выдачи результатов обеспечивается передаваемыми данными из БД, контролирующими целостность данных. Для этого в формах для исключения ссылок на несуществующие объекты решено использовать выпадающие списки, установка шаблона и диапазона значений.

Для обслуживания в режиме администрирования в системе могут происходить следующие процессы:

- применение миграций данных;

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						30
Изм.	Лист	№ докум.	Подп.	Дата		

- применение миграций схем данных;
- обновление исполняемых файлов программных модулей.

Выдача информации осуществляется на экран монитора, а также в формате \*.xls и \*.pdf.

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						31
Изм.	Лист	№ докум.	Подп.	Дата		

### 3 АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

#### 3.1 Алгоритм авторизации пользователя

Общая характеристика: Алгоритм предназначен для авторизации с различным уровнем доступа: как курьер, администратор и как заказчик.

Используемые данные: введенные пользователем значения пары логин-пароль, поля таблицы «Users».

Результаты выполнения: авторизованный пользователь.

Математическое описание: отсутствует.

Логическое описание: Блок-схема алгоритма представлена на рисунке 2.

#### 3.2 Алгоритм подачи заявки на доставку

Общая характеристика: Алгоритм предназначен для подачи заявки на доставку заказчиком.

Используемые данные: поля, заполняемые заказчиком.

Результаты выполнения: заявка на доставку.

Математическое описание: отсутствует.

Логическое описание: Блок-схема алгоритма представлена на рисунке 3.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подп.	Дата		

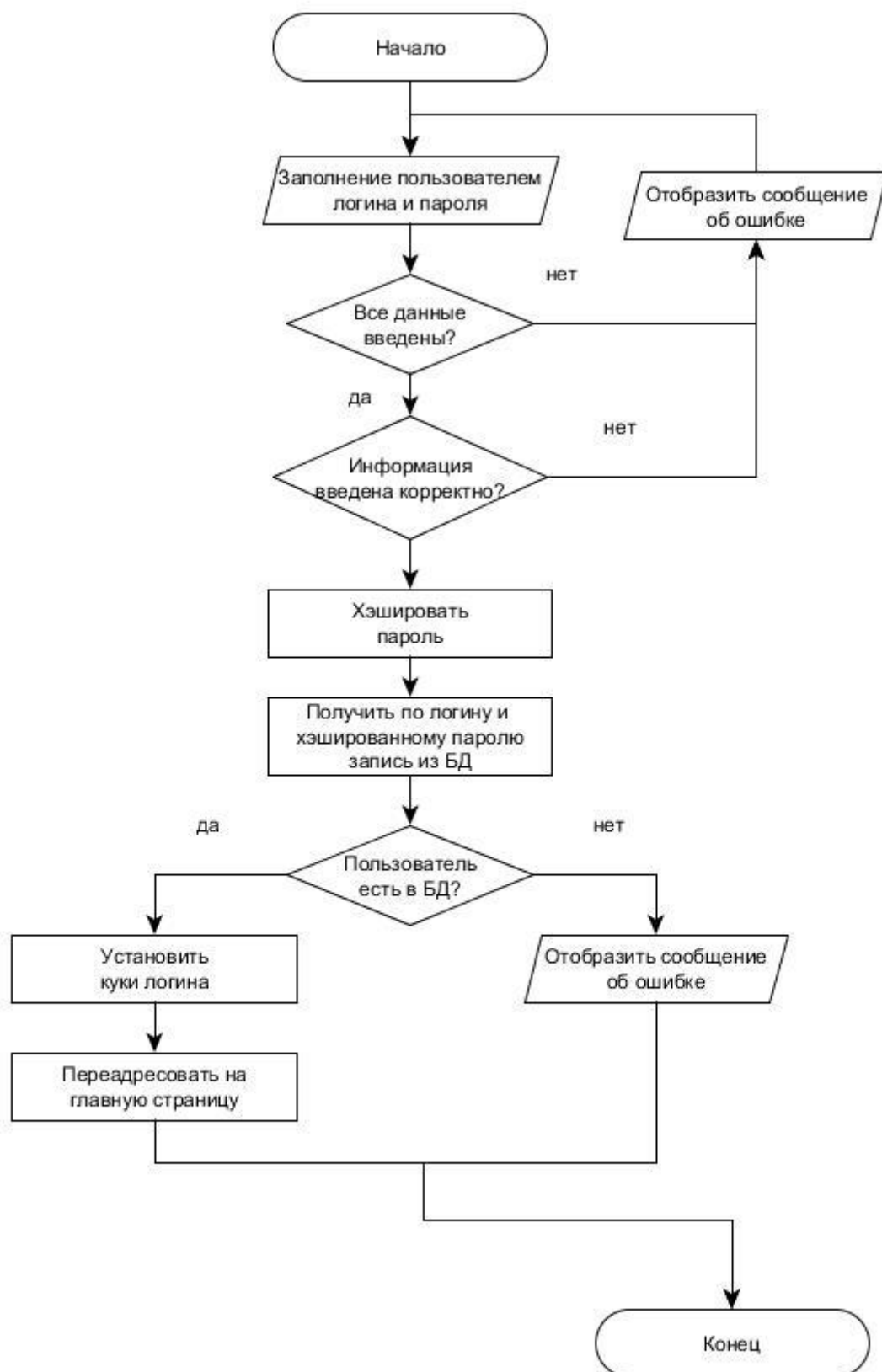


Рисунок 2 – Блок-схема алгоритма аутентификации

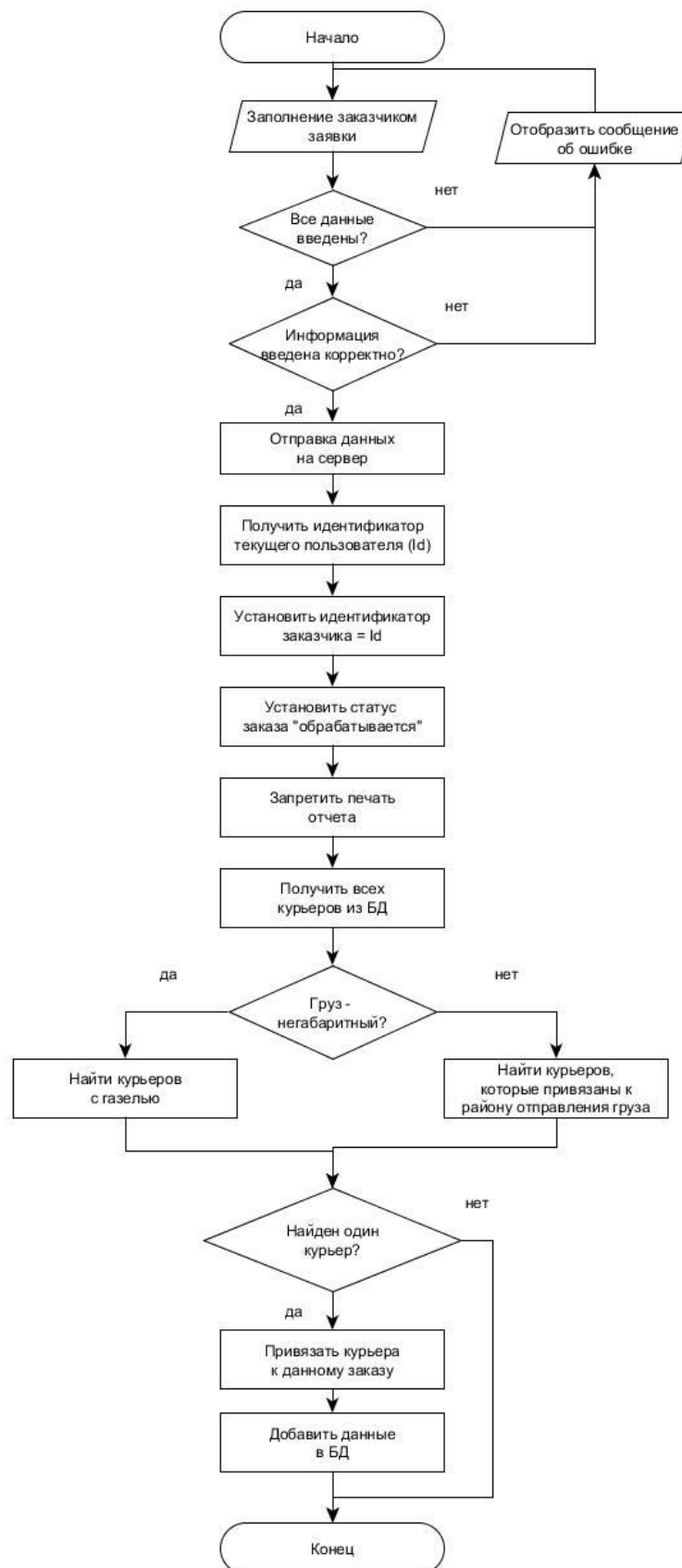


Рисунок 3 – Блок-схема алгоритма подачи заявки на доставку

### 3.3 Алгоритм получения печатной формы заказа

Общая характеристика: Алгоритм предназначен для получения печатной формы заказа курьером и заказчиком

Используемые данные: данные из БД.

Результаты выполнения: .pdf документ.

Математическое описание: отсутствует.

Логическое описание: Блок-схема алгоритма представлена на рисунке 4.



Рисунок 4 – Блок-схема алгоритма получения печатной формы заказа

### 3.4 Алгоритм подсчета рейтинга

Общая характеристика: Алгоритм предназначен для подсчета рейтинга заказчиков и курьеров.

Используемые данные: рейтинг, данные из БД.

Результаты выполнения: итоговый рейтинг.

Математическое описание:

$$(\hat{p} + \frac{z_{\alpha/2}^2}{2n} \pm z_{\alpha/2} \sqrt{[\hat{p}(1 - \hat{p}) + z_{\alpha/2}^2 / 4n] / n}) / (1 + z_{\alpha/2}^2 / n),$$

где  $\hat{p}$  (phat) – доля положительных оценок;

$z_{\alpha/2}$  (z) – квантиль  $(1-\alpha/2)$  стандартного нормального распределения;

n (total) – общее число оценок.

Логическое описание: Блок-схема алгоритма представлена на рисунке 5.

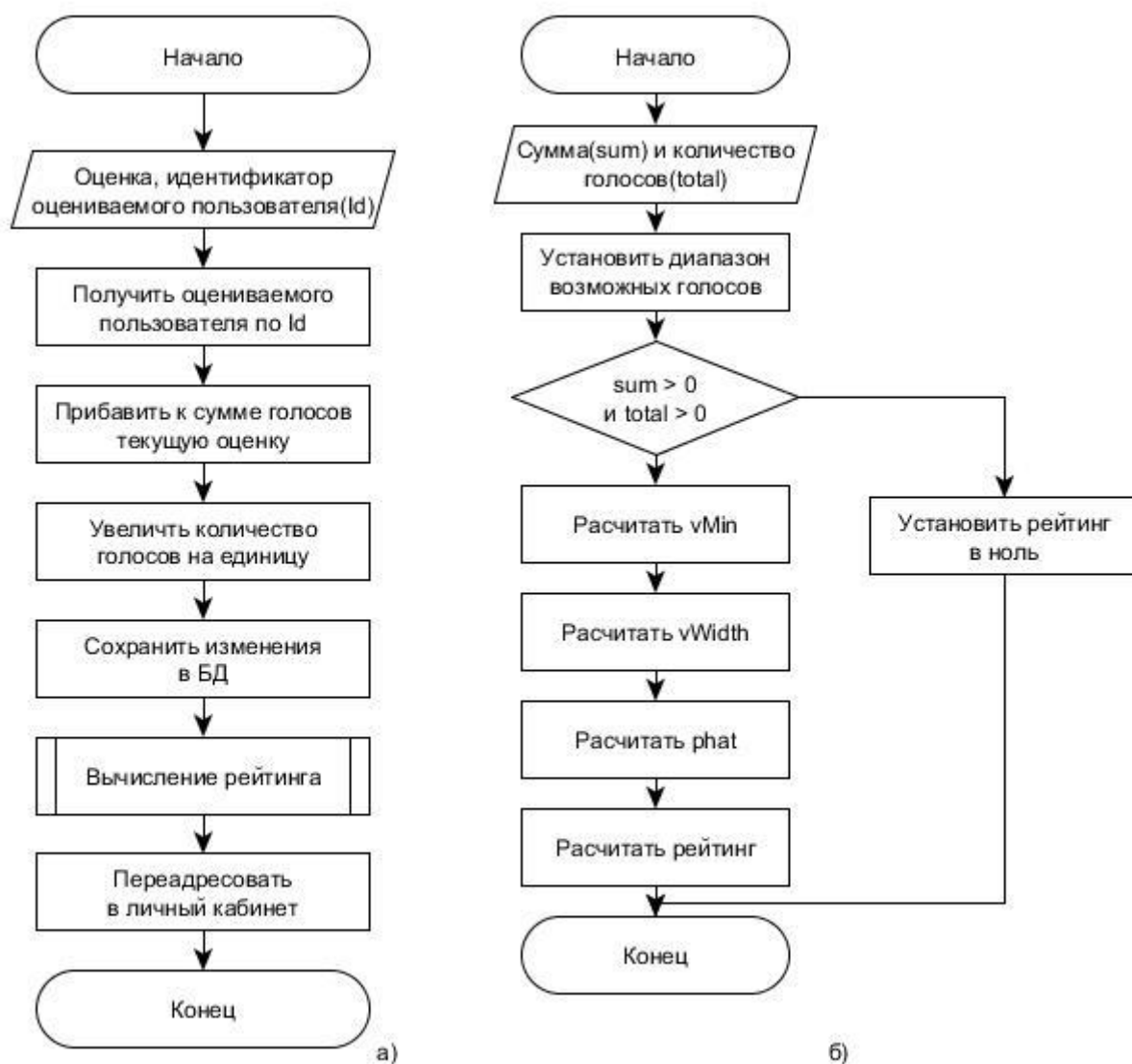


Рисунок 5 – Блок-схема алгоритма подсчета рейтинга (а) и вычисления рейтинга (б)



## 4 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

### 4.1 Структура программного обеспечения и функции его компонентов

При создании информационной системы в качестве операционной системы была использована операционная система Window 10.

В качестве инструментального средства разработки использовалась среда Microsoft Visual Studio 2017.

Так как необходимо, чтобы система функционировала в основных современных браузерах, то при разработке, а также на этапах тестирования, были использованы браузеры Edge и Яндекс.Браузер.

В качестве текстового редактора для написания документации к ИС выступал Microsoft Office.

В качестве платформы для сервера решено выбрать ASP.NET. Создание веб-сайта требует использования различных технологий: графический дизайн, верстка HTML, CSS, программирование клиентской части – интерфейс пользователя, и программирование серверной части.

### 4.2 Выбор компонентов программного обеспечения

#### 4.2.1 Операционная система

Согласно п. 1.5.3 технического задания система должна быть мультиплатформенной. Таким образом, для работы АС подходит большинство современных операционных сред, в том числе бесплатных.

Например, рассмотрим ОС Ubuntu 16.04 и ОС Windows 10. Обе операционные системы позволяют развернуть web-сервер и систему управления базами данных для корректной работы программного обеспечения. Однако с точки зрения разработки целесообразнее использовать операционную систему Windows 10 из-за следующих факторов:

— наличие мощных инструментальных средств разработки и отладки;

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						37
Изм.	Лист	№ докум.	Подп.	Дата		

- наличие удобных средств администрирования web-сервера;
- наличие опыта разработки программ на данной операционной системе;
- наличие удобных средств администрирования баз данных.

#### **4.2.2 Инструментально средство разработки и язык программирования**

Важным этапом в разработке автоматизированной системы является выбор набора инструментальных средств и языков программирования. От этого выбора зависит производительность системы, скорость поиска и исправления ошибок.

Согласно п. 1.5.3 технического задания система должна быть мультиплатформенной. Таким образом, необходимо подобрать такой язык программирования, который должен соответствовать данным требованиям.

Также согласно п. 1.3.2 технического задания желательно использование технологий, являющихся активно разрабатываемыми и поддерживаемыми широким сообществом разработчиков.

Таких языков программирования можно выделить несколько. Например, язык программирования Ruby с фреймворком Ruby on Rails и язык программирования Visual C# с фреймворком ASP.NET MVC. Оба языка прочно входят в рейтинг 20-ти самых популярных языков программирования ТЮВЕ.

Оба сравниваемых языка программирования похожи друг на друга, а также и фреймворки для построения web-ориентированных информационных систем имеют общую функциональность и принцип построения (паттерн MVC).

Однако был выбран язык Visual C# с фреймворком ASP.NET MVC из-за следующих отличительных особенностей:

- отличная поддержка ORM-систем, работающих с MS SQL Server;

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						38
Изм.	Лист	№ докум.	Подп.	Дата		

- простота установки исполняемой среды;
- наличие опыта разработки программ на данном языке программирования.

Соответственно, инструментальная среда для разработки на языке Visual C# была выбрана Visual Studio 2017 Community. Она является бесплатной и содержит все необходимые средства разработки, диагностики, отладки и тестирования программного кода.

#### **4.2.3 Вспомогательное программное обеспечение**

Было решено использовать расширение \*.xls для экспорта и сохранения отчетов в виде таблицы. Это формат файлов программы Microsoft Excel, один из самых популярных и эффективных инструментов для создания электронных таблиц, диаграмм и многого другого. С этим продуктом вы можете экспортировать файлы в другие форматы, например, xml.

Печатную форму заказа в виде таблицы с данными заказа было решено представить в виде PDF документа. Для создания PDF документов решено использовать iTextSharp – популярная открытая библиотека, которая позволяет создавать и манипулировать PDF документами.

Для валидации адреса и расчета расстояния между адресами используется API Яндекс.Карт. Это набор сервисов, которые позволяют использовать картографические данные и технологии Яндекса.

### **4.3 Разработка прикладного программного обеспечения**

#### **4.3.1 Структура прикладного программного обеспечения**

Система включает в себя следующие подсистемы:

- подсистема авторизации, регистрации и разграничения доступа; авторизация пользователей под индивидуальными почтами и паролями; за каждой учетной записью закреплена своя роль;
- подсистема рейтинга заказчиков и курьеров;

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						39
Изм.	Лист	№ докум.	Подп.	Дата		

- подсистема распределения заказов между курьерами;
- подсистема формирования отчета о заказах за определенный период;
- подсистема учета заказчиков;
- подсистема учета курьеров;
- подсистема расчета стоимости доставки груза и совершения заказа предназначена для расчета стоимости доставки пользователем, а затем для предоставления печатной формы сделанного заказа;
- подсистема движения заказа, предназначенная для отслеживания заказа и установления его статуса курьером.

Данное дробление на подсистемы было выбрано в соответствии с техническим заданием на разработку информационной системы и отраслевыми стандартами проектирования ИС.

Спецификация системы представлена в таблице 11.

Таблица 11 – Спецификация системы

№	Название модуля	Описание
1	Подсистема авторизации, регистрации и разграничения доступа	Предназначена для авторизации пользователей под индивидуальными почтами и паролями; за каждой учетной записью закреплена своя роль
2	Подсистема рейтинга заказчиков и курьеров	Предназначена для стимуляции качественной работы сотрудников и добросовестного отношения клиентов
3	Подсистема распределения заказов между курьерами	Предназначена для распределения заказов между курьерами
4	Подсистема формирования отчета о заказах за определенный период	Предназначена для формирования отчета о заказах администратором за определенный период
5	Подсистема учета заказчиков	Предназначена для ведения учета заказчиков
6	Подсистема учета курьеров	Предназначена для ведения учета заказчиков
7	Подсистема расчета стоимости доставки груза и совершения заказа	Предназначена для расчета стоимости доставки груза, в зависимости от параметров груза и дальности перевозки
8	Подсистема движения заказа	Предназначена для отслеживания заказа и установления его статуса курьером

#### 4.3.1.1 Программный модуль «Подсистема авторизации, регистрации и разграничения доступа»

Спецификация модуля представлена в таблице 12.

Таблица 12 – Спецификация модуля «Подсистема авторизации, регистрации и разграничения доступа»

№	Название	Описание
<b>Классы-контроллеры</b>		
1	public class AccountController	Содержит методы для входа и выхода с сайта, изменения логина, сброса пароля, запоминания пароля, хэширование пароля
2	public class UsersController	Содержит методы для добавления, редактирования и удаления информации о пользователях системы
<b>Классы для авторизации и регистрации</b>		
3	public class LoginModel	Содержит список полей для авторизации
4	public class RegisterModel	Содержит список полей для регистрации
<b>Классы для настройки профиля Пользователя</b>		
5	public class User	Содержит список полей для отображения информации о пользователях
6	public class Role	Содержит список полей для разграничения доступа по ролям
<b>Класс провайдера ролей</b>		
7	public class CustomRoleProvider	Содержит методы для получения набора ролей пользователя и определения, выполняет ли пользователь определенную роль в системе

#### 4.3.1.2 Программный модуль «Подсистема рейтинга заказчиков и курьеров»

Спецификация модуля представлена в таблице 13.

Таблица 13 – Спецификация модуля «Подсистема рейтинга заказчиков и курьеров»

№	Название	Описание
<b>Классы-контроллеры</b>		
1	public class CouriersController	Содержит методы для оценивания курьерами заказчиков
2	public class CustomersController	Содержит методы для оценивания заказчиками курьеров
<b>Классы для подсчета рейтинга</b>		
3	public class Customer	Содержит вычисляемое поле для подсчета рейтинга заказчика
4	public class Courier	Содержит вычисляемое поле для подсчета рейтинга курьера

#### 4.3.1.3 Программный модуль «Подсистема распределения заказов между курьерами»

Спецификация модуля представлена в таблице 14.

Таблица 14 – Спецификация модуля «Подсистема распределения заказов между курьерами»

№	Название	Описание
<b>Классы-контроллеры</b>		
1	public class OrdersController	Содержит методы для распределения заказов по районам и назначения курьера на заказ

*4.3.1.4 Программный модуль «Подсистема формирования отчета о заказах за определенный период»*

Спецификация модуля представлена в таблице 15.

Таблица 15 – Спецификация модуля «Подсистема формирования отчета о заказах за определенный период»

№	Название	Описание
<b>Классы-контроллеры</b>		
1	public class OrdersController	Содержит метод для сохранения отчета в виде таблицы

*4.3.1.5 Программный модуль «Подсистема учета заказчиков»*

Спецификация модуля представлена в таблице 16.

Таблица 16 – Спецификация модуля «Подсистема учета заказчиков»

№	Название	Описание
<b>Классы-контроллеры</b>		
1	public class CustomersController	Содержит методы для добавления, удаления и изменения заказчиков
<b>Классы для настройки профиля заказчика</b>		
2	public class Customer	Содержит список полей для отображения информации о заказчиках

*4.3.1.6 Программный модуль «Подсистема учета курьеров»*

Спецификация модуля представлена в таблице 17.

Таблица 17 – Спецификация модуля «Подсистема учета курьеров»

№	Название	Описание
<b>Классы-контроллеры</b>		
1	public class CouriersController	Содержит методы для добавления, удаления и изменения курьеров
<b>Классы для настройки профиля курьера</b>		
2	public class Courier	Содержит список полей для отображения информации о курьерах

#### 4.3.1.7 Программный модуль «Подсистема расчета стоимости доставки груза и совершения заказа»

Спецификация модуля представлена в таблице 18.

Таблица 18 – Спецификация модуля «Подсистема расчета стоимости доставки груза и совершения заказа»

№	Название	Описание
<b>Классы-контроллеры</b>		
1	public class OrdersController	Содержит методы для совершения заказа
<b>Классы для работы с заказами</b>		
2	public class Order	Содержит поля для создания и редактирования заказов

#### 4.3.1.8 Программный модуль «Подсистема движения заказа»

Спецификация модуля представлена в таблице 19.

Таблица 19 – Спецификация модуля «Подсистема движения заказа»

№	Название	Описание
<b>Классы-контроллеры</b>		
1	public class OrdersController	Содержит методы для изменения и определения статуса заказа

### 4.4 Особенности реализации, эксплуатации и сопровождения системы

Система будет эксплуатироваться работниками курьерских служб. Для использования системы необходимо установить на рабочем месте сотрудника один из современных браузеров и иметь возможность выхода в Интернет.

Выпадающая панель с поисковыми подсказками адресов и получение протяженности маршрута реализована при помощи API Яндекс.Карт.

Использование фреймворка языка разметки CSS – bootstrap при стилизации страниц из-за простоты и популярности в использовании.



## 4.5 Руководство пользователя

### 4.5.1 Требования к условиям эксплуатации

Для работы приложения требуется один из современных браузеров (Яндекс, Chrome, Edge и другие). Требования к аппаратному обеспечению устанавливаются, исходя из требований браузера.

Квалификация администратора системы – высшее техническое образование. Квалификация других сотрудников не имеет значения. Заказчики не должны обладать особыми навыками работы с ПК: важно уметь авторизовываться и заполнять формы.

### 4.5.2 Инсталляция и настройка

Для работы в системе всем Пользователям необходимо установить один из современных браузеров (Яндекс, Chrome, Edge и другие). Запустить браузер и открыть соответствующую веб-страницу.

### 4.5.3 Порядок и особенности работы

#### 4.5.3.1 Для роли «Заказчик»

Чтобы авторизоваться в системе как заказчик, нужно в правом верхнем углу главной страницы выбрать «Войти», а для того чтобы зарегистрироваться, нужно выбрать «Зарегистрироваться» (рисунок 6).



[Главная](#) | [О нас](#) | [Расчет стоимости доставки](#) |

[Войти](#) [Зарегистрироваться](#)

Рисунок 6 – Кнопки авторизации и регистрации

Нажав на «Войти» перед заказчиком откроется поле ввода информации для входа, где нужно ввести почту и пароль для авторизации (рисунок 7).

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						45
Изм.	Лист	№ докум.	Подп.	Дата		

## Вход

Почта

Пароль

Войти

[Назад](#)

Рисунок 7 – Поле авторизации

Авторизовавшись, заказчик может в своем личном кабинете поменять свою почту или пароль, нажав на кнопку изменить «Изменить почту или пароль»; перед ним откроется окно, в котором нужно ввести, что он хочет изменить (рисунок 8).

## Изменить

Почта

Пароль

Save

[Назад](#)

Рисунок 8 – Страница изменения почты и пароля

Если заказчик не авторизован, то нажав на кнопку «Зарегистрироваться» перед ним откроется поле ввода информации для регистрации, где нужно ввести почту, пароль, и повторить пароль для регистрации (рисунок 9). При вводе полей без ошибок, заказчик успешно регистрируется.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						46
Изм.	Лист	№ докум.	Подп.	Дата		

## Регистрация

Почта	<input type="text"/>
Пароль	<input type="password"/>
Подтверждение пароля	<input type="password"/>
Курьер <input type="checkbox"/>	
<input type="button" value="Зарегистрироваться"/>	

Рисунок 9 – Страница регистрации

Оказавшись на главной странице системы, заказчик может сразу перейти в личный кабинет по ссылке в верхнем правом углу (рисунок 10).

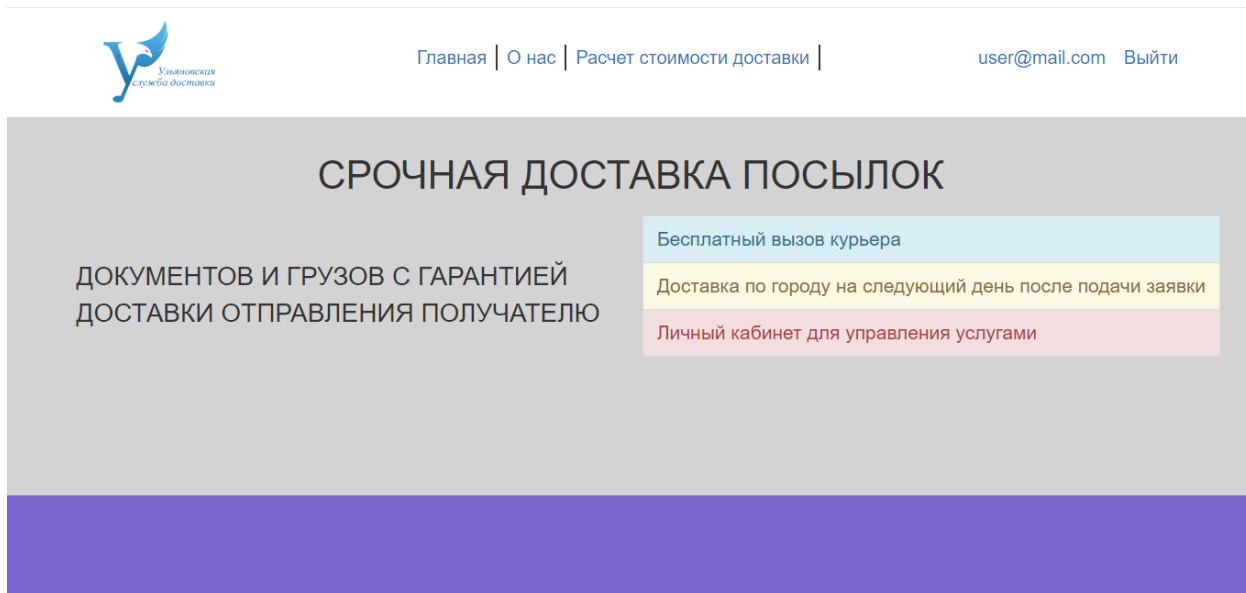


Рисунок 10 – Главная страница

В личном кабинете отображаются все заявки заказчика. Также есть возможность подачи новых заявок (рисунок 11).

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						47
Изм.	Лист	№ докум.	Подп.	Дата		

## Личный кабинет

Заказы							
Отправитель	Получатель	Адрес отправителя	Адрес получателя	Дата	Время	Статус	
Иванова Анастасия ...	Шевчук Максим Сер...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	06.06.2020	8:00	обрабатывается	<a href="#">Детали</a>
Иванова Анастасия ...	Шевчук Максим Сер...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	06.06.2020	13:00	обрабатывается	<a href="#">Детали</a>

[Совершить заказ](#)  
[Изменить почту или пароль](#)

## Рисунок 11 – Личный кабинет

Для этого нужно нажать «Совершить заказ» и в открывшейся форме ввести данные о заказе и грузе, как показано на рисунках 12-13.

## Подача заявки

Отправитель	Ангелина
Получатель	Петр
Адрес отправителя	Россия, Ульяновск, улица Розы Люксембург, 34
Район отправки	Ленинский
Адрес получателя	Россия, Ульяновск, Камышинская улица, 4
Район получения	Засвияжский
Номер отправителя	89339994455
Номер получателя	81112223344
Дата	07.06.2020
Время	8:30 ▼
Вес, кг	4

## Рисунок 12 – Страница подачи заявки на доставку

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						48
Изм.	Лист	№ докум.	Подп.	Дата		

Вес, кг   
 Длина, см   
 Ширина, см   
 Высота, см   
 негабаритный ☐  
 хрупкое ☐  
 Оплата ☒  
 отправителем  
 Цена   
 Рассчитать  
 Создать

Рисунок 13 – Страница подачи заявки на доставку(продолжение)

После того, как пользователь введет необходимую информацию, по нажатию кнопки «Рассчитать» в поле «Цена» появляется примерная стоимость доставки (рисунок 14).

Оплата ☒  
 отправителем  
 Цена   
 356р. Это примерная стоимость доставки. Окончательная установится после взвешивания груза  
 Рассчитать  
 Создать

Рисунок 14 – Расчет стоимости

Для завершения подачи заявки, заказчику необходимо нажать кнопку «Создать». Перед ним откроется страница личного кабинета, где отобразится новая заявка на доставку (рисунок 15).

#### Личный кабинет

Заказы							
Отправитель	Получатель	Адрес отправителя	Адрес получателя	Дата	Время	Статус	
Иванова Анастасия ...	Шевчук Максим Сер...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	06.06.2020	8:00	обрабатывается	<a href="#">Детали</a>
Иванова Анастасия ...	Шевчук Максим Сер...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	06.06.2020	13:00	обрабатывается	<a href="#">Детали</a>
Ангелина	Петр	Россия, Ульяновск, ...	Россия, Ульяновск, ...	07.06.2020	8:30	обрабатывается	<a href="#">Детали</a>

[Совершить заказ](#)  
[Изменить почту или пароль](#)

Рисунок 15 – Страница личного кабинета после совершения заказа

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						49
Изм.	Лист	№ докум.	Подп.	Дата		

По нажатию кнопки «Детали» заказчик может посмотреть подробную информацию о заказе (рисунок 16).

## Детали

Заказ

Отправитель	Ангелина
Получатель	Петр
Адрес отправителя	Россия, Ульяновск, улица Розы Люксембург, 34
Район отправки	Ленинский
Адрес получателя	Россия, Ульяновск, Камышинская улица, 4
Район получения	Засвияжский
Номер отправителя	89339994455
Номер получателя	81112223344
Дата	07.06.2020
Время	8:30
Вес, кг	4
Длина, см	100
Ширина, см	50
Высота, см	50
Статус	обрабатывается
Негабаритный	<input type="checkbox"/>
Хрупкое	<input type="checkbox"/>
Оплата отправителем	<input checked="" type="checkbox"/>
Цена	356
CustomerId	1
CourierId	

[Вернуться в личный кабинет](#)

Рисунок 16 – Страница подробной информации о заказе

После получения заказа, на странице подробной информации о заказе, заказчик может выставить рейтинг курьеру, нажав кнопку «Оценить». (рисунок 17).

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						50
Изм.	Лист	№ докум.	Подп.	Дата		

### Детали

#### Заказ

Отправитель	Евгений
Получатель	Дарья
Адрес отправителя	Россия, Ульяновск, проспект
Район отправки	Железнодорожный
Адрес получателя	Россия, Ульяновск, улица Б
Район получения	Заволжский
Номер отправителя	89098087766
Номер получателя	87654321234
Дата	07.06.2020
Время	8:00
Вес, кг	1
Длина, см	50
Ширина, см	30
Высота, см	22
Статус	получен
Негабаритный	<input type="checkbox"/>
Хрупкое	<input type="checkbox"/>
Оплата отправителем	<input checked="" type="checkbox"/>
Цена	120
CustomerId	9
CourierId	4

[Редактировать](#) | [Назад](#)

[Оценить](#)

Оценка

Текущий рейтинг: 0

★

★

★

★


★

Оценить

Рисунок 17 – Выставление рейтинга курьеру

#### 4.5.3.2 Для роли «Администратор»

Авторизация администратором проходит аналогично, как и для роли «Заказчик». После прохождения авторизации, главная страница выглядит, как на рисунке 18.



[Главная](#) | [О нас](#) | [Расчет стоимости доставки](#) | [Пользователи](#) | [Заказы](#) | [Выйти](#)

## СРОЧНАЯ ДОСТАВКА ПОСЫЛОК

ДОКУМЕНТОВ И ГРУЗОВ С ГАРАНТИЕЙ ДОСТАВКИ ОТПРАВЛЕНИЯ ПОЛУЧАТЕЛЮ

Бесплатный вызов курьера

Доставка по городу на следующий день после подачи заявки

Личный кабинет для управления услугами

Рисунок 18 – Главная страница администратора

Пройдя по ссылке «Пользователи», отображается таблица со всеми пользователями, зарегистрированными в системе (рисунок 19). По нажатию кнопки «Удалить» администратор может удалить пользователя (рисунок 20).



#### Пользователи

Роль	Почта	
customer	user@mail.com	<a href="#">Детали</a>   <a href="#">Удалить</a>
admin	admin@mail.com	<a href="#">Детали</a>   <a href="#">Удалить</a>
courier	courier@mail.com	<a href="#">Детали</a>   <a href="#">Удалить</a>
courier	courier2@mail.com	<a href="#">Детали</a>   <a href="#">Удалить</a>
courier	courier3@mail.com	<a href="#">Детали</a>   <a href="#">Удалить</a>
courier	courier4@mail.com	<a href="#">Детали</a>   <a href="#">Удалить</a>
courier	courier5@mail.com	<a href="#">Детали</a>   <a href="#">Удалить</a>
courier	newcour@mail.com	<a href="#">Детали</a>   <a href="#">Удалить</a>

Рисунок 19 – Страница списка всех зарегистрированных пользователей

## Удалить

### Вы уверены, что хотите удалить?

Пользователь

**Роль** courier  
**Почта** newcour@mail.com

[Удалить](#) | [Назад](#)

Рисунок 20 – Страница удаления пользователя

Пройдя по ссылке «Заказы», на главной странице отображается таблица со всеми заказами, зарегистрированными в системе (рисунок 21). По нажатию кнопки «Детали» администратор может посмотреть подробности заказа (рисунок 22). По нажатию кнопки «Удалить» администратор может удалить заказ (рисунок 23).



## Заказы

Распределить заказы

Фильтрация: все ▼ Фильтровать

Отправитель	Получатель	Адрес отправителя	Адрес получателя	Дата	Время	Статус	
Иванова Анастасия ...	Шевчук Максим Сер...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	06.06.2020	8:00	обрабатывается	<a href="#">  Детали</a>   <a href="#">Удалить</a>
Иванова Анастасия ...	Шевчук Максим Сер...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	06.06.2020	13:00	обрабатывается	<a href="#">  Детали</a>   <a href="#">Удалить</a>
Ангелина	Петр	Россия, Ульяновск, ...	Россия, Ульяновск, ...	07.06.2020	8:30	обрабатывается	<a href="#">  Детали</a>   <a href="#">Удалить</a>

[сохранить отчет](#)

Рисунок 21 – Страница списка всех зарегистрированных заказов

## Детали

### Заказ

<b>Отправитель</b>	Ангелина
<b>Получатель</b>	Петр
<b>Адрес отправителя</b>	Россия, Ульяновск, улица Розы Люксембург, 34
<b>Район отправки</b>	Ленинский
<b>Адрес получателя</b>	Россия, Ульяновск, Камышинская улица, 4
<b>Район получения</b>	Засвияжский
<b>Номер отправителя</b>	89339994455
<b>Номер получателя</b>	81112223344
<b>Дата</b>	07.06.2020
<b>Время</b>	8:30
<b>Вес, кг</b>	4
<b>Длина, см</b>	100
<b>Ширина, см</b>	50
<b>Высота, см</b>	50
<b>Негабаритный</b>	<input type="checkbox"/>
<b>Хрупкое</b>	<input type="checkbox"/>
<b>Оплата отправителем</b>	<input checked="" type="checkbox"/>
<b>Цена</b>	356
<b>CustomerId</b>	1
<b>CourierId</b>	

Рисунок 22 – Страница подробной информации о заказе

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						53
Изм.	Лист	№ докум.	Подп.	Дата		

# Удалить

Вы уверены, что хотите удалить этот заказ?

**Отправитель** Иванова Анастасия Петровна  
**Получатель** Шевчук Максим Сергеевич  
**Адрес отправителя** Россия, Ульяновск, улица Докучаева, 18  
**Адрес получателя** Россия, Ульяновск, Октябрьская улица, 22  
**Номер отправителя** 84758494565  
**Номер получателя** 86543456677  
**Дата** 06.06.2020  
**Время** 8:00  
**Вес, кг** 2  
**Длина, см** 50  
**Ширина, см** 12  
**Высота, см** 20  
**Оплата отправителем** ☐  
**Цена** 98  
**CourierId** 3  
**CustomerId** 1

[Удалить](#) | [Назад](#)

Рисунок 23 – Страница удаления заказа

После успешного удаления заказа администратором, страница со всеми заказами выглядит, как на рисунке 24.

## Заказы

[Распределить заказы](#)

Фильтрация:

Отправитель	Получатель	Адрес отправителя	Адрес получателя	Дата	Время	Статус	
Иванова Анастасия ...	Шевчук Максим Сер...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	06.06.2020	13:00	обрабатывается	<a href="#">  Детали   Удалить</a>
Ангелина	Петр	Россия, Ульяновск, ...	Россия, Ульяновск, ...	07.06.2020	8:30	обрабатывается	<a href="#">  Детали   Удалить</a>

[сохранить отчет](#)

Рисунок 24 – Страница списка всех зарегистрированных заказов после удаления одного заказа

На странице заказов администратор может получить отчет о всех совершенных заказах, нажав на кнопку «сохранить отчет». После чего отчет будет сохранен и доступен в формате \*.xml (рисунок 25-26).

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						54
Изм.	Лист	№ докум.	Подп.	Дата		

	A	B	C	D	E	F	G	H	I	J	K
1	Отправитель	Получатель	Адрес отпра	Район отпра	Адрес полу	Район полу	Номер отпра	Номер полу	Дата	Время	Статус
2	Иванова Ана	Шевчук Мак	Россия, Уль	Ленинский	Россия, Уль	Засвияжский	84758494567	89475869384	2020-06-06	13:00	обрабатывае
3	Ангелина	Петр	Россия, Уль	Ленинский	Россия, Уль	Засвияжский	89339994455	81112223344	2020-06-07	08:30	обрабатывае

Рисунок 25 – Отчет в формате \*.xml

K	L	M	N	O	P	Q	R	S	T
Статус	Вес	Длина	Ширина	Высота	Негабаритны	Хрупкий	Оплата отпра	Цена	ID пользователя
обрабатывае	2	50	12	22	ЛОЖЬ	ЛОЖЬ	ИСТИНА	49	1
обрабатывае	4	100	50	50	ЛОЖЬ	ЛОЖЬ	ИСТИНА	356	1

Рисунок 26 – Отчет в формате xml (продолжение)

Данные для отчета можно отфильтровать по дате (все, за последний месяц и неделю). При выборе варианта «все» на странице отображаются все заказы. При выборе «неделя» и «месяц» отображаются все заказы за последнюю неделю и месяц соответственно (рисунок 27-28).

#### Заказы

Распределить заказы

Фильтрация: неделя Фильтровать

Отправитель	Получатель	Адрес отправителя	Адрес получателя	Дата	Время	Статус	
Иванова Анастасия ...	Шевчук Максим Сер...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	06.06.2020	13:00	обрабатывается	<a href="#">Детали</a>   <a href="#">Удалить</a>
Ангелина	Петр	Россия, Ульяновск, ...	Россия, Ульяновск, ...	07.06.2020	8:30	обрабатывается	<a href="#">Детали</a>   <a href="#">Удалить</a>

сохранить отчет

Рисунок 27 – Страница при выборе фильтра «неделя»

#### Заказы

Распределить заказы

Фильтрация: все Фильтровать

Отправитель	Получатель	Адрес отправителя	Адрес получателя	Дата	Время	Статус	
Иванова Анастасия ...	Шевчук Максим Сер...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	06.06.2020	13:00	обрабатывается	<a href="#">Детали</a>   <a href="#">Удалить</a>
Ангелина	Петр	Россия, Ульяновск, ...	Россия, Ульяновск, ...	07.06.2020	8:30	обрабатывается	<a href="#">Детали</a>   <a href="#">Удалить</a>
Юлия	Николай	Россия, Ульяновск, ...	Россия, Ульяновск, ...	08.05.2020	11:00	обрабатывается	<a href="#">Детали</a>   <a href="#">Удалить</a>

сохранить отчет

Рисунок 28 – Страница при выборе фильтра «месяц»

Также администратору доступно распределение заказов по курьерам. Для этого на странице заказов необходимо нажать кнопку «Распределить заказы». Открывается страница распределения, где отображаются заказы, к которым пока не прикреплен курьер (рисунок 29).

Есть также возможность фильтрации заказов по районам (рисунок 30).

## Распределение

 Фильтрация:  

Отправитель	Получатель	Адрес отправителя	Адрес получателя	Дата	Время	Статус	
Иванова Анастасия ...	Шевчук Максим Сер...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	06.06.2020	13:00	обрабатывается	<a href="#">Назначить курьера</a>
Ангелина	Петр	Россия, Ульяновск, ...	Россия, Ульяновск, ...	07.06.2020	8:30	обрабатывается	<a href="#">Назначить курьера</a>

Рисунок 29 – Страница распределения заказов по курьерам

## Распределение

 Фильтрация:  

Отправитель	Получатель	Адрес отправителя	Адрес получателя	Дата	Время	Статус	
Иванова Анастасия ...	Шевчук Максим Сер...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	06.06.2020	13:00	обрабатывается	<a href="#">Назначить курьера</a>
Ангелина	Петр	Россия, Ульяновск, ...	Россия, Ульяновск, ...	07.06.2020	8:30	обрабатывается	<a href="#">Назначить курьера</a>

Рисунок 30 – Варианты фильтрации заказов по районам

По нажатию кнопки «Назначить курьера» администратор может выбрать необходимого курьера из выпадающего списка, чтобы назначить данного курьера на заказ (рисунок 31).

## Редактировать

### Заказ

CourierId


[Назад](#)

Рисунок 31 – Назначение курьера на заказ

По нажатию кнопки «Сохранить» страница распределения заказов выглядит как на рисунке 32.

## Распределение

Фильтрация:

Отправитель	Получатель	Адрес отправителя	Адрес получателя	Дата	Время	Статус	
Ангелина	Петр	Россия, Ульяновск, ...	Россия, Ульяновск, ...	07.06.2020	8:30	обрабатывается	<a href="#">Назначить курьера</a>

Рисунок 32 – Страница распределения заказов после назначения курьера на первый заказ

### 4.5.3.3 Для роли «Курьер»

Авторизация и смена почты и пароля курьером проходит аналогично, как и для роли «Заказчик».

Если курьер не авторизован, то нажав на кнопку «Зарегистрироваться» перед ним откроется поле ввода информации для регистрации, где нужно ввести почту, пароль, повторить пароль для регистрации и отметить галочку у пункта «Курьер» (рисунок 33). Далее необходимо ввести дополнительную информацию (рисунок 34). При вводе полей без ошибок, курьер успешно регистрируется.

## Регистрация

Почта	<input type="text" value="courier2@mail.com"/>
Пароль	<input type="password" value="....."/>
Подтверждение пароля	<input type="password" value="....."/>
Курьер <input checked="" type="checkbox"/>	
<input type="button" value="Зарегистрироваться"/>	

Рисунок 33 – Страница регистрации курьера

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						57
Изм.	Лист	№ докум.	Подп.	Дата		

## Введите данные

Курьер

Имя

Район

Газель ☐

[Назад](#)

Рисунок 34 – Страница заполнения дополнительных данных

Оказавшись на главной странице системы, курьер может сразу перейти в личный кабинет по ссылке в верхнем правом углу (рисунок 35).

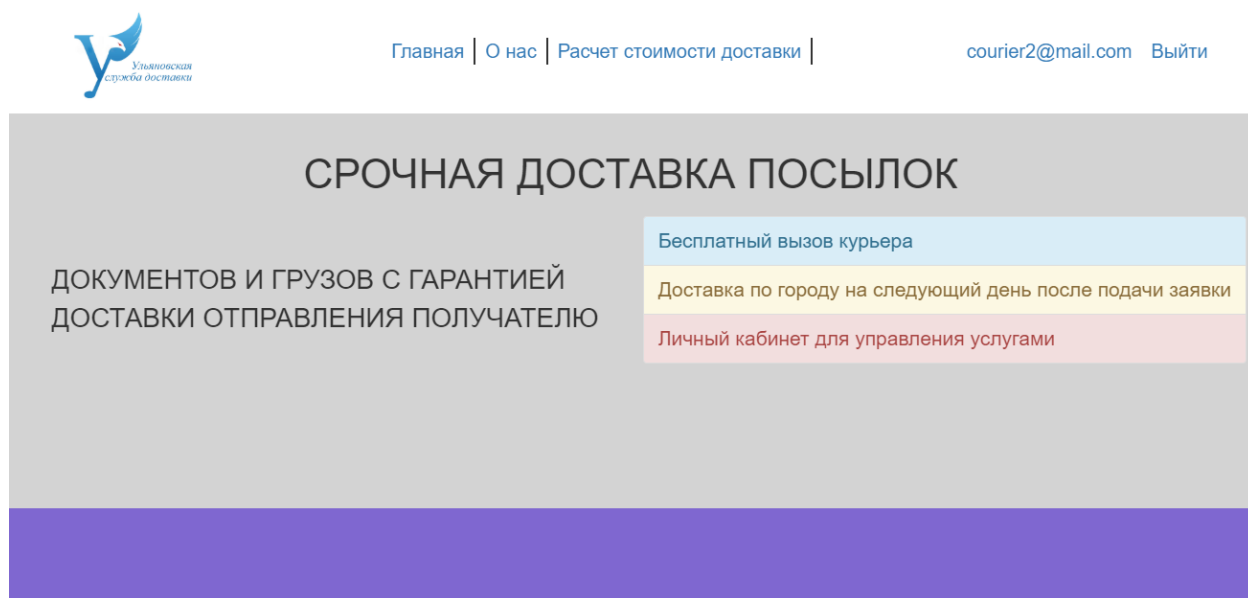


Рисунок 35 – Главная страница курьера

В личном кабинете отображаются все заявки, закрепленные за данным курьером (рисунок 36). Если отметить галочку у пункта «Заказы на сегодня» и нажатии кнопки «Фильтровать» в личном кабинете будут отображены заказы только на сегодняшний день (рисунок 37).

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						58
Изм.	Лист	№ докум.	Подп.	Дата		



## Личный кабинет

Заказы на сегодня: ☐ [Фильтровать](#)

Заказы							
Отправитель	Получатель	Адрес отправителя	Адрес получателя	Дата	Время	Статус	
Иванов Иван Ивано...	Васильев Александр...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	09.06.2020	12:00	обрабатывается	<a href="#">Детали</a>
Евгений	Дарья	Россия, Ульяновск, ...	Россия, Ульяновск, ...	07.06.2020	8:00	обрабатывается	<a href="#">Детали</a>

[Изменить почту или пароль](#)[сохранить отчет](#)

Рисунок 36 – Личный кабинет курьера



## Личный кабинет

Заказы на сегодня: ☒ [Фильтровать](#)

Заказы							
Отправитель	Получатель	Адрес отправителя	Адрес получателя	Дата	Время	Статус	
Евгений	Дарья	Россия, Ульяновск, ...	Россия, Ульяновск, ...	07.06.2020	8:00	обрабатывается	<a href="#">Детали</a>

[Изменить почту или пароль](#)[сохранить отчет](#)

Рисунок 37 – Личный кабинет после применения фильтра

Нажатие кнопки «сохранить отчет» сохраняет отчет аналогично роли «Администратор».

По нажатию кнопки «Детали» курьер может посмотреть подробности заказа (рисунок 38).

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						59
Изм.	Лист	№ докум.	Подп.	Дата		

## Детали

### Заказ

Отправитель	Евгений
Получатель	Дарья
Адрес отправителя	Россия, Ульяновск, проспект Гая, 95
Район отправки	Железнодорожный
Адрес получателя	Россия, Ульяновск, улица Врача Михайлова, 52
Район получения	Заволжский
Номер отправителя	89098087766
Номер получателя	87654321234
Дата	07.06.2020
Время	8:00
Вес, кг	1
Длина, см	50
Ширина, см	15
Высота, см	22
Статус	обрабатывается
Негабаритный	<input type="checkbox"/>
Хрупкое	<input type="checkbox"/>
Оплата отправителем	<input checked="" type="checkbox"/>
Цена	120
CustomerId	9
CourierId	4

[Редактировать](#) | [Назад](#)

Рисунок 38 – Страница подробной информации о заказе

Далее при необходимости корректировки данных заказа необходимо нажать кнопку «Редактировать». На данной странице курьер может изменять все данные, в том числе менять статуса заказа, параметры груза и делать перерасчет стоимости доставки (рисунок 39).

Дата	<input type="text" value="07.06.2020"/>
Время	<input type="text" value="8:00"/>
Статус	<input type="text" value="отправлен"/>
Вес, кг	<input type="text" value="1"/>
Длина, см	<input type="text" value="50"/>
Ширина, см	<input type="text" value="30"/>
Высота, см	<input type="text" value="22"/>
Оплата отправителем	<input checked="" type="checkbox"/>
Цена	<input type="text" value="120"/>
CustomerId	<input type="text" value="9"/>
CourierId	<input type="text" value="Второй"/>
<input type="button" value="Рассчитать"/>	
<input type="button" value="Сохранить"/>	

[Назад](#)

Рисунок 39 – Страница редактирования заказа

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						60
Изм.	Лист	№ докум.	Подп.	Дата		



После изменения ширины груза и смены статуса заказа, подробная информация о заказе обновляется (рисунок 40).

## Детали

### Заказ

Отправитель	Евгений
Получатель	Дарья
Адрес отправителя	Россия, Ульяновск, проспект Гая, 95
Район отправки	Железнодорожный
Адрес получателя	Россия, Ульяновск, улица Врача Михайлова, 52
Район получения	Заволжский
Номер отправителя	89098087766
Номер получателя	87654321234
Дата	07.06.2020
Время	8:00
Вес, кг	1
Длина, см	50
Ширина, см	30
Высота, см	22
Статус	отправлен
Негабаритный	<input type="checkbox"/>
Хрупкое	<input type="checkbox"/>
Оплата отправителем	<input checked="" type="checkbox"/>
Цена	120
CustomerId	9
CourierId	4

[Редактировать](#) | [Назад](#)

Рисунок 40 – Страница подробной информации о заказе после редактирования

После проверки данных курьером и смены статуса заказа, появляется возможность печати информации о заказе (рисунок 41).

## Личный кабинет

Заказы на сегодня: ☐ [Фильтровать](#)

Заказы							
Отправитель	Получатель	Адрес отправителя	Адрес получателя	Дата	Время	Статус	
Иванов Иван Ивано...	Васильев Алексан...	Россия, Ульяновск, ...	Россия, Ульяновск, ...	09.06.2020	12:00	обрабатывается	<a href="#">Детали</a>
Евгений	Дарья	Россия, Ульяновск, ...	Россия, Ульяновск, ...	07.06.2020	8:00	отправлен	<a href="#">Детали</a>   <a href="#">Печать</a>

[Изменить почту или пароль](#)

[сохранить отчет](#)

Рисунок 41 – Отображение доступности печати

По нажатию кнопки «Печать» происходит сохранение документа в формате \*.pdf (рисунок 42).

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						61
Изм.	Лист	№ докум.	Подп.	Дата		

Отправитель		
ФИО	Телефон	Адрес
Евгений	89098087766	Россия, Ульяновск, проспект Гая, 95

Получатель		
ФИО	Телефон	Адрес
Дарья	87654321234	Россия, Ульяновск, улица Врача Михайлова, 52

Груз					
Вес	Длина	Высота	Ширина	Оплата получателем	Цена
1	50	22	30	да	120

Я подтверждаю, что информация в накладной является полной и точной. С основными условиями пересылки я ознакомлен(а).

Подпись заказчика \_\_\_\_\_

Подпись курьера \_\_\_\_\_



Рисунок 42 – Печатная информация о заказе

Информация о заказе содержит также QR-код. Отсканировав его, можно получить подробную информацию о заказе с мобильного устройства (рисунок 43).

MegaFon 21:24 84%

192.168.1.8:3000/Orde

### Детали

#### Заказ

Отправитель	Евгений
Получатель	Дарья
Адрес отправителя	Россия, Ульяновск, проспект Гая, 95
Район отправки	Железнодорожный
Адрес получателя	Россия, Ульяновск, улица Врача Михайлова, 52
Район получения	Заволжский
Номер отправителя	89098087766
Номер получателя	87654321234
Дата	07.06.2020
Время	8:00
Вес, кг	1
Длина, см	50
Ширина, см	30
Высота, см	22
Статус	получен
Негабаритный	<input type="checkbox"/>
Хрупкое	<input type="checkbox"/>
Оплата отправителем	<input checked="" type="checkbox"/>
Цена	120
CustomerId	9
CourierId	4

Оценить

Рисунок 43 – Информация с QR-кода

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						62
Изм.	Лист	№ докум.	Подп.	Дата		

После доставки груза до получателя и смене курьером статуса на «получен» появляется возможность оценить заказчика. Процесс оценивания аналогичен процессу в роли «Заказчик».

#### 4.5.3.3 Для всех пользователей системы

Для всех пользователей системы доступна возможность предварительного расчета стоимости доставки. Для этого необходимо перейти на вкладку «Расчет стоимости доставки», заполнить поля и нажать кнопку «Рассчитать» (рисунок 44).

### Расчет

Заказ

Адрес отправителя	Россия, Ульяновск, улица Докучаева, 18
Район отправки	Ленинский
Адрес получателя	Россия, Ульяновск, Октябрьская улица, 22
Район получения	Засвияжский
Вес, кг	2
Длина, см	50
Ширина, см	15
Высота, см	22
Цена	53

53р. Это примерная стоимость доставки. Окончательная установится после взвешивания груза

Рассчитать

Рисунок 44 – Предварительный расчет стоимости доставки

#### 4.5.4 Исключительные ситуации и их обработка

При неверных действиях пользователей, неверных форматах или недопустимых значениях входных данных, система выдает пользователю соответствующие сообщения, после чего возвращается в рабочее состояние,

предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

Ниже приведены примеры (рисунок 45-51).

## Вход

The screenshot shows a login form titled "Вход". It has two input fields: "Почта" (Email) containing "nastya" and "Пароль" (Password) containing three dots. Below the email field, a red error message reads "Некорректный адрес". Below the password field, a red error message reads "Поле Пароль должно иметь тип строки или массива с минимальной длиной '6'". A "Войти" (Login) button is at the bottom right of the form.

[Назад](#)

Рисунок 45 – Некорректный ввод данных при авторизации

## Вход

The screenshot shows a login form titled "Вход". Above the input fields, a red error message reads "Пользователя с таким логином и паролем нет". The "Почта" (Email) field contains "nastya@ya.ru" and the "Пароль" (Password) field contains six dots. A "Войти" (Login) button is at the bottom right of the form.

[Назад](#)

Рисунок 46 – Авторизация с неверным логином и паролем

## Регистрация

The screenshot shows a registration form titled "Регистрация". It has three input fields: "Почта" (Email) containing "nastya@mail.com", "Пароль" (Password) containing six dots, and "Подтверждение пароля" (Confirm password) containing six dots. Below the confirm password field, a red error message reads "Пароли не совпадают". There is a checkbox labeled "Курьер" which is unchecked. A "Зарегистрироваться" (Register) button is at the bottom right of the form.

Рисунок 47 – Несовпадение паролей при регистрации

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						64
Изм.	Лист	№ докум.	Подп.	Дата		

## Регистрация

- Пользователь с таким логином уже существует

Почта courier2@mail.com

Пароль .....

Подтверждение пароля .....

Курьер ☒

Зарегистрироваться

Рисунок 48 – Регистрация уже существующего логина

## Подача заявки

Отправитель

A

Поле Отправитель должно иметь тип строки или массива с минимальной длиной "2".

Получатель

Дарья

Адрес  
отправителя

Россия, Республика Татарстан, Казань, Вахитовский район, улица Татарстан, 7

Неверный адрес

Район отправки

Вахитовский

Адрес получателя

Россия, Ульяновск, Московское шоссе

Неполный адрес, требуется уточнение

Район получения

Номер  
отправителя

456789

Некорректный номер телефона.

Номер получателя

89138475849

Дата

07.06.2020

Неверная дата

Рисунок 49 – Некорректный ввод данных при подаче заявки

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						65
Изм.	Лист	№ докум.	Подп.	Дата		

Время

Вес, кг   
Поле Вес, кг должно иметь значение между 0 и 10.

Длина, см   
Поле Длина, см должно иметь значение между 0 и 200.

Ширина, см   
Значением поля Ширина, см должно быть число.

Высота, см   
Поле Высота, см должно иметь значение между 0 и 200.

Негабаритный ☐

Хрупкое ☐

Оплата отправителем ☐

Цена

Рисунок 50 – Некорректный ввод данных при подаче заявки (продолжение)

Вес, кг   
Поле

Длина, см   
Поле

Ширина, см

Высота, см

Негабаритный ☐

Хрупкое ☐

Уведомление от сайта localhost

Введите адреса

Рисунок 51 – Расчет стоимости доставки без выбора адреса

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						66
Изм.	Лист	№ докум.	Подп.	Дата		

## 5 ТЕСТИРОВАНИЕ СИСТЕМЫ

### 5.1 Условия и порядок тестирования

Объектом тестирования является прикладное программное обеспечение –автоматизированная система курьерской службы.

Методом тестирования была выбрана модель черного ящика.

Общий порядок тестирования:

- снимаются входные данные;
- снимаются выходные данные.

Тип проверки функционирования системы – ручной.

### 5.2 Исходные данные для контрольных примеров

#### 5.2.1 Описание ситуаций для элементов системы

Для тестирования валидации электронной почты попробуем ввести почту «nastya» (неправильная), а потом «nastya@mail.com» (правильная). Для тестирования работоспособности авторизации, пробуем авторизоваться с помощью почты «nastya@mail.com» и пароля «123477» (неправильный), а потом пароль «123456» (правильный).

### 5.3 Результаты тестирования

Корректная работа описанных выше исходных данных представлена в п.4.5.3 Руководства пользователя. Соответственно в остальных случаях обращаться к п.4.5.4.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						67
Изм.	Лист	№ докум.	Подп.	Дата		

## 6 ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

### 6.1 Определение трудоемкости разработки системы

Ожидаемая продолжительность работы  $q$  рассчитывается как математическое ожидание для  $\beta$  - распределения. В этом случае, ожидаемая продолжительность работ вычисляется по формуле:

$$q = \frac{3T_{min} + 2T_{max}}{5},$$

где  $T_{min}$  – минимальная продолжительность работы;

$T_{max}$  – максимальная продолжительность работы.

Перечень работ и трудоемкость их выполнения приведены в таблице 20.

Таблица 20 – Трудоемкость работ по разработке системы

№	Наименование рабочих процессов	Трудоемкость (чел./ч)
1	Формирование требований к системе	$\frac{(3 * 10) + (2 * 15)}{5} = 12$
2	Разработка технического задания	$\frac{(3 * 15) + (2 * 25)}{5} = 19$
3	Моделирование структуры и функционала	$\frac{(3 * 20) + (2 * 40)}{5} = 28$
4	Реализация программного решения	$\frac{(3 * 122) + (2 * 185)}{5} = 164$
5	Отладка и тестирование системы	$\frac{(3 * 25) + (2 * 65)}{5} = 41$
6	Написание рабочей документации	$\frac{(3 * 50) + (2 * 70)}{5} = 58$
7	Сдача проекта	$\frac{(3 * 10) + (2 * 20)}{5} = 14$
	<b>Итого</b>	<b>336</b>

Процесс реализации программного решения будет наиболее трудоемким при разработке.



## 6.2 Затраты на разработку системы

Затраты на разработку системы (смета затрат) – сумма всех затрат на производство и реализацию продукции, составленная по экономическим элементам расходов.

Расчет затрат осуществляется через составление общей сметы, которая включает в себя следующие статьи:

- затраты на материальные ресурсы;
- затраты на электроэнергию;
- затраты на оплату труда;
- затраты на обязательные взносы;
- амортизация основных фондов.

### 6.2.1 Затраты на материальные ресурсы

Расчет затрат приведен в таблице 21.

Таблица 21 – Затраты на материальные ресурсы

№	Наименование	Количество израсходованного материала	Цена за единицу, руб	Сумма, руб
1	Ноутбук	1 шт.	30 990	30 990
2	Принтер	1 шт.	6 500	6 500
Итого затраты на материальные ресурсы				37 490

### 6.2.2 Затраты на электроэнергию

Общая сумма затрат на электроэнергию  $Z_э$  рассчитывается по формуле:

$$Z_э = \sum_{i=1}^n M_i \times K_i \times T_i \times Ц,$$

где  $i$  – вид электрооборудования;

$n$  – количество электрооборудования;

$M_i$  – паспортная мощность  $i$ -го электрооборудования, кВт;

$K_i$  – коэффициент использования мощности  $i$ -го электрооборудования (принимается  $K_i = 0.7 \div 0.9$ );

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						69
Изм.	Лист	№ докум.	Подп.	Дата		

$T_i$  – время работы  $i$ -го оборудования за весь период разработки АИС, ч;

$\Pi$  – цена электроэнергии, руб/кВт·ч.

Исходные данные и результаты расчетов представлены в таблице 22.

Таблица 22 – Затраты на электроэнергию

Наименование оборудования	$M_i$ , кВт	$K_i$	$T_i$ , ч	$\Pi$ , руб/кВт·ч	Сумма, руб
Ноутбук	0,07	0,8	336	3,77	70,94
Искусственное освещение	0,04		124		14,95
Итого затраты на электроэнергию					85,89

### 6.2.3 Затраты на оплату труда

Существует повременная и сдельная форма оплаты труда. В первом случае, размер заработной платы зависит от отработанного времени и от квалификации работника. Во втором – от количества произведенной продукции.

При анализе будем считать что программист, разрабатывающий систему имеет среднюю квалификацию. Сумма основных затрат на оплату труда  $Z_{тр}$  определяется по формуле:

$$Z_{тр} = \sum_{i=1}^n \text{ЧС}_i \times T_i ,$$

где  $\text{ЧС}_i$  – часовая ставка  $i$ -го работника, руб.;

$T_i$  – время на разработку системы, ч;

$i$  – порядковый номер работника;

$n$  – количество работников.

Результаты расчета приведены в таблице 23.

Таблица 23 – Затраты на оплату труда

№	Категория работника	Трудоемкость разработки, чел./ч	Часовая ставка, руб./ч	Сумма, руб
1	Разработчик системы	336	100	33 600
Итого затраты на основную оплату труда				33 600

## 6.2.4 Затраты на обязательные взносы

Отчисления на социальные нужды – обязательные отчисления по нормам, установленным законодательством государственного социального страхования, в Фонд социального страхования РФ, Пенсионный фонд РФ, фонды обязательного медицинского страхования от затрат на оплату труда работников, включаемых в себестоимость продукции (работ, услуг), по элементу «Затраты на оплату труда» (кроме тех видов оплаты, на которые страховые взносы не начисляются).

Результаты расчета обязательных взносов приведены в таблице 24.

Таблица 24 – Отчисления на обязательные взносы

Вид	Отчисления, %
Пенсия:	16
– страховое	6
– накопительное	
Социальные выплаты(ФСС)	2,9
Медицинское страхование (ФФОМС)	5,1
Взносы на страхование от несчастных случаев	0,2
Итого обязательных взносов	30,2

Используя рассчитанную ранее заработную плату, рассчитаем размер отчислений в социальные фонды по формуле:

$$\text{ЕСН} = \frac{30,2\%}{100} \% \cdot \text{ЗП} = 0,302 \cdot 33600 = 10147,2 \text{ (руб.)},$$

где ЕСН – размер единого социального налога в рублях;

ЗП – итоговая заработная плата сотрудника в рублях.

## 6.2.5 Затраты на амортизационные отчисления

Амортизационные отчисления – средства в виде денег, направляемых на ремонт или строительство, изготовление новых основных средств. Сумма амортизационных отчислений включается в издержки производства (себестоимость) продукции и тем самым переходит в цену.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						71
Изм.	Лист	№ докум.	Подп.	Дата		

При расчете амортизационных вычислений используется линейный способ. Расходы на амортизацию  $P_a$  рассчитываются по формуле:

$$P_a = \sum_{i=1}^n \frac{C_i \times H_{ai} \times T_{pi}}{T_{\varepsilon i}},$$

где  $C_i$  – стоимость  $i$ -го оборудования, руб.;

$H_{ai}$  – годовая норма амортизации  $i$ -го оборудования, % доля;

$T_{pi}$  – время работы  $i$ -го оборудования за весь период разработки, ч;

$T_{\varepsilon i}$  – эффективный фонд времени работы  $i$ -го оборудования за год, ч/год (40-часовая рабочая неделя на 52 недели/год, кроме праздников);

$i$  – вид оборудования;

$n$  – количество оборудования.

Расходы на амортизацию оборудования (ноутбука и принтера) составят:

$$P_a = (30\,990 \times 0,2 + 6\,500 \times 0,1) \frac{336}{2000} = 1\,150,464 (\text{руб.})$$

## 6.2.6 Расчет прочих затрат

В статью «Прочие затраты» включаются расходы на канцелярские товары. Расчет затрат приведен в таблице 25

Таблица 25 – Расходы на прочие затраты

Наименование	Единица измерения	Количество израсходованного материала	Цена за единицу, руб	Сумма, руб
Пачка бумаги формата А4	шт.	1	260	260
Канцелярские принадлежности	шт.	1	350	350
Оплата услуг интернет-провайдера	руб/мес	2	350	700
Итого прочие затраты				1 310

### 6.3 Себестоимость проекта

Итоговый расчет себестоимости проекта приведен в таблице 26.

Таблица 26 – Себестоимость проекта

№	Статьи затрат	Сумма, руб
1	Затраты на материальные ресурсы	37 490
2	Затраты на электроэнергию	85,89
3	Затраты на оплату труда	33 600
4	Затраты на обязательные взносы	10 147,2
5	Амортизация основных фондов	1 150,464
6	Прочие затраты	1 310
Итого по смете		83 783,554

### 6.4 Определение экономической эффективности проекта

После расчетов совокупности всех затрат необходимо рассчитать прибыль, которую получит предприятие от разработки данной системы.

Экономическая эффективность – результативность экономической системы, выражающаяся в отношении полезных конечных результатов ее функционирования к затраченным ресурсам. Она определяется как разность между годовой экономией (или годовым приростом) и нормативной прибылью.

Повышение производительности труда высчитывается по формуле:

$$Pi = \frac{\Delta Ti * 100\%}{(Ti - \Delta Ti)},$$

где  $Ti$  - временные затраты до автоматизации, мин.;

$\Delta Ti$  – экономия времени (разница затрат после и до автоматизации), мин.;

Прирост производительности труда представлен в таблице 27.

Таблица 27 – Прирост производительности труда

Вид работ	До автоматизации Т <sub>і</sub> , мин	После автоматизации (Т <sub>і</sub> – ΔТ <sub>і</sub> ), мин	Экономия ΔТ <sub>і</sub> , мин	Повышение производительности труда Р <sub>і</sub> , %
Расчет стоимости доставки	10	1	9	900
Подача заявки	15	5	10	200
Просмотр и составление отчетности о работе	15	3	12	400

Учитывая налог на прибыль (ИТ-сфера), составляющий 13%, среднегодовая заработная плата составит:

$$З_{г} = 33\ 600 - 33\ 600 * 0,13 = 29\ 232 \text{ (руб.)}$$

Годовой прирост высчитывается по формуле:

$$П = З_{г} * \sum_{i=0}^n \left( \frac{P_i}{100\%} \right),$$

где  $З_{г}$  – среднегодовая заработная плата пользователя, руб.;

$P_i$  – повышение производительности труда, %.

$$П = 29\ 232 * (9 + 2 + 4) = 438\ 480 \text{ (руб.)}$$

Годовой экономический эффект Э (руб), рассчитывается по формуле:

$$Э = П - КЕ_{н},$$

где  $П$  – годовая экономия (или годовой прирост), руб.;

$К$  – единовременные затраты (из Таблицы 6), руб;

$E_{н}$  – нормативный коэффициент эффективности (сложившийся в ИТ-отрасли уровень рентабельности  $E_{н} = 0,15$ ).

$$Э = 438\ 480 - 83\ 783,554 * 0,15 = 425\ 912,4669 \text{ руб.}$$

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						74
Изм.	Лист	№ докум.	Подп.	Дата		

Как видно из расчетов, экономический эффект намного превышает затраты на создание системы ( $\Xi > 0$ ).

Расчетный коэффициент экономической эффективности (абсолютная экономическая эффективность) составит:

$$E_p = \frac{\Pi}{K} = \frac{438\,480}{83\,783,554} = 5,233.$$

$E_p > E_n$ , это значит, что капитальные затраты можно считать целесообразными.

Срок окупаемости проекта:

$$T = \frac{K}{\Pi} = \frac{83\,783,554}{438\,480} = 0,191 \text{ лет или } 2 \text{ месяца и } 28 \text{ дней.}$$

## **6.5 Выводы по технико-экономическому анализу и обоснованию проекта разработки**

Произведенный экономический анализ эффективности создания и эксплуатации программного продукта доказывает целесообразность его использования.

Согласно рассчитанным данным о смете затрат была получена себестоимость проекта – 83 783,554руб. Также в ходе проведения разработки проекта вычисленная чистая прибыль (экономический эффект) составила 438 480 руб.

Разрабатываемая система окупается за 2 месяца. Следовательно, организации, принявшие решение применить данную ИС, сократят в будущем затраты на автоматизируемые системой бизнес-процессы.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						75
Изм.	Лист	№ докум.	Подп.	Дата		

## ЗАКЛЮЧЕНИЕ

Выпускная квалификационная работа представляет собой веб-сайт и удовлетворяет всем основным требованиям технического задания. Страницы сайта отображаются корректно.

Были реализованы следующие задачи:

- реализация подсистемы рейтинга заказчиков и курьеров;
- реализация подсистемы расчета стоимости доставки груза и совершения заказа;
- разработка алгоритма получения печатной формы заказа;
- обеспечение возможности получения отчета о заказах;
- реализация подсистемы движения заказа.

Интерфейс системы приятный на глаз и прост в использовании.

Дальнейшее развитие ИС предусматривает возможность более детального подбора курьеров, отправка сообщения о доставке, возможность совершения заказа без регистрации и более надежной системы защиты целостности системы.

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						76
Изм.	Лист	№ докум.	Подп.	Дата		



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Metanit [Электронный ресурс]. – Режим доступа: <https://metanit.com/> – Заглавие с экрана. – (Дата обращения: 25.04.2020).
2. ГОСТ 34.601-90. Автоматизированные системы. Стадии создания. – М.:Изд-во стандартов, 1997. – 5 с.
3. ГОСТ 19.701-90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. – М.: Стандартинформ, 2010. – 8 с.
4. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. – Введ. 1990-01-01. – М.: Издательство стандартов, 2004. – 12 с.
5. ГОСТ 19.401-78. Единая система программной документации. Текст программы. Требования к содержанию и оформлению. – Введ. 1980-01-01. – М.: Стандартинформ, 2010. – 2 с.
6. Родионов, В. В. Выполнение выпускной квалификационной работы бакалавра: учебно-методическое пособие для студентов направления 09.03.02 «Информационные системы и технологии» / В. В. Родионов. – Ульяновск: УлГТУ, 2017. – 74 с.
7. Учебный курс. Создание модели данных Entity Framework для приложения ASP.NET MVC [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/microsoft/blog/133316/> – Заглавие с экрана. – (Дата обращения: 26.04.2020).
8. Изучаем ASP.NET MVC 5 [Электронный ресурс]. – Режим доступа: [https://professorweb.ru/my/ASP\\_NET/mvc/level1/](https://professorweb.ru/my/ASP_NET/mvc/level1/) - Заглавие с экрана. – (Дата обращения: 20.05.2020).
9. Документация по C# [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/>. – Заглавие с экрана. – (Дата обращения: 12.04.2019).

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						77
Изм.	Лист	№ докум.	Подп.	Дата		

10. Как правильно сортировать контент на основе оценок пользователей [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/darudar/blog/143188/>. – Заглавие с экрана. – (Дата обращения: 12.05.2019).
11. Документация API Яндекс.Карт [Электронный ресурс]. – Режим доступа: <https://tech.yandex.ru/maps/jsapi/doc/2.1/quick-start/index-docpage/>. – Заглавие с экрана. – (Дата обращения: 10.04.2019).
12. Обзор C# библиотек для работы с PDF [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/112707/>. – Заглавие с экрана. – (Дата обращения: 9.05.2019).
13. Запись таблиц в PDF [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/articles/25.php>. – Заглавие с экрана. – (Дата обращения: 9.05.2019).
14. Объект Workbook (Excel) [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/office/vba/api/excel.workbook>. – Заглавие с экрана. – (Дата обращения: 10.05.2019).
15. QR Code Generator [Электронный ресурс]. – Режим доступа: <http://goqr.me/api/doc/create-qr-code/>. – Заглавие с экрана. – (Дата обращения: 20.05.2019).
16. Как сделать запрос GET с помощью Visual C# [Электронный ресурс]. – Режим доступа: <https://support.microsoft.com/ru-ru/help/307023/how-to-make-a-get-request-by-using-visual-c>. – Заглавие с экрана. – (Дата обращения: 20.05.2019).

# ПРИЛОЖЕНИЕ А

(обязательное)

## Текст программы

### Файл HomeController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;
using UlskDel.Models;
using Jint;

namespace UlskDel.Controllers
{
    public class HomeController : Controller
    {
        // создаем контекст данных
        OrderContext db = new OrderContext();

        public ActionResult Index()
        {
            IEnumerable<Order> orders = db.Orders;
            // передаем все объекты в динамическое свойство Orders в ViewBag
            ViewBag.Orders = orders;
            return View();
        }

        public ActionResult About()
        {
            return View();
        }

        public ActionResult Count()
        {
            return View();
        }
    }
}
```

### Файл AccountController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;
using UlskDel.Models;

namespace UlskDel.Controllers
{
    public class AccountController : Controller
    {
        OrderContext db = new OrderContext();
        public ActionResult Login()
```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						79
Изм.	Лист	№ докум.	Подп.	Дата		

```

    {
        return View();
    }
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Login(LoginModel model)
    {
        if (ModelState.IsValid)
        {
            string pwd = GetHash(model.Password);
            // поиск пользователя в бд
            User user = db.Users.FirstOrDefault(u => u.Email == model.Name &&
u.Password == pwd);

            if (user != null)
            {
                FormsAuthentication.SetAuthCookie(model.Name, true);
                return RedirectToAction("Index", "Home");
            }
            else
            {
                ModelState.AddModelError("", "Пользователя с таким логином и
паролем нет");
            }
        }
        return View(model);
    }

    public ActionResult Register()
    {
        return View();
    }
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Register(RegisterModel model, bool isCourier = false)
    {
        if (ModelState.IsValid)
        {
            string pwd = GetHash(model.Password);
            User user = db.Users.FirstOrDefault(u => u.Email == model.Name &&
u.Password == pwd);
            int role = 1;
            if (isCourier)
            {
                role = 3;
            }

            if (user == null)
            {
                User x = db.Users.Add(new User { Email = model.Name, Password =
pwd, RoleId = role });
                if (!isCourier)
                {
                    db.Customers.Add(new Customer { Id = x.Id, sumVotes = 0,
totalVotes = 0 });
                }
                db.SaveChanges();
                user = db.Users.Where(u => u.Email == model.Name && u.Password ==
pwd).FirstOrDefault();
                if (user != null)
                {
                    FormsAuthentication.SetAuthCookie(model.Name, true);
                    if (isCourier)
                    {

```

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						80
Изм.	Лист	№ докум.	Подп.	Дата		

```

        return RedirectToAction("Create", "Couriers");
    } else
    {
        return RedirectToAction("Index", "Home");
    }
}
}
else
    ModelState.AddModelError("", "Пользователь с таким логином уже
существует");
}
return View(model);
}
public ActionResult Logoff()
{
    FormsAuthentication.SignOut();
    return RedirectToAction("Index", "Home");
}

private string GetHash(string input)
{
    var md5 = MD5.Create();
    var hash = md5.ComputeHash(Encoding.UTF8.GetBytes(input));
    return Convert.ToBase64String(hash);
}
}
}

```

#### Файл CouriersController.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Data.Entity.Core.Objects;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using UlskDel.Models;

namespace UlskDel.Controllers
{
    public class CouriersController : Controller
    {
        private OrderContext db = new OrderContext();

        // GET: Couriers
        public ActionResult Index()
        {
            var couriers = db.Couriers.Include(c => c.User);
            return View(couriers.ToList());
        }

        public ActionResult Rate(int id)
        {
            Order c = db.Orders.Include(y => y.Customer).FirstOrDefault(x => x.OrderId
== id);
            if (c != null)
                return PartialView(c);
            return HttpNotFound();
        }

        [HttpPost]
    }
}

```

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						81
Изм.	Лист	№ докум.	Подп.	Дата		

```

order) public ActionResult Rate(string Answer, [Bind(Include = "CustomerId")] Order
{
    //Заказчик
    Customer c = db.Customers.FirstOrDefault(x => x.Id == order.CustomerId);
    c.sumVotes = c.sumVotes + Convert.ToInt32(Answer);
    c.totalVotes = c.totalVotes + 1;
    db.Entry(c).State = EntityState.Modified;
    db.SaveChanges();

    User user = db.Users.Where(x => x.Email ==
User.Identity.Name).FirstOrDefault();
    int id = user.Id;
    return RedirectToAction("Details", "Couriers", new { id });
}

// GET: Couriers/Details/5
public ActionResult Details(int? id, bool searchString = false)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Courier courier = db.Couriers.Include(t => t.Orders).FirstOrDefault(t =>
t.Id == id);
    DateTime now = DateTime.Now.Date;
    List<Order> meny_data = db.Orders.Where(x => x.CourierId == id).ToList();
    if (searchString)
    {
        meny_data = db.Orders.Where(t => t.CourierId == id &&
t.Date.CompareTo(now) == 0).ToList();
        //courier = db.Couriers.Include(t => t.Orders.Select(x => x.Date ==
DateTime.Now)).FirstOrDefault(t => t.Id == id);
    }
    ViewData["MyList"] = meny_data;
    //ViewBag.meny_data = meny_data;

    if (courier == null)
    {
        return HttpNotFound();
    }
    return View(courier);
}

// GET: Couriers/Create
public ActionResult Create()
{
    ViewBag.Id = new SelectList(db.Users, "Id", "Email");
    return View();
}

// POST: Couriers/Create
// Чтобы защититься от атак чрезмерной передачи данных, включите определенные
свойства, для которых следует установить привязку. Дополнительные
// сведения см. в статье https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Area, oversize, Name")] Courier
courier)
{
    if (ModelState.IsValid)
    {

```

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						82
Изм.	Лист	№ докум.	Подп.	Дата		

```

        User user = db.Users.Where(x => x.Email ==
User.Identity.Name).FirstOrDefault();
        int id = user.Id;
        courier.Id = id;
        courier.sumVotes = 0;
        courier.totalVotes = 0;
        courier.time = DateTime.Now;
        db.Couriers.Add(courier);
        db.SaveChanges();
        return RedirectToAction("Index", "Home");
    }

    ViewBag.Id = new SelectList(db.Users, "Id", "Email", courier.Id);
    return View(courier);
}

// GET: Couriers/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Courier courier = db.Couriers.Find(id);
    if (courier == null)
    {
        return HttpNotFound();
    }
    ViewBag.Id = new SelectList(db.Users, "Id", "Email", courier.Id);
    return View(courier);
}

// POST: Couriers/Edit/5
// Чтобы защититься от атак чрезмерной передачи данных, включите определенные
свойства, для которых следует установить привязку. Дополнительные
// сведения см. в статье https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
    "Id,sumVotes,totalVotes,time,Area,Name")] Courier courier)
{
    if (ModelState.IsValid)
    {
        db.Entry(courier).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.Id = new SelectList(db.Users, "Id", "Email", courier.Id);
    return View(courier);
}

// GET: Couriers/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Courier courier = db.Couriers.Include(c => c.User).FirstOrDefault(c =>
c.Id ==id);
    if (courier == null)
    {
        return HttpNotFound();
    }
}

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						83
Изм.	Лист	№ докум.	Подп.	Дата		

```

        return View(courier);
    }

    // POST: Couriers/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Courier courier = db.Couriers.Find(id);
        db.Couriers.Remove(courier);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

## Файл CustomersController.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using UlskDel.Models;

namespace UlskDel.Controllers
{
    public class CustomersController : Controller
    {
        private OrderContext db = new OrderContext();

        // GET: Customers
        public ActionResult Index()
        {
            var customers = db.Customers.Include(c => c.User);
            return View(customers.ToList());
        }

        public ActionResult Rate(int id)
        {
            Order c = db.Orders.Include(y => y.Courier).FirstOrDefault(x => x.OrderId
== id);
            if (c != null)
                return PartialView(c);
            return HttpNotFound();
        }

        [HttpPost]
        public ActionResult Rate(string Answer, [Bind(Include = "CourierId")] Order
order)
        {

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						84
Изм.	Лист	№ докум.	Подп.	Дата		



```

        //Курьер заказа
        Courier c = db.Couriers.FirstOrDefault(x => x.Id == order.CourierId);
        c.sumVotes = c.sumVotes + Convert.ToInt32(Answer);
        c.totalVotes = c.totalVotes + 1;
        db.Entry(c).State = EntityState.Modified;
        db.SaveChanges();

        User user = db.Users.Where(x => x.Email ==
User.Identity.Name).FirstOrDefault();
        int id = user.Id;
        return RedirectToAction("Details", "Customers", new { id });
    }

    // GET: Customers/Details/5
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }

        Customer customer = db.Customers.Include(t => t.Orders).FirstOrDefault(t
=> t.Id == id);
        if (customer == null)
        {
            return HttpNotFound();
        }
        return View(customer);
    }

    // GET: Customers/Create
    public ActionResult Create()
    {
        ViewBag.Id = new SelectList(db.Users, "Id", "Email");
        return View();
    }

    // POST: Customers/Create
    // Чтобы защититься от атак чрезмерной передачи данных, включите определенные
свойства, для которых следует установить привязку. Дополнительные
// сведения см. в статье https://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include = "Id,sumVotes, totalVotes")]
Customer customer)
    {
        if (ModelState.IsValid)
        {
            db.Customers.Add(customer);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        ViewBag.Id = new SelectList(db.Users, "Id", "Email", customer.Id);
        return View(customer);
    }

    // GET: Customers/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
    }

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						85
Изм.	Лист	№ докум.	Подп.	Дата		

```

        Customer customer = db.Customers.Find(id);
        if (customer == null)
        {
            return HttpNotFound();
        }
        ViewBag.Id = new SelectList(db.Users, "Id", "Email", customer.Id);
        return View(customer);
    }

    // POST: Customers/Edit/5
    // Чтобы защититься от атак чрезмерной передачи данных, включите определенные
    // свойства, для которых следует установить привязку. Дополнительные
    // сведения см. в статье https://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include = "Id,sumVotes,totalVotes")] Customer
customer)
    {
        if (ModelState.IsValid)
        {
            db.Entry(customer).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        ViewBag.Id = new SelectList(db.Users, "Id", "Email", customer.Id);
        return View(customer);
    }

    // GET: Customers/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Customer customer = db.Customers.Find(id);
        if (customer == null)
        {
            return HttpNotFound();
        }
        return View(customer);
    }

    // POST: Customers/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Customer customer = db.Customers.Find(id);
        db.Customers.Remove(customer);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						86
Изм.	Лист	№ докум.	Подп.	Дата		

## Файл LKController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using UlskDel.Models;
using System.Data.Entity;

namespace UlskDel.Controllers
{
    public class LKController : Controller
    {
        private OrderContext db = new OrderContext();
        // GET: LK
        public ActionResult Index()
        {
            User user = db.Users.Where(x => x.Email ==
User.Identity.Name).FirstOrDefault();
            int id = user.Id;
            if (User.IsInRole("customer"))
            {
                return RedirectToAction("Details", "Customers", new { id });
            } else
            {
                return RedirectToAction("Details", "Couriers", new { id });
            }
        }
    }
}
```

## Файл OrdersController.cs

```
using iTextSharp.text;
using iTextSharp.text.pdf;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using UlskDel.Models;
using iTextSharp.text.html;
using iTextSharp.tool.xml.pipeline.html;
using iTextSharp.tool.xml.html;
using iTextSharp.tool.xml.pipeline.css;
using iTextSharp.tool.xml;
using iTextSharp.tool.xml.pipeline.end;
using System.Xml.Linq;
using iTextSharp.tool.xml.parser;
using System.Text;
using ExcelLibrary.SpreadSheet;

namespace UlskDel.Controllers
{
    public class OrdersController : Controller
    {
        private OrderContext db = new OrderContext();
```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						87
Изм.	Лист	№ докум.	Подп.	Дата		

```

[HttpGet]
[AllowAnonymous]
public JsonResult CheckAddress(string Address_Sender)
{
    var result = Address_Sender.Contains("Ульяновск");
    return Json(result, JsonRequestBehavior.AllowGet);
}

[HttpGet]
[AllowAnonymous]
public JsonResult CheckDate(DateTime Date)
{
    DateTime now = DateTime.Now;
    var result = Date.Date.CompareTo(now.Date) > 0 &&
Date.Date.CompareTo(now.AddYears(1)) < 0;
    return Json(result, JsonRequestBehavior.AllowGet);
}

// GET: Orders
public ActionResult Index(int id = 2)
{
    var elem = from s in db.Orders
               select s;

    List<SelectListItem> item = new List<SelectListItem>();
    //фильтрация по периоду
    item.Add(new SelectListItem { Text = "неделя", Value = "0"});
    item.Add(new SelectListItem { Text = "месяц", Value = "1"});
    item.Add(new SelectListItem { Text = "все", Value = "2", Selected =
true});

    DateTime week = DateTime.Now.AddDays(-7);
    DateTime month = DateTime.Now.AddMonths(-1);
    DateTime now = DateTime.Now;
    ViewBag.id = item;
    switch (id)
    {
        case 0: elem = elem.Where(s => s.Date.CompareTo(week)>0
&& s.Date.CompareTo(now)<0);break;
        case 1: elem = elem.Where(s => s.Date.CompareTo(month)>0
&& s.Date.CompareTo(now) < 0); break;
        case 2: elem = db.Orders;break;
    }

    return View(elem.ToList());
}

// GET: Orders/Details/5
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Order order = db.Orders.Find(id);
    if (order == null)
    {
        return HttpNotFound();
    }
    return View(order);
}

// GET: Orders/Create
public ActionResult Create()
{

```

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						88
Изм.	Лист	№ докум.	Подп.	Дата		

```

        return View();
    }

    // POST: Orders/Create
    // Чтобы защититься от атак чрезмерной передачи данных, включите определенные
    свойства, для которых следует установить привязку. Дополнительные
    // сведения см. в статье https://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include =
    "OrderId,Sender,Receiver,Address_Sender,Address_Receiver,Area_Sender,Area_Receiver,Pho
    ne_Sender,Phone_Receiver,Date,Time,Weight,Length,Width,Height,Price,Big,Fragile,Who_pa
    y")] Order order)
    {
        if (ModelState.IsValid)
        {
            User user = db.Users.Where(x => x.Email ==
            User.Identity.Name).FirstOrDefault();
            int id = user.Id;
            order.CustomerId = id;
            order.Status = "обрабатывается";
            order.Print = false;
            //Находим курьера для получения посылки
            var elem = from s in db.Couriers
                        select s;
            if (order.Big)
            {
                elem = elem.Where(x => x.oversize);
            } else
            {
                elem = elem.Where(x => x.Area == order.Area_Sender);
            }
            //привязываем курьера к данному заказу
            if (elem.Count() == 1)
                order.CourierId = elem.FirstOrDefault().Id;
            db.Orders.Add(order);
            db.SaveChanges();
            return RedirectToAction("Index", "LK");
        }

        return View(order);
    }

    // GET: Orders/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Order order = db.Orders.Find(id);
        if (order == null)
        {
            return HttpNotFound();
        }
        IEnumerable < SelectListItem > userTypeList = new
        SelectList(db.Couriers.ToList(), "Id", "Name");
        ViewData["userTypeList"] = userTypeList;
        return View(order);
    }

    // POST: Orders/Edit/5
    // Чтобы защититься от атак чрезмерной передачи данных, включите определенные
    свойства, для которых следует установить привязку. Дополнительные

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						89
Изм.	Лист	№ докум.	Подп.	Дата		

```

        // сведения см. в статье https://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include =
"OrderId,Sender,Receiver,Address_Sender,Address_Receiver,Area_Sender,Area_Receiver,Phone_Sender,Phone_Receiver,Date,Time,Status,Weight,Length,Width,Height,Big,Fragile,Who_pay,Price,CustomerId,CourierId")] Order order)
        {
            IEnumerable<SelectListItem> userTypeList = new
SelectList(db.Couriers.ToList(), "Id", "Name");
            if (!ModelState.IsValid)
            {
                Console.WriteLine(order.CourierId);
            }
            if (ModelState.IsValid)
            {
                db.Entry(order).State = EntityState.Modified;
                if (order.Status!="обрабатывается")
                {
                    order.Print = true;
                }
                //Нашли курьера данного заказа
                Courier courier = db.Couriers.FirstOrDefault(a => a.Id ==
order.CourierId);
                if (order.Status != "получен" && courier.Orders.Count == 1)
                {
                    //Курьер завершил все заказы и стал свободен
                    courier.time = DateTime.Now;
                }
                //заказчик
                Customer customer = db.Customers.FirstOrDefault(a => a.Id ==
order.CustomerId);
                //Стоимость в зависимости от рейтинга
                if (customer.rating > 0 && customer.rating < 2.07)
                {
                    order.Price = (int)Math.Round(order.Price * 1.2);
                }
                else if (customer.rating >= 3.1)
                {
                    order.Price = (int)Math.Round(order.Price * 0.8);
                }
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            return View(order);
        }

        // GET: Orders/Delete/5
        public ActionResult Delete(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Order order = db.Orders.Find(id);
            if (order == null)
            {
                return HttpNotFound();
            }
            return View(order);
        }

        // POST: Orders/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						90
Изм.	Лист	№ докум.	Подп.	Дата		

```

public ActionResult DeleteConfirmed(int id)
{
    Order order = db.Orders.Find(id);
    db.Orders.Remove(order);
    db.SaveChanges();
    return RedirectToAction("Index");
}

//печать в pdf
public ActionResult Print(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Order order = db.Orders.Find(id);
    if (order == null)
    {
        return HttpNotFound();
    }
    string Path = Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
+ "/MyFile2.pdf";
    Document DC = new Document();
    DC.SetPageSize(PageSize.A4.Rotate());
    FileStream FS = System.IO.File.Create(Path);
    PdfWriter writer = PdfWriter.GetInstance(DC, FS);
    DC.Open();

    BaseFont baseFont = BaseFont.CreateFont(@"C:\Windows\Fonts\arial.ttf",
BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
    iTextSharp.text.Font font = new iTextSharp.text.Font(baseFont,
iTextSharp.text.Font.DEFAULTSIZE, iTextSharp.text.Font.NORMAL);
    PdfPTable table = new PdfPTable(3); //число столбцов
    //Добавим в таблицу общий заголовок
    PdfPCell cell = new PdfPCell(new Phrase("Отправитель ", font));
    cell.Colspan = 3;
    cell.HorizontalAlignment = 1;
    //Убираем границу первой ячейки, чтобы балы как заголовок
    cell.Border = 0;
    table.AddCell(cell);
    cell = new PdfPCell(new Phrase(new Phrase("ФИО", font)));
    //Фоновый цвет (необязательно, просто сделаем по красивее)
    cell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY;
    table.AddCell(cell);
    cell = new PdfPCell(new Phrase(new Phrase("Телефон", font)));
    //Фоновый цвет (необязательно, просто сделаем по красивее)
    cell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY;
    table.AddCell(cell);
    cell = new PdfPCell(new Phrase(new Phrase("Адрес", font)));
    //Фоновый цвет (необязательно, просто сделаем по красивее)
    cell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY;
    table.AddCell(cell);
    table.AddCell(new Phrase(order.Sender, font));
    table.AddCell(new Phrase(order.Phone_Sender, font));
    table.AddCell(new Phrase(order.Address_Sender, font));
    //Добавляем таблицу в документ
    DC.Add(table);

    //receiver
    PdfPTable table1 = new PdfPTable(3); //число столбцов
    //Добавим в таблицу общий заголовок
    PdfPCell cell1 = new PdfPCell(new Phrase("Получатель ", font));
    cell1.Colspan = 3;
    cell1.HorizontalAlignment = 1;

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						91
Изм.	Лист	№ докум.	Подп.	Дата		

```

//Убираем границу первой ячейки, чтобы балы как заголовок
cell1.Border = 0;
table1.AddCell(cell1);
cell1 = new PdfPCell(new Phrase(new Phrase("ФИО", font)));
//Фоновый цвет (необязательно, просто сделаем по красивее)
cell1.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY;
table1.AddCell(cell1);
cell1 = new PdfPCell(new Phrase(new Phrase("Телефон", font)));
//Фоновый цвет (необязательно, просто сделаем по красивее)
cell1.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY;
table1.AddCell(cell1);
cell1 = new PdfPCell(new Phrase(new Phrase("Адрес", font)));
//Фоновый цвет (необязательно, просто сделаем по красивее)
cell1.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY;
table1.AddCell(cell1);
table1.AddCell(new Phrase(order.Receiver, font));
table1.AddCell(new Phrase(order.Phone_Receiver, font));
table1.AddCell(new Phrase(order.Address_Receiver, font));
//Добавляем таблицу в документ
DC.Add(table1);

//order
PdfPTable table2 = new PdfPTable(6); //число столбцов
//Добавим в таблицу общий заголовок
PdfPCell cell2 = new PdfPCell(new Phrase("Груз ", font));
cell2.Colspan = 6;
cell2.HorizontalAlignment = 1;
//Убираем границу первой ячейки, чтобы балы как заголовок
cell2.Border = 0;
table2.AddCell(cell2);
cell2 = new PdfPCell(new Phrase(new Phrase("Вес", font)));
cell2.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY;
table2.AddCell(cell2);
cell2 = new PdfPCell(new Phrase(new Phrase("Длина", font)));
cell2.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY;
table2.AddCell(cell2);
cell2 = new PdfPCell(new Phrase(new Phrase("Высота", font)));
cell2.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY;
table2.AddCell(cell2);
cell2 = new PdfPCell(new Phrase(new Phrase("Ширина", font)));
cell2.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY;
table2.AddCell(cell2);
cell2 = new PdfPCell(new Phrase(new Phrase("Оплата получателем", font)));
cell2.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY;
table2.AddCell(cell2);
cell2 = new PdfPCell(new Phrase(new Phrase("Цена", font)));
cell2.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY;
table2.AddCell(cell2);
table2.AddCell(new Phrase(order.Weight.ToString(), font));
table2.AddCell(new Phrase(order.Length.ToString(), font));
table2.AddCell(new Phrase(order.Height.ToString(), font));
table2.AddCell(new Phrase(order.Width.ToString(), font));
string str = "нет";
if (order.Who_pay)
{
    str = "да";
}
table2.AddCell(new Phrase(str, font));
table2.AddCell(new Phrase(order.Price.ToString(), font));
//Добавляем таблицу в документ
DC.Add(table2);

DC.Add(new Paragraph("Я подтверждаю, что информация в накладной является
полной и точной. С основными условиями пересылки я ознакомлен(а).", font));

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						92
Изм.	Лист	№ докум.	Подп.	Дата		



```

        DC.SetMargins(100, 10, 10, 10);
        DC.Add(new Paragraph("Подпись заказчика_____", font));
        DC.SetMargins(100, 10, 10, 10);
        DC.Add(new Paragraph("Подпись курьера_____", font));
        DC.SetMargins(100, 10, 100, 10);

        iTextSharp.text.Image pic =
iTextSharp.text.Image.GetInstance(GET("https://api.qrserver.com/v1/create-qr-
code/?size=150x150&charset-source=ISO-8859-1",
"data=http://192.168.1.8:3000/Orders/Details/" + order.OrderId));
        DC.Add(pic);
        DC.Close();

        return RedirectToAction("Index", "LK");

    } private static byte[] GET(string Url, string Data)
    {
        WebRequest req = WebRequest.Create(Url + "&" + Data);
        byte[] bytes;
        using (HttpWebResponse res = (HttpWebResponse)req.GetResponse())
        using (Stream responseStream = res.GetResponseStream())
        using (MemoryStream mstream = new MemoryStream())
        {
            responseStream.CopyTo(mstream);
            bytes = mstream.ToArray();
        }
        return bytes;
    }

    //Сохранение отчета в xls
    [HttpPost]
    public ActionResult Save(List<Order> order )
    {
        List<Order> model = (List<Order>)TempData["FullModel"];
        string file = Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
+ "/newdoc1.xls";
        Workbook workbook = new Workbook();
        Worksheet worksheet = new Worksheet("Отчет");
        worksheet.Cells[0, 0] = new Cell("Отправитель");
        worksheet.Cells[0, 1] = new Cell("Получатель");
        worksheet.Cells[0, 2] = new Cell("Адрес отправителя");
        worksheet.Cells[0, 3] = new Cell("Район отправителя");
        worksheet.Cells[0, 4] = new Cell("Адрес получателя");
        worksheet.Cells[0, 5] = new Cell("Район получателя");
        worksheet.Cells[0, 6] = new Cell("Номер отправителя");
        worksheet.Cells[0, 7] = new Cell("Номер получателя");
        worksheet.Cells[0, 8] = new Cell("Дата");
        worksheet.Cells[0, 9] = new Cell("Время");
        worksheet.Cells[0, 10] = new Cell("Статус");
        worksheet.Cells[0, 11] = new Cell("Вес");
        worksheet.Cells[0, 12] = new Cell("Длина");
        worksheet.Cells[0, 13] = new Cell("Ширина");
        worksheet.Cells[0, 14] = new Cell("Высота");
        worksheet.Cells[0, 15] = new Cell("Негабаритный");
        worksheet.Cells[0, 16] = new Cell("Хрупкий");
        worksheet.Cells[0, 17] = new Cell("Оплата отправителем");
        worksheet.Cells[0, 18] = new Cell("Цена");
        worksheet.Cells[0, 19] = new Cell("ID пользователя");

        for (int i = 0; i <= model.Count-1; i++)
        {
            worksheet.Cells[i + 1, 0] = new Cell(model[i].Sender);
            worksheet.Cells[i + 1, 1] = new Cell(model[i].Receiver);
            worksheet.Cells[i + 1, 2] = new Cell(model[i].Address_Sender);
        }
    }

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						93
Изм.	Лист	№ докум.	Подп.	Дата		

```

        worksheet.Cells[i + 1, 3] = new Cell(model[i].Area_Sender.ToString());
        worksheet.Cells[i + 1, 4] = new Cell(model[i].Address_Receiver);
        worksheet.Cells[i + 1, 5] = new
Cell(model[i].Area_Receiver.ToString());
        worksheet.Cells[i + 1, 6] = new Cell(model[i].Phone_Sender);
        worksheet.Cells[i + 1, 7] = new Cell(model[i].Phone_Receiver);
        worksheet.Cells[i + 1, 8] = new Cell(model[i].Date, @"YYYY-MM-DD");
        worksheet.Cells[i + 1, 9] = new Cell(model[i].Time, @"hh:mm");
        worksheet.Cells[i + 1, 10] = new Cell(model[i].Status);
        worksheet.Cells[i + 1, 11] = new Cell((decimal)model[i].Weight);
        worksheet.Cells[i + 1, 12] = new Cell((decimal)model[i].Length);
        worksheet.Cells[i + 1, 13] = new Cell((decimal)model[i].Width);
        worksheet.Cells[i + 1, 14] = new Cell((decimal)model[i].Height);
        worksheet.Cells[i + 1, 15] = new Cell((bool)model[i].Big);
        worksheet.Cells[i + 1, 16] = new Cell((bool)model[i].Fragile);
        worksheet.Cells[i + 1, 17] = new Cell((bool)model[i].Who_pay);
        worksheet.Cells[i + 1, 18] = new Cell((int)model[i].Price);
        worksheet.Cells[i + 1, 19] = new Cell((int)model[i].CustomerId);
    }
    workbook.Worksheets.Add(worksheet);
    workbook.Save(file);
    return RedirectToAction("Index");
}

public ActionResult Distribute(int id = 4)
{
    var elem = from s in db.Orders
               select s;

    var cour = from s in db.Couriers
               select s;

    List<SelectListItem> item = new List<SelectListItem>();
    //фильтрация по районам
    item.Add(new SelectListItem { Text = "Ленинский", Value = "0" });
    item.Add(new SelectListItem { Text = "Засвияжский", Value = "1" });
    item.Add(new SelectListItem { Text = "Заволжский", Value = "2" });
    item.Add(new SelectListItem { Text = "Железнодорожный", Value = "3" });
    item.Add(new SelectListItem { Text = "Все", Value = "4", Selected = true
});

    DateTime week = DateTime.Now.AddDays(-7);
    DateTime month = DateTime.Now.AddMonths(-1);
    DateTime now = DateTime.Now;
    ViewBag.id = item;
    switch (id)
    {
        case 0:
            elem = elem.Where(s => s.Area_Sender == Areas.Ленинский);
            cour = cour.Where(x => x.Area == Areas.Ленинский);
            break;
        case 1:
            elem = elem.Where(s => s.Area_Sender == Areas.Засвияжский);
            cour = cour.Where(x => x.Area == Areas.Засвияжский);
            break;
        case 2:
            elem = elem.Where(s => s.Area_Sender == Areas.Заволжский);
            cour = cour.Where(x => x.Area == Areas.Заволжский);
            break;
        case 3:
            elem = elem.Where(s => s.Area_Sender == Areas.Железнодорожный);
            cour = cour.Where(x => x.Area == Areas.Железнодорожный);
            break;
        case 4: elem = db.Orders; break;
    }
}

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						94
Изм.	Лист	№ докум.	Подп.	Дата		

```

        ViewData["MyList"] = cour.ToList();
        elem = elem.Where(x => x.CourierId == null);
        return View(elem.ToList());
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

## Файл UsersController.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Security.Cryptography;
using System.Text;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;
using UlskDel.Models;

namespace UlskDel.Controllers
{
    public class UsersController : Controller
    {
        private OrderContext db = new OrderContext();

        // GET: Users
        public ActionResult Index()
        {
            var users = db.Users.Include(u => u.Role);
            return View(users.ToList());
        }

        // GET: Users/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            User user = db.Users.Include(u => u.Role).FirstOrDefault(u => u.Id == id);
            if (user == null)
            {
                return HttpNotFound();
            }
            return View(user);
        }

        // GET: Users/Create
        public ActionResult Create()
        {
            ViewBag.RoleId = new SelectList(db.Roles, "Id", "Name");
            return View();
        }
    }
}

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						95
Изм.	Лист	№ докум.	Подп.	Дата		

```

// POST: Users/Create
// Чтобы защититься от атак чрезмерной передачи данных, включите определенные
свойства, для которых следует установить привязку. Дополнительные
// сведения см. в статье https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Id,Email,Password,RoleId")] User
user)
{
    if (ModelState.IsValid)
    {
        db.Users.Add(user);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    ViewBag.RoleId = new SelectList(db.Roles, "Id", "Name", user.RoleId);
    return View(user);
}

// GET: Users/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    User user = db.Users.Find(id);
    if (user == null)
    {
        return HttpNotFound();
    }
    ViewBag.RoleId = new SelectList(db.Roles, "Id", "Name", user.RoleId);
    return View(user);
}

// POST: Users/Edit/5
// Чтобы защититься от атак чрезмерной передачи данных, включите определенные
свойства, для которых следует установить привязку. Дополнительные
// сведения см. в статье https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "Id,Email,Password,RoleId")] User
user)
{
    if (ModelState.IsValid)
    {
        string pwd = GetHash(user.Password);
        db.Entry(user).State = EntityState.Modified;
        user.Password = pwd;
        db.SaveChanges();
        // Обновление Логина в куках (by User.Identity.Name)
        //при изменении логина
        FormsAuthenticationTicket ticket = new FormsAuthenticationTicket(1,
            user.Email,
            DateTime.Now,
            DateTime.Now.AddMinutes(30),
            false,
            "someData",
            FormsAuthentication.FormsCookiePath);
        // encrypt the ticket
        string encTicket = FormsAuthentication.Encrypt(ticket);
        // create the cookie

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						96
Изм.	Лист	№ докум.	Подп.	Дата		

```

        Response.Cookies.Add(new
        HttpCookie(FormsAuthentication.FormsCookieName, encTicket));

        return View(user);
    }
    ViewBag.RoleId = new SelectList(db.Roles, "Id", "Name", user.RoleId);
    return View(user);
}

// GET: Users/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    User user = db.Users.Include(u => u.Role).FirstOrDefault(u => u.Id == id);
    if (user == null)
    {
        return HttpNotFound();
    }
    return View(user);
}

// POST: Users/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    User user = db.Users.Find(id);
    DeleteRole(user);
    db.Users.Remove(user);
    db.SaveChanges();
    return RedirectToAction("Index");
}

private void DeleteRole(User user)
{
    switch (user.RoleId)
    {
        case 1:
            Customer customer = db.Customers.Find(user.Id);
            db.Customers.Remove(customer);
            break;
        case 3:
            Courier courier = db.Couriers.Find(user.Id);
            db.Couriers.Remove(courier);
            break;
    }
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}

private string GetHash(string input)
{
    var md5 = MD5.Create();
    var hash = md5.ComputeHash(Encoding.UTF8.GetBytes(input));
}

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						97
Изм.	Лист	№ докум.	Подп.	Дата		

```

        return Convert.ToBase64String(hash);
    }
}

```

## Файл CustomRoleProvider.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;
using UlskDel.Models;
using System.Data.Entity;

namespace UlskDel.Providers
{
    public class CustomRoleProvider : RoleProvider
    {
        public override string[] GetRolesForUser(string username)
        {
            string[] roles = new string[] { };
            using (OrderContext db = new OrderContext())
            {
                // Получаем пользователя
                User user = db.Users.Include(u => u.Role).FirstOrDefault(u => u.Email
== username);
                if (user != null && user.Role != null)
                {
                    // получаем роль
                    roles = new string[] { user.Role.Name };
                }
                return roles;
            }
        }
        public override void CreateRole(string roleName)
        {
            throw new NotImplementedException();
        }
        public override bool IsUserInRole(string username, string roleName)
        {
            using (OrderContext db = new OrderContext())
            {
                // Получаем пользователя
                User user = db.Users.Include(u => u.Role).FirstOrDefault(u => u.Email
== username);

                if (user != null && user.Role != null && user.Role.Name == roleName)
                    return true;
                else
                    return false;
            }
        }
        public override void AddUsersToRoles(string[] usernames, string[] roleNames)
        {
            throw new NotImplementedException();
        }

        public override string ApplicationName
        {
            get { throw new NotImplementedException(); }
            set { throw new NotImplementedException(); }
        }

        public override bool DeleteRole(string roleName, bool throwOnPopulatedRole)

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						98
Изм.	Лист	№ докум.	Подп.	Дата		

```

        {
            throw new NotImplementedException();
        }

        public override string[] FindUsersInRole(string roleName, string
usernameToMatch)
        {
            throw new NotImplementedException();
        }

        public override string[] GetAllRoles()
        {
            throw new NotImplementedException();
        }

        public override string[] GetUsersInRole(string roleName)
        {
            throw new NotImplementedException();
        }

        public override void RemoveUsersFromRoles(string[] usernames, string[]
roleNames)
        {
            throw new NotImplementedException();
        }

        public override bool RoleExists(string roleName)
        {
            throw new NotImplementedException();
        }
    }
}

```

## Файл Models.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;

namespace UlskDel.Models
{
    public class LoginModel
    {
        [Required]
        [RegularExpression(@"[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}",
ErrorMessage = "Некорректный адрес")]
        [Display(Name = "Почта")]
        [MaxLength(50)]
        public string Name { get; set; }

        [Required]
        [DataType(DataType.Password)]
        [MinLength(6)]
        [MaxLength(50)]
        [Display(Name = "Пароль")]
        public string Password { get; set; }
    }

    public class RegisterModel
    {
        [Required]
        [RegularExpression(@"[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}",
ErrorMessage = "Некорректный адрес")]

```

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						99
Изм.	Лист	№ докум.	Подп.	Дата		

```

        [Display(Name = "Почта")]
        [MaxLength(50)]
        public string Name { get; set; }

        [Required]
        [DataType(DataType.Password)]
        [MinLength(3)]
        [MaxLength(50)]
        [Display(Name = "Пароль")]
        public string Password { get; set; }

        [Required]
        [DataType(DataType.Password)]
        [MinLength(3)]
        [Compare("Password", ErrorMessage = "Пароли не совпадают")]
        [Display(Name = "Подтверждение пароля")]
        public string ConfirmPassword { get; set; }
    }
}

```

## Файл Order.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace UlskDel.Models
{
    public class Order
    {
        public int OrderId { get; set; }
        [Display(Name = "Отправитель")]
        [MinLength(2)]
        [MaxLength(50)]
        [Required]
        public string Sender { get; set; }
        [Display(Name = "Получатель")]
        [MinLength(2)]
        [MaxLength(50)]
        [Required]
        public string Receiver { get; set; }
        [Display(Name = "Адрес отправителя")]
        [Required]
        [MaxLength(200)]
        [Remote("CheckAddress", "Orders", ErrorMessage = "Неверный адрес")]
        public string Address_Sender { get; set; }
        [Display(Name = "Адрес получателя")]
        [Required]
        [MaxLength(200)]
        public string Address_Receiver { get; set; }
        [Required]
        [Display(Name = "Район отправки")]
        public Areas Area_Sender { get; set; }
        [Required]
        [Display(Name = "Район получения")]
        public Areas Area_Receiver { get; set; }
        [Required]
        [RegularExpression(@"^((8|\+7)[\-\ ]?)?(\(?[0-9]{3}\)?[\-\ ]?)?[\d\-\ ]{7,10}$",
        ErrorMessage = "Некорректный номер телефона.")]
        [Display(Name = "Номер отправителя")]
        public string Phone_Sender { get; set; }
    }
}

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						100
Изм.	Лист	№ докум.	Подп.	Дата		



```

        [Required]
        [RegularExpression(@"^((8|\+7)[\ - ]?)?(\\(?\d{3}\\)?)?[\ - ]?)?[\d\ - ]{7,10}$",
ErrorMessage = "Некорректный номер телефона.")]
        [Display(Name = "Номер получателя")]
        public string Phone_Receiver { get; set; }
        [Required]
        [DataType(DataType.Date)]
        [Remote("CheckDate", "Orders", ErrorMessage = "Неверная дата")]
        [Display(Name = "Дата")]
        public System.DateTime Date { get; set; }
        [Required]
        [DataType(DataType.Time)]
        [Display(Name = "Время")]
        public System.DateTime Time { get; set; }
        [Display(Name = "Статус")]
        public string Status { get; set; }
        [Display(Name = "Вес, кг")]
        [Range(0,10)]
        [Required]
        public float Weight { get; set; }
        [Display(Name = "Длина, см")]
        [Range(0,200)]
        [Required]
        public float Length { get; set; }
        [Display(Name = "Ширина, см")]
        [Range(0, 200)]
        [Required]
        public float Width { get; set; }
        [Display(Name = "Высота, см")]
        [Range(0, 200)]
        [Required]
        public float Height { get; set; }
        [Display(Name = "Негабаритный")]
        public bool Big { get; set; }
        [Display(Name = "Хрупкое")]
        public bool Fragile { get; set; }
        [Display(Name = "Оплата отправителем")]
        public bool Who_pay { get; set; }
        [Display(Name = "Цена")]
        [Required(ErrorMessage = "Нажмите кнопку Рассчитать")]
        public int Price { get; set; }
        public bool Print { get; set; }
        public int CustomerId { get; set; }
        public Customer Customer { get; set; }
        public int? CourierId { get; set; }
        [ForeignKey("CourierId")]
        public Courier Courier { get; set; }
    }

    public enum Areas : byte
    {
        Железнодорожный = 1,
        Заволжский = 2,
        Засвияжский = 3,
        Ленинский = 4
    }
}

```

## Файл OrderContext.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						101
Изм.	Лист	№ докум.	Подп.	Дата		

```

using System.Data.Entity;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace UlskDel.Models
{
    public class OrderContext : DbContext
    {
        public OrderContext() : base("DefaultConnection")
        {
        }
        public DbSet<Order> Orders { get; set; }
        public DbSet<User> Users { get; set; }
        public DbSet<Role> Roles { get; set; }
        public DbSet<Customer> Customers { get; set; }
        public DbSet<Admin> Admins { get; set; }
        public DbSet<Courier> Couriers { get; set; }
    }
    public class User
    {
        public int Id { get; set; }
        [Display(Name = "Почта")]
        [MinLength(4)]
        public string Email { get; set; }
        [Display(Name = "Пароль")]
        [DataType(DataType.Password)]
        [MinLength(3)]
        public string Password { get; set; }
        public int RoleId { get; set; }
        public Role Role { get; set; }
    }

    public class Role
    {
        public int Id { get; set; }
        [Display(Name = "Роль")]
        public string Name { get; set; }
    }

    public class Customer
    {
        [Key]
        [ForeignKey("User")]
        public int Id { get; set; }
        public User User { get; set; }
        //сумма всех голосов
        public int sumVotes { get; set; }
        //кол-во голосов
        public int totalVotes { get; set; }
        [Display(Name = "Рейтинг")]
        public double rating
        {
            get
            {
                //диапазон возможных голосов
                int[] votesRange = { 1, 2, 3, 4, 5 };
                if (sumVotes > 0 && totalVotes > 0) {
                    float z = 1.64485f;
                    int vMin = votesRange.Min();
                    float vWidth = votesRange.Max() - vMin;
                    float phat = (sumVotes - totalVotes * vMin) / vWidth /
totalVotes;
                }
            }
        }
    }
}

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						102
Изм.	Лист	№ докум.	Подп.	Дата		

```

        double rating = (phat + z * z / (2 * totalVotes) - z *
Math.Sqrt((phat* (1 - phat) + z * z / (4 * totalVotes)) / totalVotes)) / (1 + z * z /
totalVotes);

        return Math.Round(rating * vWidth + vMin, 6);
    }
    return 0;
}
}
public ICollection<Order> Orders { get; set; }
public Customer()
{
    Orders = new List<Order>();
}
}

public class Admin
{
    [Key]
    [ForeignKey("User")]
    public int Id { get; set; }
    public User User { get; set; }
}

public class Courier
{
    [Key]
    [ForeignKey("User")]
    public int Id { get; set; }
    public User User { get; set; }
    [MaxLength(50)]
    [Display(Name = "Имя")]
    public string Name { get; set; }
    public int sumVotes { get; set; }
    public int totalVotes { get; set; }
    [Display(Name = "Рейтинг")]
    public double rating
    {
        get
        {
            int[] votesRange = { 1, 2, 3, 4, 5 };
            if (sumVotes > 0 && totalVotes > 0)
            {
                float z = 1.64485f;
                int vMin = votesRange.Min();
                float vWidth = votesRange.Max() - vMin;
                float phat = (sumVotes - totalVotes * vMin) / vWidth / totalVotes;
                double rating = (phat + z * z / (2 * totalVotes) - z *
Math.Sqrt((phat * (1 - phat) + z * z / (4 * totalVotes)) / totalVotes)) / (1 + z * z /
totalVotes);

                return Math.Round(rating * vWidth + vMin, 6);
            }
            return 0;
        }
    }
    [Display(Name = "Время")]
    public DateTime time { get; set; }
    [Display(Name = "Газель")]
    public bool oversize { get; set; }
    [Display(Name = "Район")]
    public Areas Area { get; set; }
    [Required]
    [Remote("IsExist", "Courier", ErrorMessage = "URL exist!")]
    public ICollection<Order> Orders { get; set; }
    public Courier()

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						103
Изм.	Лист	№ докум.	Подп.	Дата		

```

        {
            Orders = new List<Order>();
        }
    }
}

```

## Файл OrderDbInitializer.cs

```

using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Web;

namespace UlskDel.Models
{
    public class OrderDbInitializer : DropCreateDatabaseIfModelChanges<OrderContext>
    {
        protected override void Seed(OrderContext db)
        {
            Role admin = new Role { Name = "admin" };
            Role customer = new Role { Name = "customer" };
            Role courier = new Role { Name = "courier" };
            db.Roles.Add(admin);
            db.Roles.Add(customer);
            db.Roles.Add(courier);

            string pwdUser = GetHash("123");
            User first = db.Users.Add(new User
            {
                Email = "user@mail.com",
                Password = pwdUser,
                Role = customer
            });
            base.Seed(db);
            db.SaveChanges();

            Customer cust = new Customer { Id = first.Id, sumVotes = 0, totalVotes = 0 };
            db.Customers.Add(cust);

            string pwd = GetHash("123456");
            User second = db.Users.Add(new User
            {
                Email = "admin@mail.com",
                Password = pwd,
                Role = admin
            });
            base.Seed(db);

            string pwd2 = GetHash("123456");
            User cour1 = db.Users.Add(new User
            {
                Email = "courier@mail.com",
                Password = pwd2,
                Role = courier
            });
            base.Seed(db);

            User cour2 = db.Users.Add(new User
            {

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						104
Изм.	Лист	№ докум.	Подп.	Дата		

```

        Email = "courier2@mail.com",
        Password = pwd2,
        Role = courier
    });
    base.Seed(db);
    User cour3 = db.Users.Add(new User
    {
        Email = "courier3@mail.com",
        Password = pwd2,
        Role = courier
    });
    base.Seed(db);
    User cour4 = db.Users.Add(new User
    {
        Email = "courier4@mail.com",
        Password = pwd2,
        Role = courier
    });
    base.Seed(db);
    User cour5 = db.Users.Add(new User
    {
        Email = "courier5@mail.com",
        Password = pwd2,
        Role = courier
    });
    base.Seed(db);
    db.SaveChanges();

    var cour = new List<Courier>
    {
        new Courier {Id = cour1.Id, sumVotes = 0, totalVotes = 0, time =
DateTime.Now, Area = Areas.Ленинский, Name = "Первый" },
        new Courier {Id = cour2.Id, sumVotes = 0, totalVotes = 0, time =
DateTime.Now, Area = Areas.Железнодорожный, Name = "Второй" },
        new Courier {Id = cour3.Id, sumVotes = 0, totalVotes = 0, time =
DateTime.Now, Area = Areas.Заволжский, Name = "Третий" },
        new Courier {Id = cour4.Id, sumVotes = 0, totalVotes = 0, time =
DateTime.Now, Area = Areas.Засвияжский, Name = "Четвертый" },
        new Courier {Id = cour5.Id, sumVotes = 0, totalVotes = 0, time =
DateTime.Now, Area = Areas.Засвияжский, oversize = true, Name = "Пятый" }
    };
    cour.ForEach(s => db.Couriers.Add(s));
    db.SaveChanges();
}

private string GetHash(string input)
{
    var md5 = MD5.Create();
    var hash = md5.ComputeHash(Encoding.UTF8.GetBytes(input));
    return Convert.ToBase64String(hash);
}
}
}

```

### Файл myscript.js

```

ymaps.ready(init);
var suggestView1;
var suggestView2;

function init() {
    suggestView1 = new ymaps.SuggestView('suggest', {boundedBy: [[54.35, 48.24],[54.30,
48.73]], results:3});
    suggestView2 = new ymaps.SuggestView('suggest2', {boundedBy: [[54.35,
48.24],[54.30, 48.73]], results:3});
}

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		105

```

var mapDivId = 'map'; //Id контейнера для карты
var mapCenter = [55.76, 37.64]; //Координата центра карты по умолчанию
map = new ymaps.Map(mapDivId, { center: mapCenter, zoom: 10 });
geocode('#suggest', '#notice');
geocode('#suggest2', '#notice2');

// При клике по кнопке запускаем верификацию введённых данных.
$('#suggest').on('change', function (e) {
    geocode('#suggest', '#notice');
});
$('#suggest2').on('change', function (e) {
    geocode('#suggest2', '#notice2');
});

function geocode(item, notice) {
    // Забираем запрос из поля ввода.
    var request = $(item).val();
    // Геокодируем введённые данные.
    ymaps.geocode(request).then(function (res) {
        var obj = res.geoObjects.get(0),
            error, hint;
        if (obj) {
            // Об оценке точности ответа геокодера можно прочитать тут:
            https://tech.yandex.ru/maps/doc/geocoder/desc/reference/precision-docpage/
            switch
            (obj.properties.get('metaDataProperty.GeocoderMetaData.precision')) {
                case 'exact':
                    break;
                case 'number':
                case 'near':
                case 'range':
                    error = 'Неточный адрес, требуется уточнение';
                    hint = 'Уточните номер дома';
                    break;
                case 'street':
                    error = 'Неполный адрес, требуется уточнение';
                    hint = 'Уточните номер дома';
                    break;
                case 'other':
                default:
                    error = 'Неточный адрес, требуется уточнение';
                    hint = 'Уточните адрес';
            }
        } else {
            error = 'Адрес не найден';
            hint = 'Уточните адрес';
        }

        // Если геокодер возвращает пустой массив или неточный результат, то
        показываем ошибку.
        if (error) {
            showError(error, item, notice);
            //showMessage(hint);
        } else {
            showResult(obj, item, notice);
        }
    }, function (e) {
        console.log(e)
    })
}

function showResult(obj, item, notice) {

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						106
Изм.	Лист	№ докум.	Подп.	Дата		

```

        // Удаляем сообщение об ошибке, если найденный адрес совпадает с поисковым
        запросом.
        $(item).removeClass('input_error');
        $(notice).css('display', 'none');

        var mapContainer = $('#map'),
            bounds = obj.properties.get('boundedBy'),
            // Рассчитываем видимую область для текущего положения пользователя.
            mapState = ymaps.util.bounds.getCenterAndZoom(
                bounds,
                [mapContainer.width(), mapContainer.height()]
            ),
            // Сохраняем полный адрес для сообщения под картой.
            address = [obj.getCountry(), obj.getAddressLine()].join(', '),
            // Сохраняем укороченный адрес для подписи метки.
            shortAddress = [obj.getThoroughfare(), obj.getPremiseNumber(),
            obj.getPremise()].join(' ');
        // Убираем контролы с карты.
        mapState.controls = [];
        // Создаём карту.
        createMap(mapState, shortAddress);
        // Выводим сообщение под картой.
        //showMessage(address);
    }

    function showError(message, item, notice) {
        $(notice).text(message);
        $(item).addClass('input_error');
        $(notice).css('display', 'block');
        // Удаляем карту.
        if (map) {
            map.destroy();
            map = null;
        }
    }

    function createMap(state, caption) {
        // Если карта еще не была создана, то создадим ее и добавим метку с адресом.
        if (!map) {
            map = new ymaps.Map('map', state);
            placemark = new ymaps.Placemark(
                map.getCenter(), {
                    iconCaption: caption,
                    balloonContent: caption
                }, {
                    preset: 'islands#redDotIconWithCaption'
                });
            map.geoObjects.add(placemark);
            // Если карта есть, то выставляем новый центр карты и меняем данные и
            позицию метки в соответствии с найденным адресом.
        } else {
            map.setCenter(state.center, state.zoom);
            placemark.geometry.setCoordinates(state.center);
            placemark.properties.set({ iconCaption: caption, balloonContent: caption
        });
        }
    }

    function showMessage(message) {
        $('#messageHeader').text('Данные:');
        $('#message').text(message);
    }

    suggestView1.events.add('select', function (event) {

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						107
Изм.	Лист	№ докум.	Подп.	Дата		

```

var selected = event.get('item').value;
ymaps.geocode(selected, {
    results: 1
}).then(function (res) {
    return ymaps.geocode(res.geoObjects.get(0).geometry.getCoordinates(), {
        kind: 'district',
        results: 10
    }).then(function (res) {
        var founded = res['metaData']['geocoder']['found'];
        $('label.suggest.description').html("");
        for (i = 0; i <= founded - 1; i++) {
            var info = res.geoObjects.get(i).properties.getAll();
            var name = info['name'];
            if (name.search('район') != -1) {
                name = name.replace(' район', '');
                $('#area-sender').val(name);
                console.log(name);
            }
        }
    });
});

suggestView2.events.add('select', function (event) {
    var selected = event.get('item').value;
    ymaps.geocode(selected, {
        results: 1
    }).then(function (res) {
        return ymaps.geocode(res.geoObjects.get(0).geometry.getCoordinates(), {
            kind: 'district',
            results: 10
        }).then(function (res) {
            var founded = res['metaData']['geocoder']['found'];
            $('label.suggest.description').html("");
            for (i = 0; i <= founded - 1; i++) {
                var info = res.geoObjects.get(i).properties.getAll();
                var name = info['name'];
                if (name.search('район') != -1) {
                    name = name.replace(' район', '');
                    $('#area-receiver').val(name);
                    console.log(name);
                }
            }
        });
    });
});

function onClick() {
    var A = document.getElementById("suggest").value;
    var B = document.getElementById("suggest2").value;
    var weight = document.getElementById("Weight").value;
    var length = document.getElementById("Length").value;
    var width = document.getElementById("Width").value;
    var height = document.getElementById("Height").value;
    var fragile = document.getElementById("fragile");

    if (!A.trim() || !B.trim()) {
        alert("Введите адреса");
    } else {
        ymaps.route([A, B]).then(
            function (route) {
                var distance = route.getHumanLength(); //Получаем расстояние
                var s = distance.replace(' ', '');
            }
        );
    }
}

```

					ВКР — УЛГТУ — 09.03.02 — 16/114 — 2020 ПЗ	Лист
						108
Изм.	Лист	№ докум.	Подп.	Дата		



```

var x = s.toString();
var k = x.indexOf("&");
x = x.substr(0, k); //расстояние
//Получаем объемный вес
var v = (length * width * height) / 5000;
var price = weight;
if (v > parseFloat(weight)) {
    price = v;
}
price = Math.round(price * 6 + 4 * x); //цена
if (fragile.checked) {
    price = price + 50;
}
$('#price').val(price);
price = price + "р. Это примерная стоимость доставки.Окончательная
установится после взвешивания груза";
$('#message').text(price);
console.log(price);
map.geoObjects.add(route); //Рисуем маршрут на карте
},
function (error) {
    alert('Ошибка: ' + error.message);
}
);
}
}

```

					ВКР – УЛГТУ – 09.03.02 – 16/114 – 2020 ПЗ	Лист
						109
Изм.	Лист	№ докум.	Подп.	Дата		