

Федеральное государственное образовательное бюджетное
учреждение высшего образования
**«Финансовый университет при Правительстве
Российской Федерации»**

Департамент анализа данных, принятия решений и финансовых технологий

Курсовая работа по дисциплине
«Современные технологии программирования»
на тему:
«Информационно-справочная система издательства»

Выполнил:
студент ПИ19-4
Кузнецов М. А.



Научный руководитель:
Д.э.н., профессор
Демин И. С.

Москва
2021

Содержание

Введение.....	3
1. Постановка задачи.....	3
2. Описание предметной области	4
2.1. Глоссарий предметной области.....	5
3. Актуальность автоматизации.....	5
3.1. Требования к информационной системе	6
4. Описание программы.....	8
4.1. Алгоритмические решения	8
4.2. Описание интерфейса программы.....	9
4.3. Состав приложения.....	26
5. Назначение и состав классов программы	29
Заключение	36
Список литературы	37

Введение

В наше время рост популярности информационных технологий затрагивает многие сферы деятельности, появляется большое количество информационных ресурсов, на которых обработана и отражена работа с большим потоком информации. Компании нуждаются в таких ресурсах, которые смогут контролировать, управлять и извлекать большие данные. Автоматизация сайтов предоставляет собой некие решения, с помощью которых владелец и сотрудники смогут выполнить самостоятельное управление ресурсом, то есть у них появится возможность автоматизировать достаточно большое количество задач. Приведу пример: у нас есть некая компания, выпускающие продукты (игры, программное обеспечение и т.п.), ей необходимо работать и обновлять большой поток информации с определенной периодичностью.

Если заниматься этим вручную, то данная компания будет нуждаться в расширении своего штата сотрудников. Для того, чтобы не допустить этого, такую задачу можно решить слиянием внутренних программных ресурсов данной компании.

1. Постановка задачи

Необходимо разработать кроссплатформенное приложение (Клиент) и приложение, которое занимается хранением и обработкой информации, (Сервер), выполняющее учет и отображение объектов предметной области, хранящихся в базе данных (БД), и предоставляющее справочную и статистическую информацию. Информационная модель предметной области должна быть не проще двухуровневой структуры данных. Допускается реализация на языке программирования Java. Задача сервера состоит в том, чтобы он мог обрабатывать данные, которые получает от Клиента, сохраняя их в Базу Данных, уметь делать запросы к Базе Данных. Необходимо реализовать информационную модель, представленную в виде

пользовательских классов и таблиц Базы Данных. Взаимодействовать с Базой Данных необходимо при помощи ORM.

2. Описание предметной области

В данной работе предметной областью является издательство. Глобально, издательство представляет собою организацию, которая связывает людей, желающих заказать печатную продукцию с авторами и типографиями.

В рамках курсовой работы было необходимо разработать информационно-справочную систему издательства. Клиентами данного издательства являются юридические лица. Для них должна быть возможность заказа интересующей их продукции к определенной дате в требуемом количестве. В виду того, что у клиентов-компаний может измениться расчетный счет, телефон, электронный адрес и т.д., необходимо наличие Личного кабинета клиента. Клиентская база фирмы постоянно растет, появляются новые клиенты.

Со стороны издательства работа выглядит следующим образом. Заказ отображается в системе, сотрудник издательства его обрабатывает. Под обработкой заказа будем понимать процесс его согласования с автором, клиентом и типографией. Заказ считается обработанным тогда, когда достигнуты все договоренности. Кроме того, нужно своевременно оповещать клиента о том, в каком состоянии готовности находится его заказ.

Издательство может изменять внутреннюю структуру, а именно, сотрудничество с авторами или типографиями может быть закончено. Также, могут быть заключены договоры с новыми типографиями, авторами и сотрудниками.

Документооборот издательства может быть весьма большим, поэтому сотрудники имеют различные должности. Например, существует менеджер по работе с клиентами, менеджер продукции, менеджер по работе с типографиями и т. д.

2.1. Глоссарий предметной области

Таблица 1. Основные понятия

Название термина	Описание термина
Продукция	Материальная единица, подлежащая продаже.
Договор	Составленный клиентом заказ у издательства.
Автор	Физическое лицо, трудом которого создана продукция.
Типография	Предприятие, которое занимается печатью и выпуском продукции.
Клиент	Юридическое лицо, заказывающее у издательства продукцию.
Сотрудник	Физическое лицо, работающее в издательстве.

3. Актуальность автоматизации

Актуальность автоматизации описанной предметной области обусловлена сложностью учета большого количества заказов разных типов продукции, большого количества клиентов. Проанализировав предметную область, было установлено, что сотрудники вынуждены тратить большое количество времени на рутинные задачи. Например, они должны вручную обрабатывать платежные реквизиты сторон, тратить много времени на формальную коммуникацию и так далее.

Кроме того, издательство развивается и клиентов становится все больше. Также, растет и ассортимент предлагаемой продукции, внешние связи издательства также разрастаются. Ввиду этого возникает большая проблема учета, контроля и аналитики этих процессов.

Также, работа с издательством должна быть комфортна клиенту. Например, не нужно его заставлять вбивать вручную реквизиты и информацию о своей компании, это должно происходить автоматически.

Таким образом, разрабатываемая программа должна помочь решить все вышеперечисленные проблемы, а именно: облегчить работу сотрудникам издательства, увеличить удобство использования услуг этой области, предоставить возможности для аналитики внутренних процессов издательства.

3.1. Требования к информационной системе

Таким образом, проанализировав предметную область, были сформулированы следующие требования к разрабатываемому программному продукту.

Прежде всего, Клиент должен иметь возможность заказать необходимое количество требуемой продукции к определенной дате. Издательство должно найти исполнителя данного товара. Кроме того, издательство берет на себе подбор типографии для выполнения заказа.

Необходимо предусмотреть личный кабинет клиента, с регистрацией и авторизацией. Клиент должен иметь возможность отслеживать, создавать и редактировать свои заказы. Также, у него имеются параметры: имя, фамилия, возраст, телефонный номер, адрес, реквизиты, электронная почта и название организации. Помимо этого, необходимо хранить данные для авторизации каждого пользователя, а именно логин и пароль. Отметим, что логин должен быть уникальным.

У типографий есть параметры: название, реквизиты, телефонный номер, адрес. Как было сказано выше, нужно предоставить возможность для удаления

и обновления информации о типографии, а также возможность добавления новой типографии.

У авторов есть: имя, фамилия, возраст, телефонный номер, адрес, реквизиты, и почта. Также, как и для типографий, для авторов нужно предусмотреть возможность удаления, добавления и обновления.

Продукция имеет название, описание и типографии, в которых она производится. Для нее также должны быть возможности удаления, добавления и обновления.

Главной сущностью разрабатываемой системы являются договоры. По сути, договор является связующим звеном между клиентом, автором, типографией и издательством. Главными параметрами договоров является название товара, название типографии, имя автора и платежные реквизиты всех заинтересованных сторон. Важно разработать систему статусов, необходимую для оповещения клиентов об актуальных изменениях в заказах.

Со стороны сотрудников информационная система должна иметь следующий функционал. Во-первых, необходимо предоставить сотрудникам возможность редактирования заказов. Во-вторых, должна быть функция удаления, добавления и обновления информации о таких сущностях, как авторы, типографии и продукции. В-третьих, нужно разграничить зоны ответственности между сотрудниками, то есть, например, менеджер по работе с клиентами не должен иметь возможность взаимодействовать с типографиями. Должность каждого сотрудника устанавливает генеральный директор.

Также, разрабатываемая система должна иметь возможность добавления и удаления сотрудников. Данная возможность должна быть только у генерального директора.

Для разделения обязанностей необходимо предусмотреть следующие роли:

- 1) USER – сотрудник, который работает с клиентами.

- 2) ADMIN – генеральный директор, в обязанность которого входит учет, анализ, контроль всех сфер, наблюдение за безопасностью системы, выдача сотрудникам доступ к их личным кабинетам.
- 3) TYPOGRAPHY_MANAGER – менеджер по работе с типографиями, производит подбор типографий, печатающих указанный клиентом товар.
- 4) ACCOUNTANT – бухгалтер, в обязанности которого входит учет и анализ договоров, по которым договору проставляется статус и цена. Так же бухгалтер производит анализ цен, в результате чего находит среднюю цену договоров, максимальную и минимальную сумму заказа, учет выручки организации.
- 5) CLIENT_MANAGER – главный сотрудник по работе с клиентами.
- 6) AUTHOR_MANAGER – менеджер по работе с авторами, производит подбор определенного автора по поставленной клиентом задаче.
- 7) PRODUCTION_MANAGER – менеджер по продукции, главной задачей которого является контроль каждой продукции.
- 8) Так же есть роль CLIENT, доступная только клиентам издательства.

4. Описание программы

4.1. Алгоритмические решения

Данная задача реализуется на языке программирования Java. За клиентскую сторону пользовательского интерфейса отвечает язык разметки гипертекста HTML и шаблонизатор FreeMarker. Также, используется технология Bootstrap - фреймворк для создания дизайна. За программно-аппаратную часть сервиса отвечает язык программирования Java, а именно библиотека Spring – фреймворк для разработки веб-приложений. В качестве базы данных была выбрана база MySQL.

В качестве хранилища данных выступает База Данных, к которой мы можем обращаться с помощью JpaRepository (это интерфейс Spring для работы с БД).

FreeMaker помогает нам составлять условия для экранирования тех или иных данных, вывод данных через список, так же он умеет импортировать написанные нами функции, служащие для уменьшения избыточности и дублирования кода.

Сервер, написанный на Spring, помогает нам безопасно работать с БД. Дело в том, что прямая работа клиента с базой данных не считается безопасной, так как есть большая вероятность ошибок, вследствие которых в базе данных могут возникнуть непредвиденные ошибки. Для того, чтобы избежать таких ситуаций нам необходим сервер, который будет принимать от клиента запрос, проверять данные этого запроса, если проверка не нашла ошибок, обращаться к БД, получать ответ от базы данных и отправлять нужную клиенту информацию.

4.2. Описание интерфейса программы

При запуске приложения, на главной странице <http://localhost:8080/> появится разворачиваемая панель для помощи посетителям сайта. Ниже расположен список доступной для заказа клиентом продукции (см. Рисунок 1).

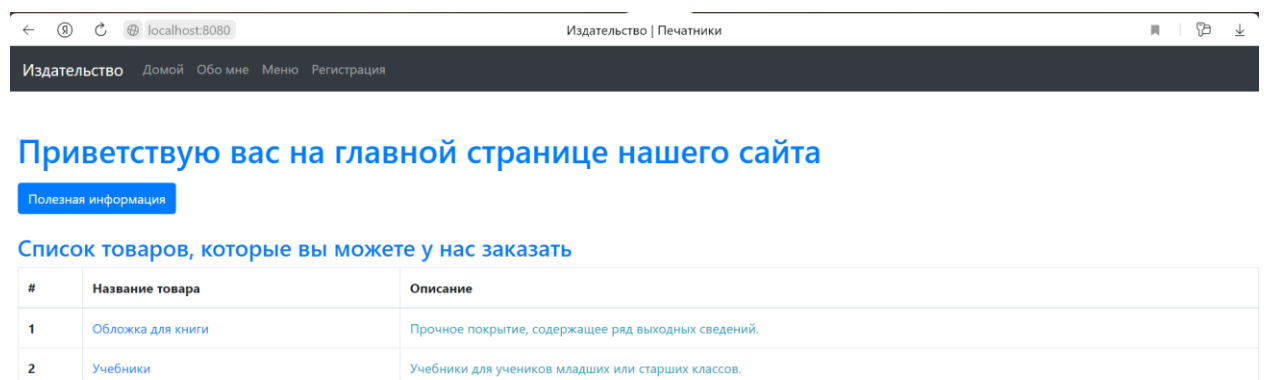


Рисунок 1 – Главная страница сайта

При отсутствии продукции окно страницы будет выглядеть следующим образом (см. Рисунок 2).

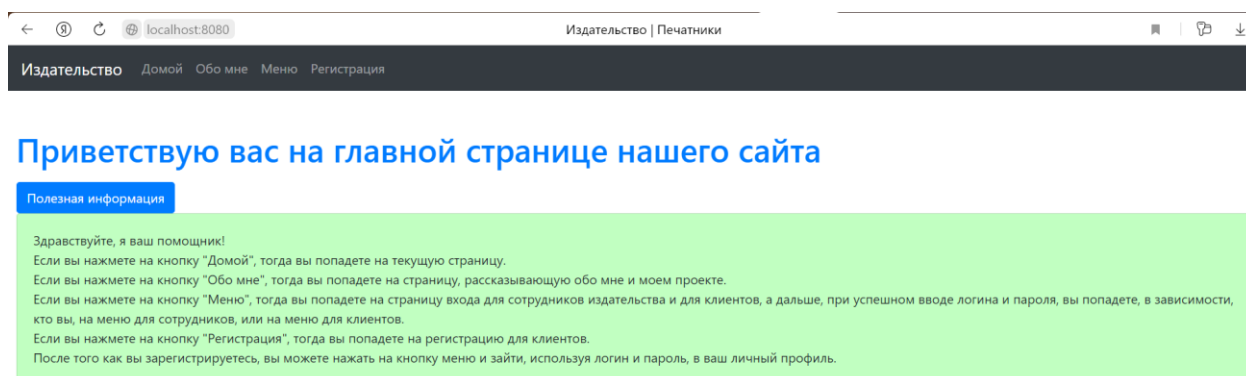


Рисунок 2 – Главная страница сайта при отсутствии продукции

Была разработана панель навигации, расположенная в верхней части экрана. Там есть такие кнопки, как «Домой» – позволяет попасть обратно на главную страницу <http://localhost:8080/>, «Обо мне» – позволяет показать информацию об авторе данной программы (см. Рисунок 3).

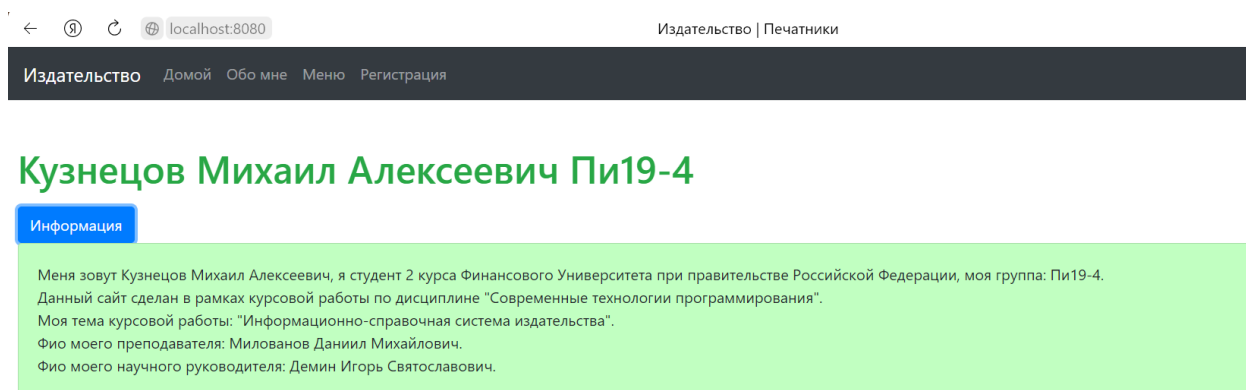


Рисунок 3 – Страница об авторе

При нажатии на кнопку «Меню» откроется авторизация для перехода на личный кабинет сотрудника или клиента издательства. Если пользователь не авторизован, то при нажатии на данную кнопку попадет на страницу авторизации (см. Рисунок 4). Более подробно функционал данной кнопки будет рассмотрен в дальнейших разделах курсовой работы.

← Я ↻ localhost:8080

Издательство Домой Обо мне Меню Регистрация

Логин: Admin

Пароль: •

Sign in

Рисунок 4 – Авторизация

При нажатии на кнопку «Регистрация», откроется форма регистрации клиента (см. Рисунок 5).

Издательство Домой Обо мне Меню Регистрация

Страница регистрации

Введите ваше имя

Введите вашу фамилию

Введите название вашей организации

Информация для входа

Введите ваш логин

Введите ваш пароль

Добавить

Рисунок 5 – страница регистрации

Если клиентом будут правильно заполнены поля регистрации, то ему предоставится доступ к личному кабинету (см. Рисунок 6). Это необходимо для того, чтобы клиент мог создавать нужные ему заказы у издательства. Если клиент неправильно заполнит поля регистрации или выберет существующий

в системе логин, то ему автоматически поступит сообщение об ошибке (см. Рисунок 7).

Страница регистрации

Давид
Чепенко
ООО МосОбразование

Информация для входа

Admin
1

Добавить

Рисунок 6 – Заполненная страница регистрации

Страница регистрации

Такой логин уже существует, попробуйте придумать новый!

Введите ваше имя
Введите вашу фамилию
Введите название вашей организации

Информация для входа

Введите ваш логин
Введите ваш пароль

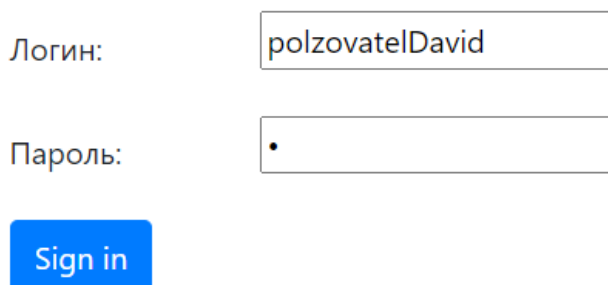
Добавить

Рисунок 7 – Сообщение об ошибке при регистрации

Опишем процесс авторизации. Авторизация существует в двух вариантах – для клиента и для сотрудника.

Клиент имеет возможность авторизоваться в системе для перехода на свой личный кабинет только после того, как зарегистрировался в системе.

Для авторизации клиенту необходимо ввести свой логин и пароль, созданный им на странице регистрации (см. Рисунок 8).



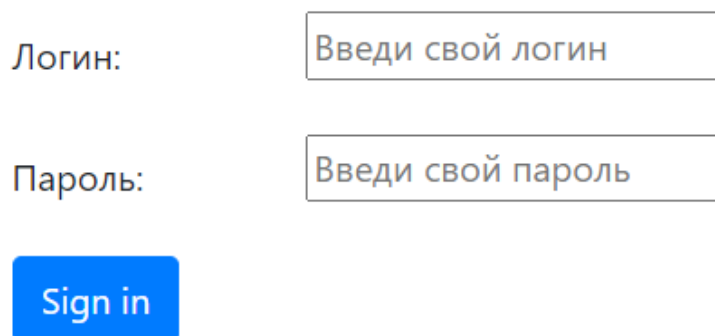
Логин:

Пароль:

[Sign in](#)

Рисунок 8 – Заполненная страница авторизации

При вводе неправильных данных пользователь получит сообщение об ошибке (см. Рисунок 9).



Логин:

Пароль:

[Sign in](#)

Вы ввели не тот логин/пароль

Рисунок 9 – Сообщение об ошибке при авторизации

Если пользователь введет свои данные корректно, то ему будет предоставлен доступ в его личный кабинет (см. Рисунок 10).

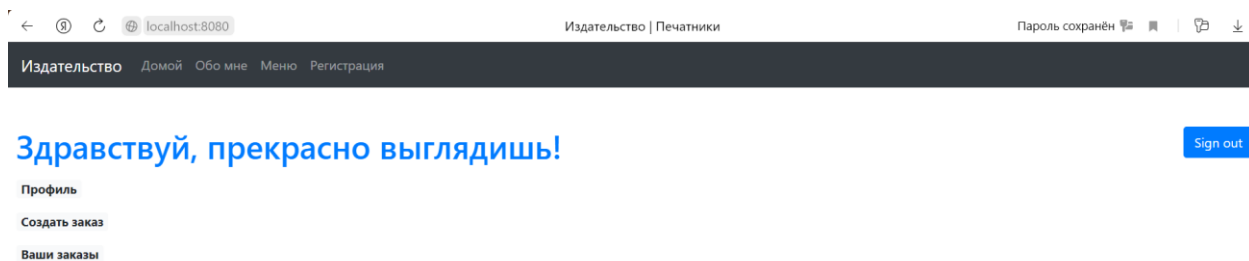


Рисунок 20 – Личный кабинет клиента

Клиенту доступны некоторые функции, такие как: «Sign out» - при нажатии на нее произойдет выход из личного кабинета текущего пользователя, после чего автоматически откроется форма авторизации (см. Рисунок 11).

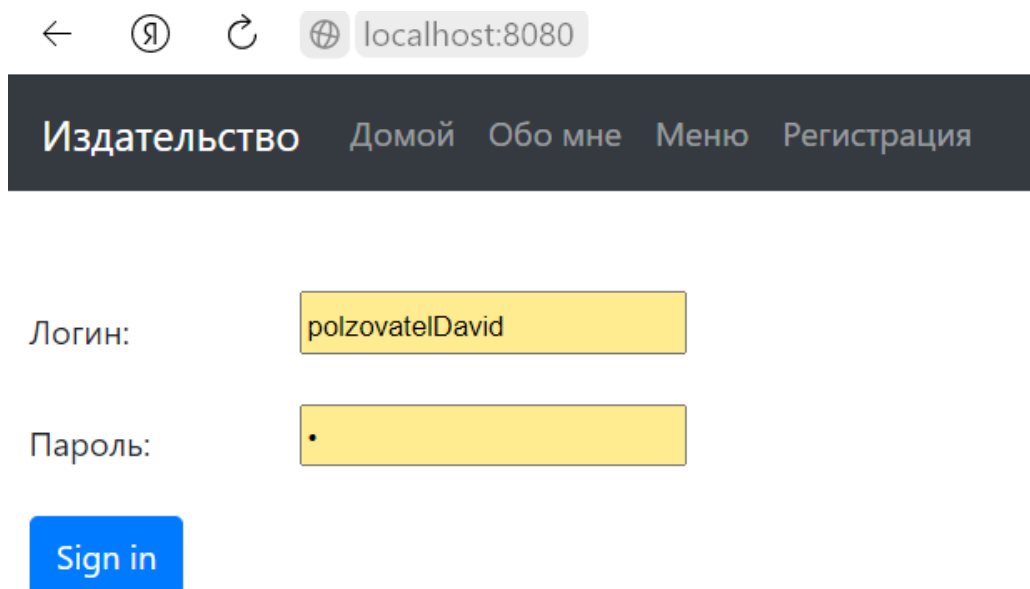


Рисунок 31 – Окно авторизации при выходе из личного кабинета

Так же существует кнопка «Профиль», при нажатии на которую клиент сможет увидеть свои данные, введенные им на странице регистрации (см. Рисунок 12).

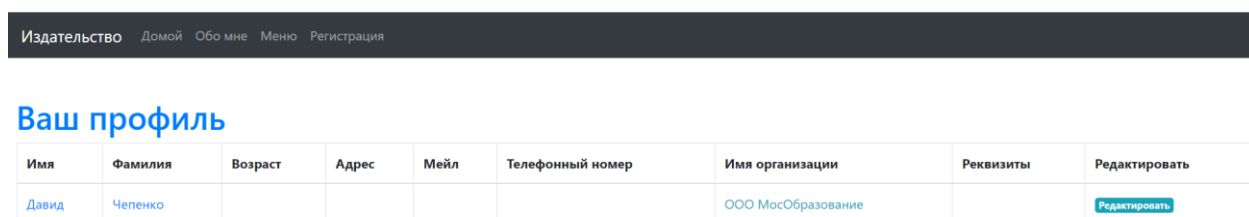


Рисунок 42 – Профиль клиента

Предусмотрена возможность редактирования данных. При нажатии на кнопку «Редактировать», пользователь попадет на страницу редактирования данных своего профиля (см. Рисунок 13).

Рисунок 53 – Страница редактирования профиля

Клиент может предоставить о себе следующую информацию: адрес, электронная почта, телефонный номер, реквизиты. Внешний вид формы с заполненными полями представлен на Рисунке 14.

Рисунок 64 – Заполненная страница редактирования профиля

При нажатии на кнопку сохранить, данные клиента будут изменены на те, что были введены им в полях редактирования (см. Рисунок 15).


Ваш профиль


Имя	Фамилия	Возраст	Адрес	Мейл	Телефонный номер	Имя организации	Реквизиты	Редактировать
Давид	Чепенко	18	Строгино, Москва	npb2324@mail.ru	89163183249	ООО МосОбразование	42567776664444	Редактировать

Рисунок 75 – Профиль клиента

Если клиент нажмет на кнопку «Создать заказ», откроется форма создания нового заказа (см. Рисунок 16).

Введите количество товара

дд.мм.гггг 


Выберите продукт, который вы хотите заказать Обложка для книги 


[Создать заказ](#)

Рисунок 86 – Страница заказа

В данной форме клиенту необходимо указать нужный ему товар и его количество, а также выбрать дату доставки данного заказа (см. Рисунок 17).

10

30.05.2021 

Выберите продукт, который вы хотите заказать Учебники 

[Создать заказ](#)

Рисунок 97 – Заполненная страница заказа

После нажатия на кнопку «Создать заказ» пользователь попадет на страницу своих заказов (см. Рисунок 18).

Издательство [Домой](#) [Обо мне](#) [Меню](#) [Регистрация](#)

Ваши заказы

#	Ваши реквизиты	Срок выполнения заказа	Цена	Количество	Статус	Заказанный товар	Редактировать
1	4256777664444	30.05.2021 3:00:00		10	Не обработан	Учебники	Редактировать

Рисунок 108 – Страница заказов клиента

При создании нового заказа цену определяет сотрудник издательства. Клиент может видеть статус своего заказа. Отдельно стоит рассмотреть систему статусов:

- 1) Новым заказам присваивается статус «Не обработан».
- 2) При рассмотрении и утверждении заказа сотрудником компании, статус изменится на «Заказ в работе».
- 3) При отсутствии продукции, распаде текущей типографии, отказ от услуг автора, заказу присвоится статус «Заказ отклонен» с последующим

возвратом денежных средств на указанный в договоре расчетный счет клиента.

Клиент имеет право изменять свой заказ только в том случае, если статус данного заказа «Не обработан» (см. Рисунок 19).

Редактирование заказа

Ваши реквизиты	Срок выполнения заказа	Цена	Количество	Статус	Заказанный товар
42567776664444	30.05.2021 3:00:00		<input type="text" value="10"/>	Не обработан	Учебники
<input type="button" value="Сохранить"/>					

Рисунок 119 – Страница редактирования заказа клиента

При нажатии на кнопку сохранения, данные заказа будут изменены на те, что были введены клиентом в полях редактирования заказа (см. Рисунок 20).

Ваши заказы

#	Ваши реквизиты	Срок выполнения заказа	Цена	Количество	Статус	Заказанный товар	Редактировать
1	42567776664444	30.05.2021 3:00:00		20	Не обработан	Учебники	<input type="button" value="Редактировать"/>

Рисунок 20 – Страница заказов клиента

Возможности клиента разобраны, теперь перейдем к возможностям сотрудников. Перед переходом в личный кабинет сотрудник должен пройти авторизацию (см. Рисунок 21).

Логин:	<input type="text" value="Admin"/>
Пароль:	<input type="password"/>
<input type="button" value="Sign in"/>	

Рисунок 21 – Страница авторизации

Если сотрудник введет свои данные в окне авторизации корректно, ему будет предоставлен доступ в личный кабинет (см. Рисунок 22).

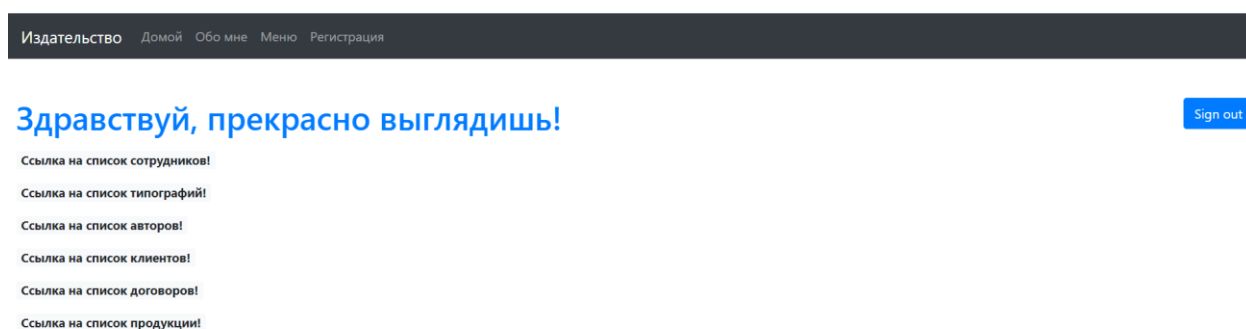


Рисунок 22 – Личный кабинет сотрудника

Сотруднику издательства доступны некоторые функции:

- 1) «Sign out» - так же, как и в личном кабинете клиента, при нажатии на эту кнопку будет совершен выход из личного кабинета текущего пользователя.
- 2) «Ссылка на список сотрудников» - при нажатии пользователь попадет на список всех сотрудников издательства (см. Рисунок 23).

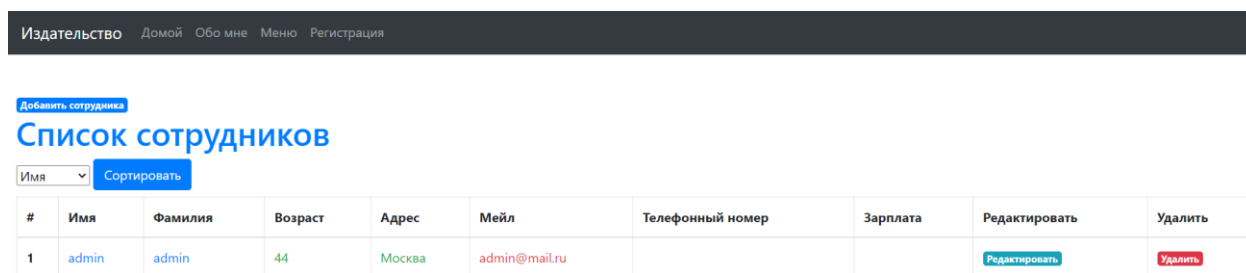


Рисунок 23 – Список сотрудников издательства

За добавление сотрудников отвечает администратор, он может добавлять, удалять, изменять информацию о сотрудниках и ставить им роли. При нажатии на кнопку «Добавить сотрудника» появится форма добавления сотрудника. Если логин будет не уникальным, администратору поступит сообщение об ошибке (см. Рисунок 24).

Добавление нового сотрудника

Такой логин уже существует, попробуйте придумать новый!

Введите возраст сотрудника:

Введите имя сотрудника:

Введите фамилию сотрудника:

Введите адрес сотрудника:

Введите почту сотрудника:

Введите телефон сотрудника:

Введите зарплату сотрудника:

Информация для входа

Задайте логин для входа:

Задайте пароль для входа:

Рисунок 24 – Добавление нового сотрудника

Автоматически при нажатии на кнопку «Добавить», администратор попадет на список сотрудников издательства (см. Рисунок 25).

Издательство Домой Обо мне Меню Регистрация									
Добавить сотрудника									
<h2>Список сотрудников</h2>									
<input type="text" value="Имя"/> <input type="button" value="Сортировать"/>									
#	Имя	Фамилия	Возраст	Адрес	Мейл	Телефонный номер	Зарплата	Редактировать	Удалить
1	admin	admin	44	Москва	admin@mail.ru			<input type="button" value="Редактировать"/>	<input type="button" value="Удалить"/>
2	Михаил	Кузнецов	22	Коммунарка, Москва	mKuznetsov@mail.ru	+79852840507	188 000	<input type="button" value="Редактировать"/>	<input type="button" value="Удалить"/>

Рисунок 25 – Список сотрудников издательства

В данной форме администратор может редактировать, сортировать и удалять сотрудников. При нажатии на кнопку «Редактировать», откроется форма редактирования выбранного сотрудника (см. Рисунок 26).

Редактирование сотрудника

Имя	Фамилия	Возраст	Адрес	Мейл	Телефонный номер	Зарплата
Михаил	Кузнецов	22	Коммунарка, Москва	mKuznetsov@mail.ru	+79852840507	188000

☒ USER

☐ ADMIN

☐ TYPOGRAPHY_MANAGER

☐ ACCOUNTANT

☐ CLIENT_MANAGER

☐ AUTHOR_MANAGER

☐ PRODUCTION_MANAGER


[Сохранить](#)

Рисунок 26 – Страница редактирования сотрудника

Так же доступна сортировка по имени, фамилии и зарплате (см. Рисунок 27).

[Добавить сотрудника](#)

Список сотрудников

Имя  [Сортировать](#)

Имя

Фамилия

Зарплата


	Имя	Фамилия	Возраст
--	-----	---------	---------

Рисунок 27 – Варианты сортировок

Сделаем одну сортировку по зарплате (см. Рисунок 28).

[Добавить сотрудника](#)

Список сотрудников

Имя  [Сортировать](#)

#	Имя	Фамилия	Возраст	Адрес	Мейл	Телефонный номер	Зарплата	Редактировать	Удалить
1	Михаил	Кузнецов	22	Коммунарка, Москва	mKuznetsov@mail.ru	+79852840507	188 000	Редактировать	Удалить
2	admin	admin	44	Москва	admin@mail.ru		189 000	Редактировать	Удалить

Рисунок 28 – Список сотрудников издательства после сортировки по зарплате

Так же есть возможность удаления сотрудника. К примеру, удалим сотрудника с именем «Михаил» (см. Рисунок 29).

[Добавить сотрудника](#)

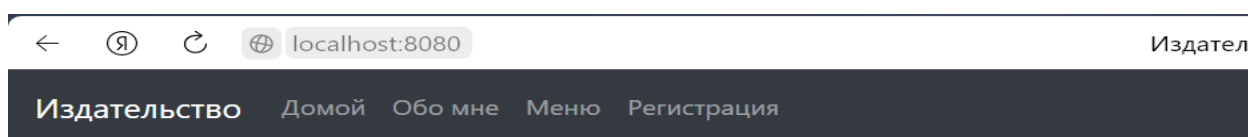
Список сотрудников

Имя Сортировать

#	Имя	Фамилия	Возраст	Адрес	Мейл	Телефонный номер	Зарплата	Редактировать	Удалить
1	admin	admin	44	Москва	admin@mail.ru		189 000	Редактировать	Удалить

Рисунок 29 – Удаление сотрудника

При нажатии на кнопку «Ссылка на список типографий» сотруднику будет предоставлена возможность перехода на список всех типографий. При отсутствии типографий окно приложения будет выглядеть следующим образом (см. Рисунок 30).



[Добавить типографию](#)

Нет типографий

Рисунок 30 – Страница типографий при отсутствии типографий

При нажатии на кнопку «Добавить типографию», появится форма добавления новой типографии (см. Рисунок 31).

Добавление новой типографии

Введите название типографии: Солнечный Ветер

Введите реквизиты типографии: 4256 4444 6644

Введите телефонный номер типографии: +79037928822

Введите адрес типографии: Солнечногорск, Москов

[Добавить](#)

Рисунок 31 – Добавление новой типографии

После добавления новой типографии пользователь попадет на список типографий (см. Рисунок 32). После этого, все остальные действия станут однотипными как с добавлением, редактированием и удалением сотрудника.

[Добавить типографию](#)

Список типографий

#	Имя	Реквизиты	Телефонный номер	Адрес	Продукция	Редактировать	Удалить
1	Солнечный Ветер	4256 4444 6644	+79037928822	Солнечногорск, Московская область		Редактировать	Удалить

Рисунок 32 – Страница типографий

При нажатии на кнопку «Ссылка на список авторов», появится список всех авторов с возможностями удаления, редактирования и добавления новых (см. Рисунок 33).

[Добавить автора](#)

Список авторов

Имя [Сортировать](#)

#	Имя	Фамилия	Возраст	Адрес	Мейл	Телефонный номер	Редактировать	Удалить
1	Константин	Кузин	34	Санкт-Петербург, Россия	kkuzin81@gmail.com	+79062367282	Редактировать	Удалить

Рисунок 33 – Страница авторов

При нажатии на кнопку «Ссылка на список клиентов», появится список всех клиентов издательства, редактировать и удалять клиентов нельзя (см. Рисунок 34).

Список наших клиентов

#	Имя	Фамилия	Возраст	Адрес	Мейл	Телефонный номер	Имя организации	Реквизиты
1	Давид	Чепенко	18	Строгино, Москва	nrb2324@mail.ru	89163183249	ООО МосОбразование	42567776664444

Рисунок 34 – Страница всех клиентов издательства

При нажатии на кнопку «Ссылка на список договоров» пользователь попадает на список всех заказов, которые сделали клиенты (см. Рисунок 35).

Список договоров

Цена Сортировать

Максимальная цена заказа Выбрать

Необработанные заказы

#	Реквизиты клиента	Реквизиты издателя	Дата заключения договора	Цена	Количество	Название товара	Статус	Типография	Автор	Редактировать
1	42567776664444		30.05.2021 3:00:00		20	Учебники	Не обработан	Нет типографий	Нет авторов	Редактировать

Обработанные заказы

#	Реквизиты клиента	Реквизиты издателя	Дата заключения договора	Цена	Количество	Название товара	Статус	Типография	Автор	Редактировать
---	-------------------	--------------------	--------------------------	------	------------	-----------------	--------	------------	-------	---------------

Отмененные заказы

#	Реквизиты клиента	Реквизиты издателя	Дата заключения договора	Цена	Количество	Название товара	Статус	Типография	Автор	Редактировать
---	-------------------	--------------------	--------------------------	------	------------	-----------------	--------	------------	-------	---------------

Рисунок 35 – Страница всех заказов

Как было написано ранее, заказы клиента делятся на разные таблицы в зависимости от их статуса. При нажатии на кнопку редактировать откроется форма редактирования договора/заказа клиента (см. Рисунок 36).

Редактирование договора

Реквизиты клиента Срок выполнения заказа Цена Количество Автор Типография Название товара

42567776664444 30.05.2021 3:00:00 20 Константин Солнечный Ветер Учебники

[Сохранить](#)

Рисунок 36 – Страница редактирования договора

При нажатии на кнопку сохранения, заказ автоматически попадет в таблицу обработанных заказов (см. Рисунок 37).

Список договоров

Цена Сортировать

Максимальная цена заказа Выбрать

Необработанные заказы

#	Реквизиты клиента	Реквизиты издателя	Дата заключения договора	Цена	Количество	Название товара	Статус	Типография	Автор	Редактировать
---	-------------------	--------------------	--------------------------	------	------------	-----------------	--------	------------	-------	---------------

Обработанные заказы

#	Реквизиты клиента	Реквизиты издателя	Дата заключения договора	Цена	Количество	Название товара	Статус	Типография	Автор	Редактировать
1	42567776664444	4256 4444 6644	30.05.2021 3:00:00	10 300	20	Учебники	Заказ в работе	Солнечный Ветер	Константин	Редактировать

Отмененные заказы

#	Реквизиты клиента	Реквизиты издателя	Дата заключения договора	Цена	Количество	Название товара	Статус	Типография	Автор	Редактировать
---	-------------------	--------------------	--------------------------	------	------------	-----------------	--------	------------	-------	---------------

Рисунок 37 – Страница всех заказов

Так же, есть возможность анализировать заказы клиентов: сотрудник издательства может найти самые дорогие и дешевые заказы, существует

возможность поиска средней цены и нахождения выручки предприятия (см. Рисунок 38).

Рисунок 38 – Варианты анализа заказов

Выберем, для примера, аналитику по максимальной цене заказа. При нажатии на кнопку, сотруднику будет выведена информация о самом дорогом заказе (см. Рисунок 39).

Список договоров

Цена

Максимальная цена заказа

Ответ: 10 300

Необработанные заказы

#	Реквизиты клиента	Реквизиты издателя	Дата заключения договора	Цена	Количество	Название товара	Статус	Типография	Автор	Редактировать
---	-------------------	--------------------	--------------------------	------	------------	-----------------	--------	------------	-------	---------------

Обработанные заказы

#	Реквизиты клиента	Реквизиты издателя	Дата заключения договора	Цена	Количество	Название товара	Статус	Типография	Автор	Редактировать
1	4256777666444	4256 4444 6644	30.05.2021 3:00:00	10 300	20	Учебники	Заказ в работе	Солнечный Ветер	Константин	<input type="button" value="Редактировать"/>

Отмененные заказы

#	Реквизиты клиента	Реквизиты издателя	Дата заключения договора	Цена	Количество	Название товара	Статус	Типография	Автор	Редактировать
---	-------------------	--------------------	--------------------------	------	------------	-----------------	--------	------------	-------	---------------

Рисунок 39 – Страница заказов с выводом ответа по выбранному варианту анализа

При нажатии на кнопку «Ссылка на список продукции», сотруднику будет доступен список существующих и доступных видов продукции для заказа у издательства (см. Рисунок 40).

Список продукции

#	Имя	Описание	Список типографий	Редактировать	Удалить
1	Обложка для книги	Прочное покрытие, содержащее ряд выходных сведений.		<input type="button" value="Редактировать"/>	<input type="button" value="Удалить"/>
2	Учебники	Учебники для учеников младших или старших классов.		<input type="button" value="Редактировать"/>	<input type="button" value="Удалить"/>

Рисунок 40 – Страница продукции

При нажатии на кнопку «Редактировать», откроется форма редактирования выбранного вида продукции (см. Рисунок 41)

Редактирование продукции

Имя

Описание

Обложка для книги

Прочное покрытие, сод

☐ Солнечный Ветер

Сохранить

Рисунок 41 – Страница редактирования продукции

Есть возможность выбрать типографию, которая будет печатать данный вид продукции (см. Рисунок 42).

Редактирование продукции

Имя

Описание

Обложка для книги

Прочное покрытие, сод

☒ Солнечный Ветер

Сохранить

Рисунок 42 – Заполненная страница редактирования продукции

При сохранении изменений пользователь попадет на страницу списка продукции. Так же сотрудник сможет увидеть, какие типографии печатают данный вид продукции (см. Рисунок 43).

Добавить продукцию

Список продукции

#	Имя	Описание	Список типографий	Редактировать	Удалить
1	Обложка для книги	Прочное покрытие, содержащее ряд выходных сведений.	Солнечный Ветер	Редактировать	Удалить
2	Учебники	Учебники для учеников младших или старших классов.		Редактировать	Удалить

Рисунок 43 – Страница продукции

При нажатии на кнопку «Ссылка на список типографий», сотрудник сможет заметить, что в поле «Продукция», у типографии «Солнечный Ветер»,

добавилась продукция, которую данная типография печатает (см. Рисунок 44).
Процесс выбора типографии показан на Рисунке 42.

Добавить типографию

Список типографий

#	Имя	Реквизиты	Телефонный номер	Адрес	Продукция	Редактировать	Удалить
1	Солнечный Ветер	4256 4444 6644	+79037928822	Солнечногорск, Московская область	Обложка для книги	Редактировать	Удалить

Рисунок 44 – Страница типографий

4.3. Состав приложения

Код программы подразделяется на часть сервера и часть интерфейса.
Ниже показаны модели Базы данных для данного проекта: логическая (см. Рисунок 45) и реляционная (см. Рисунок 46).

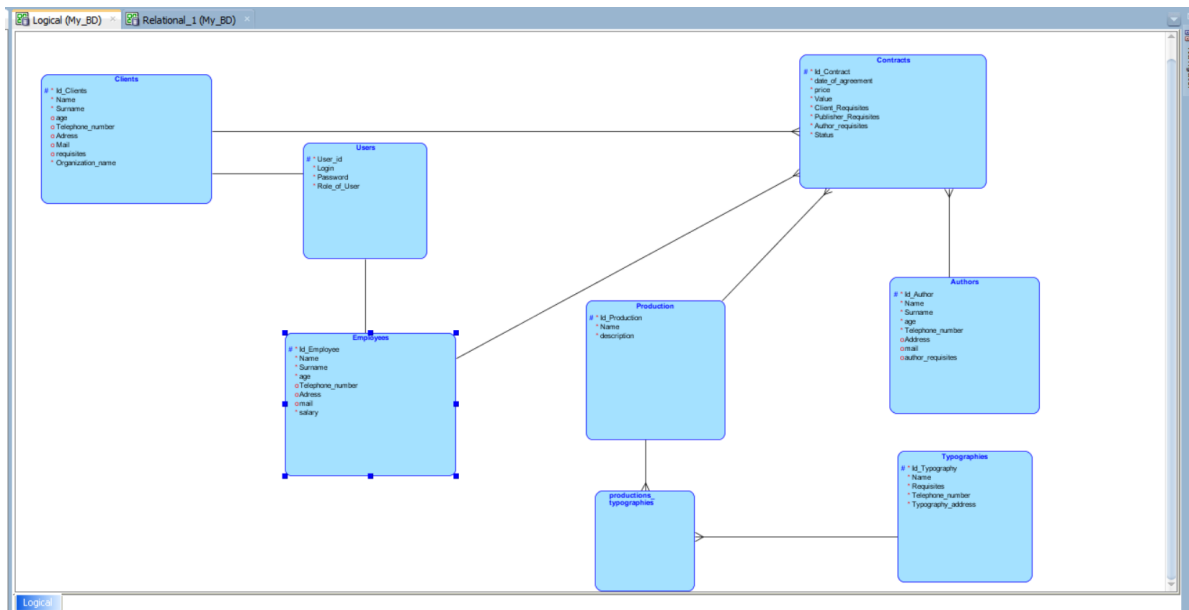


Рисунок 45 – Логическая модель базы данных

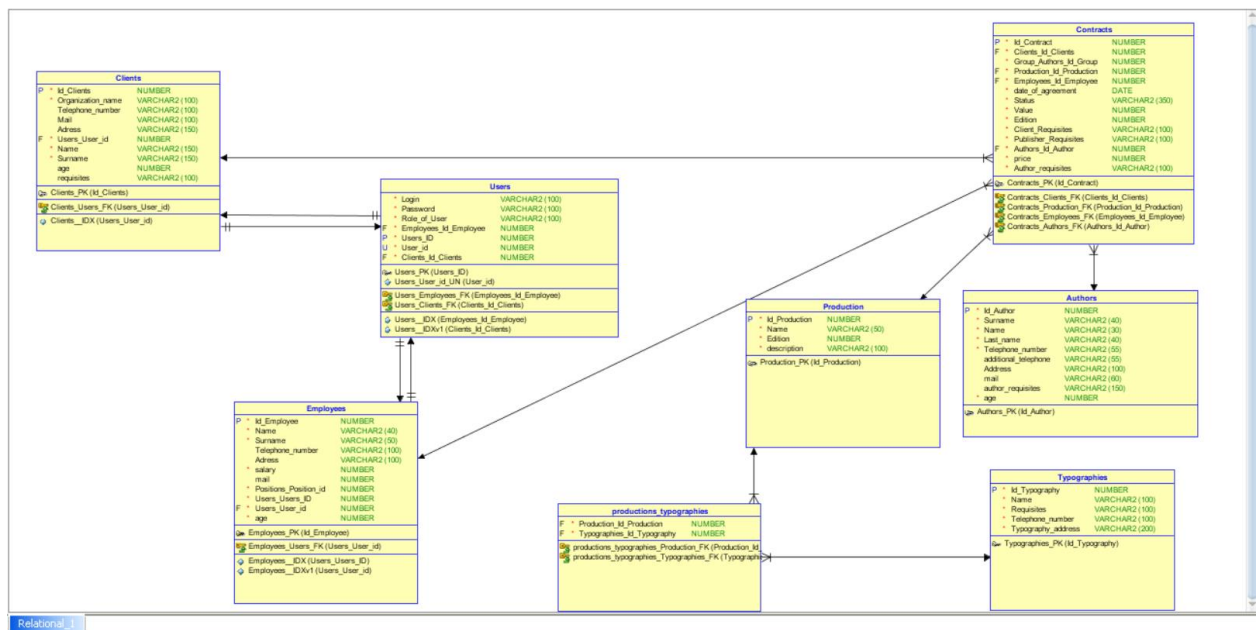


Рисунок 46 – Реляционная модель базы данных

Расскажем подробнее про каждую из существующих таблиц:

- 1) Clients – клиенты,
- 2) Users – учетные записи для клиентов или для сотрудников
- 3) Employees – сотрудники
- 4) Contracts – договоры
- 5) Authors – авторы
- 6) Typographies – типографии
- 7) Production – товары.

Clients имеет в качестве внешнего ключа Clients_Users_Fk (Foreign Key), это необходимо для того, чтобы была осуществлена связь с таблицей Users. По тому же самому принципу таблица Employees связана с Users внешним ключом Employee_Users_Fk. Таблица Productions связана с таблицей Typographies связью «Многие ко Многим», чтобы нормализовать отношения между ними, их связь друг с другом будет храниться в таблице production_typographies, в данной таблице будут находиться внешние ключи Production_id и Typography_id. У таблицы Contracts существует несколько внешних ключей: Contracts_Clients_Fk – для связи с таблицей Clients, Contracts_Productions_Fk – для связи с таблицей Production,

Contracts_Employees_Fk – для связи с таблицей Employees,
Contracts_Authors_Fk – для связи с таблицей Authors.

Далее будет представлена структура данного приложения (см. Рисунок 47).

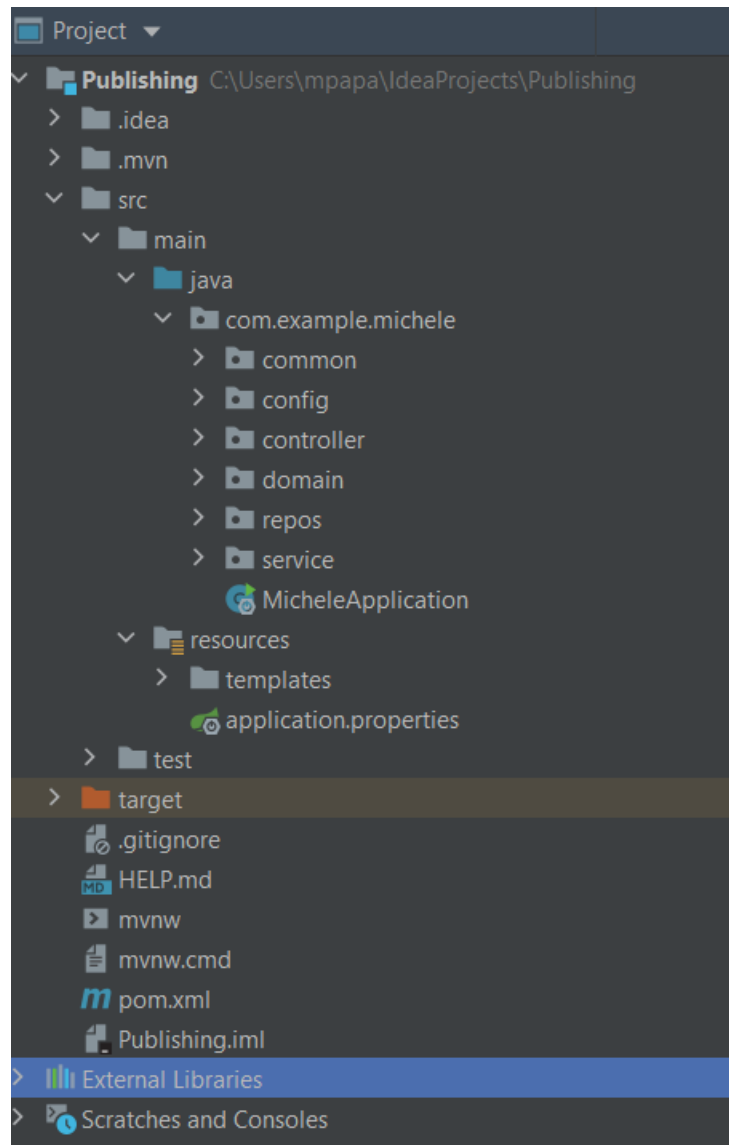


Рисунок 47 – Структура приложения

Рассмотрим package «com.example.michele» - в package common находятся общий класс Person необходимый для уменьшения избыточности кода. Для классов Сотрудники, Клиенты и Авторы не нужно прописывать дублирующие поля(name, surname, age, telephone_number, address, email), достаточно просто унаследовать данные классы от класса Person.

В `package config` находятся два класса, которые осуществляют авторизацию пользователей и переход на страницу входа на страницу логина. В `package controller` лежат самые важные части программы – контроллеры, они проверяют запросы пользователя, связываются с Базой Данных, выполняет нужное действие, потом получает от БД данные, далее показывают их клиенту, в результате чего клиент может видеть результат того, или иного запроса. Так же существует контроллер, который улавливают ошибки и не дает сайту аварийно завершать свою работу. В следующем `package domain` находятся объекты для работы сервера.

В `package repos` находятся интерфейсы для работы с БД при помощи `JpaRepository`. В папке `resources` находятся `.ftl` файлы, которые отвечают за отображение пользователю интерфейса на сайте.

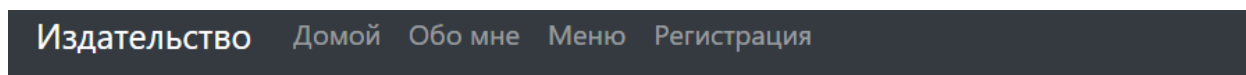
5. Назначение и состав классов программы

Начнем разбор состава классов программы с класса `Person` из `package common`. Данный класс необходим для того, чтобы уменьшить избыточность кода. Все те классы, которые наследуются от `Person`, а именно: `Author`, `Client`, `Employer` имеют одинаковые поля: имя, фамилия, возраст, телефонный номер, почта и адрес. Для того, чтобы не писать в каждом классе те же самые поля, было принято решение создать данный класс.

Далее, в `package config` находятся два класса: `WebSecurityConfig` и `MvcConfig`, они нужны для реализации функционала, необходимого для работы авторизации и аутентификации пользователей. Ключевым методом является метод `configure`, в котором устанавливаются параметры для авторизации, а именно – какие страницы доступны пользователям без авторизации, какая страница является страницей авторизацией, какой запрос присылается при ошибке в авторизации (см. Рисунок 9), так же возможность выхода из учетной записи включается в данном классе. Существует еще один метод – `configure`, только в качестве параметра он включает в себя

AuthenticationManagerBuilder auth, данный метод показывает, каким образом будет проходить авторизация.

В следующем package controller расположены контроллеры, служащие для того, чтобы с помощью аннотаций @GetMapping, @PostMapping обрабатывать запросы от пользователя. Контроллеры имеют связи с БД и проверяют правильность данных запроса пользователя перед внесением в БД изменений. Также, контроллеры возвращают файлы, которые отображают пользователю интерфейс на сайте. В некоторых контроллерах присутствует аннотация @PreAuthorize("hasAnyAuthority()") служащая для того, чтобы проверять по ролям пользователей доступ к определенным страницам приложения. Например, если сотрудник, чья роль не ADMIN нажмет на кнопку «Ссылка на список сотрудников», то ему отправится сообщение о несоответствии ролей (см. Рисунок 48).



Упс... Что-то пошло не так!

Доступ запрещен

Рисунок 48 – Вывод сообщения пользователю о несоответствии ролей

Рассмотрим работу метода редактирования на примере редактирования Продукции (см. выше Рисунок 41). Данный метод имеет аннотацию @GetMapping и показывает пользователю форму редактирования. В данной форме сотрудник может, к примеру, выбрать типографию для выбранной продукции, изменить ее описание или оставить все поля без изменений. Как только пользователь нажмет кнопку сохранить, производится отправка формы методом POST. Данный запрос обрабатывается соответствующим методом в контроллере, который изменит, по заданным пользователем полям,

информацию в Базе Данных о выбранной продукции и покажет пользователю форму с измененной им продукцией (см. Рисунок 43).

В классе `AuthorController` присутствуют методы для добавления, изменения, удаления и просмотра авторов.

В классе `ClientController` содержатся методы для отображения у сотрудников издательства данных обо всех клиентах.

В классе `ClientUserController` присутствуют методы для отображения личного кабинета клиента, где он может посмотреть и изменить свои данные, создать заказ и увидеть список своих заказов.

Класс `ContractController` позволяет сотрудникам издательства смотреть, проставлять статусы, находить авторов и типографии для выполнения заказов клиента.

Класс `EmployerController` необходим для отслеживания сотрудников. В данном классе присутствуют методы для добавления, редактирования, удаления сотрудников издательства, класс доступен только для администратора. С помощью аннотации `@PreAuthorize("hasAuthority('ADMIN')")` доступ к данной странице имеет только администратор издательства (или же, генеральный директор компании). Поэтому, если любой другой сотрудник, кроме администратора, попытается нажать на кнопку «Ссылка на список сотрудников», то, как было сказано выше, ему отправится сообщение о несоответствии ролей (см. Рисунок 48).

Отдельно рассмотрим класс `ExceptionHandler`. В данном классе обрабатывается любая ошибка, возникающая на сайте. Класс реализует интерфейс `ErrorController`, определяя в нем метод `getErrorPath`, возвращающий путь, по которому возможно обработать ошибку.

Далее, ошибку обрабатывает метод `handleError`, имеющий аннотацию `@RequestMapping("/error")`. Суть метода состоит в том, что он определяет код ошибки, и в зависимости от него сообщает пользователю о типе произошедшей ошибки.

К примеру, пользователь отправляет неправильный запрос на сайт. Если бы данного класса не существовало, то программа бы завершилась аварийно. Но так как данный класс присутствует, то при таком запросе (см. Рисунок 49), будет определен код ошибки (404) и выведется на экран пользователю сообщения «Ресурс не найден».

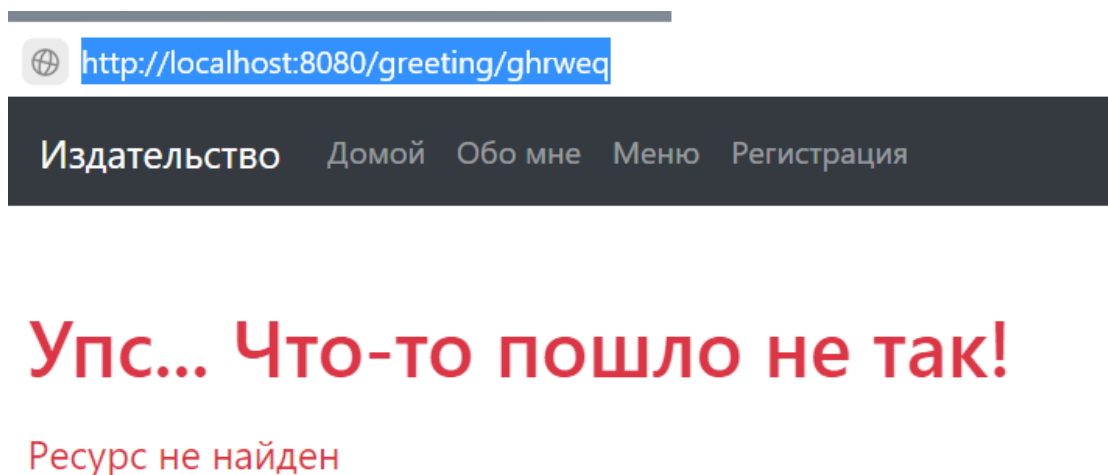


Рисунок 49 – Вывод сообщения пользователю об ошибке

В классе `LoginController` реализуется функционал для отображения страницы авторизации. Кроме того, в этом классе производится обработка ситуации, когда пользователь ввел не тот логин или пароль. В этом случае выводится предупреждения пользователю об ошибке, на экран. (см. Рисунок 9)

Класс `MainController` необходим для отображения главной страницы сайта. В данном классе находится метод `greeting`, который отображает личный кабинет пользователя в зависимости от роли данного пользователя, пример: вход только в личный кабинет клиента для клиента, или вход только в личный кабинет сотрудника для сотрудника издательства.

В классе `ProductionController` содержатся методы, для отображения, удаления, добавления и редактирования продукции.

В классе `RegistrationController` присутствуют методы для регистрации нового клиента. Стоит отметить то, что недопустимы несколько пользователей с

одинаковым логином. В случае, если пользователь захочет зарегистрироваться по логину, который уже существует, он получит сообщение об ошибке (см. Рисунок 9).

В классе `TypographyController` содержатся методы для отображения, удаления, добавления и редактирования типографий.

Перейдем в `package domain`. Каждый класс, который находится в данном `package`, представляет собой отдельную таблицу в Базе Данных и обозначается аннотацией `@Entity` – сущность. Аннотация `@Table(name=)` говорит о названии данной таблицы в базе данных. Опишем одну из сущностей, работа остальных будет аналогична рассмотренной сущности.

Описывать будем сущность `Author`. В ней содержатся, так же, как и в Базе Данных (см. Рисунок 45,46) поля: `id` – означает, что данное поле является первичный ключом для данной таблицы. Для этого поля установлены две аннотации `@Id` и `@GeneratedValue(strategy = GenerationType.AUTO)`. Аннотация `@Id` означает, что данной поля является уникальным идентификатором. Аннотация `@GeneratedValue(strategy = GenerationType.AUTO)` означает, что значения поля первичного ключа будет определено автоматически.

Кроме того, класс `Author` унаследован от класса `Person`. Соответственно, в нем присутствуют поля и методы, характерные для `Person`. Также, присутствует одно собственное поле `author_requisites`, предназначенное для хранения реквизитов автора. Присутствуют геттеры и сеттеры для того, чтобы можно было извлекать значения или присваивать новые значения полям.

В некоторых сущностях, таких, как `Client`, `Employer`, `Contract` и др. присутствуют аннотации, характеризующие тип связей между таблицами. Такими аннотациями являются `@OneToOne`, `@OneToMany`, `@ManyToOne` и `@ManyToMany`. Они характеризуют связи «один-к-одному», «один-ко-многим», «многие-к-одному» и «многие-ко-многим» соответственно. Отметим, что так как связь «многие-ко-многим» реализуется с помощью

третьей сущности, данная аннотация позволяет создать промежуточную сущность автоматически.

В package `repos` хранятся интерфейсы каждой сущности из package `domain`. Каждый интерфейс является наследником интерфейса `JpaRepository`, что необходимо для работы с БД.

Рассмотрим репозиторий `AuthorRepo`. Остальные интерфейсы устроены аналогично. Интерфейс `AuthorRepo` позволяет находить из БД данные по любому полю из таблицы `author`. Например, существуют методы: `Author findById(Long id)` и `void deleteById(Long id)`.

Первый отвечает за поиск автора по указанному идентификатору, второй за удаление автора. Также существует метод `authorRepo.findAll()` извлечения всех данных из указанной таблицы `authors`.

Далее рассмотрим package `service`. В нем присутствует всего один класс – `UserService`, реализующий интерфейс `UserDetailsService`. Этот класс необходим для проверки авторизации пользователей.

Класс `MicheleApplication` запускает приложение.

В папке `resources/templates` находится набор элементов внешнего вида (в виде файлов) для каждой сущности для отображения пользователю интерфейса на сайте. Приведем список файлов, необходимых для отображения интерфейса сайта:

- 1) `addAuthor` - файл необходим для показа формы добавления авторов.
- 2) `aboutMe` – файл для показа формы информации об авторе проекта.
- 3) `addContract` – файл для добавления договора (у клиента).
- 4) `addEmployee` – файл для добавления сотрудника.
- 5) `addProduction` – файл для добавления продукции.
- 6) `addTypography` – файл для добавления типографии.
- 7) `author` – файл для отображения всех авторов.
- 8) `authorEdit` – файл для редактирования указанного автора.
- 9) `client` – файл для отображения всех клиентов.
- 10) `clientContractEdit` – файл для редактирования клиентов договора.

- 11) clientContracts – файл со списком договоров у клиента.
- 12) contract – файл со списком всех договоров.
- 13) contractEdit – файл для редактирования сотрудником договора.
- 14) editProfilClient – файл для редактирования клиентом своих данных.
- 15) employee – файл со списком всех сотрудников.
- 16) employeeEdit – файл редактирования сотрудника.
- 17) errorMessage – файл вывода ошибок сайта.
- 18) greeting1 – файл с возможностями клиентов.
- 19) greeting – файл с возможностями сотрудников.
- 20) login – файл для показа формы авторизации.
- 21) mainPage – файл для показа главной страницы.
- 22) production – файл для показа всех продуктов.
- 23) productionEdit – файл для редактирования продукции.
- 24) profilClient – файл для отображения клиенту данных его учетной записи.
- 25) registration – файл для регистрации клиентов.
- 26) typography – файл для показа всех типографий.
- 27) typographyEdit – файл для редактирования типографии.

В папке resources/templates/parts лежат 2 файла, один из которых отвечает за настройку формата, а второй за отображение верхней панели. Эти файлы используются остальными шаблонами для уменьшения избыточности кода.

За подключение к БД из проекта отвечает файл application.properties. В нем указывается адрес Базы Данных, имя пользователя – владельца базы, пароль и другие настройки.

Заключение

В результате проделанной работы было создано приложение, представляющее собою информационную систему для издательства. В приложении были реализованы функции добавления, удаления и редактирования различных объектов. Весь исходный код прокомментирован по стандарту JavaDoc.

Были получены практические навыки по программированию на языке высокого уровня Java с использованием фреймворка Spring. Кроме того, были получены навыки проектирования и взаимодействия с базами данных.

Список литературы

1. Козмина Ю., Харроп Р. Spring 5 для профессионалов. - Киев: Диалектика-Вильямс, 2019. - 1120 с.
2. Коузен К. Современный Java. Рецепты программирования. - М.: ДМК Пресс, 2018. - 274 с.
3. Мартин Роберт К. Чистый код. Создание анализ и рефакторинг. - СПб: Питер, 2019. - 464 с.
4. Прохоренок Н.А. JavaFX. - СПб: БХВ-Петербург, 2020. - 768 с.
5. Шилдт Г. Java. Полное руководство. - Киев: Диалектика, 2018. - 1488 с.
6. JPA: Хранение перечислений в базе данных [Электронный ресурс] //habr.com. URL: <https://habr.com/ru/post/77982/>
7. Baeldung [Электронный ресурс] //baeldung.com. URL: <https://www.baeldung.com/>
8. Many-To-Many Relationship in JPA [Электронный ресурс] //baeldung.com. URL: <https://www.baeldung.com/jpa-many-to-many>
9. Habr [Электронный ресурс] //habr.com. URL: <https://habr.com/ru/>
10. Отношение ManyToOne в Hibernate и Spring [Электронный ресурс] //sysout.ru. URL: <https://sysout.ru/otnoshenie-manytoone-v-hibernate-i-spring/>
11. HtmlBook [Электронный ресурс] //htmlbook.ru. URL: <http://htmlbook.ru/>
12. Sorting Query Results with Spring Data [Электронный ресурс] //baeldung.com. URL: <https://www.baeldung.com/spring-data-sorting>
13. Spring [Электронный ресурс] //spring.io. URL: <https://spring.io/>
14. Bootstrap: Colors [Электронный ресурс] //getbootstrap.com. URL: <https://getbootstrap.com/docs/4.0/utilities/colors/#color>
15. Bootstrap: Tables [Электронный ресурс] //getbootstrap.com. URL: <https://getbootstrap.com/docs/4.0/content/tables/>
16. Bootstrap: Collapse [Электронный ресурс] //getbootstrap.com. URL: <https://getbootstrap.com/docs/4.1/components/collapse/>
17. Bootstrap: Badges [Электронный ресурс] //getbootstrap.com. URL: <https://getbootstrap.com/docs/4.1/components/badge>