|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**КУРСОВАЯ РАБОТА**

по дисциплине: Шаблоны программных платформ языка Джава

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки:09.03.04 Программная инженерия

Тема: «Приложение «Магазин компьютерных комплектующих»»

Студент: Мурадов Натик Намигович

Группа: ИКБО-16-19

Работа представлена к защите 05.07.2021 (дата)\_\_\_\_\_\_\_\_\_\_\_\_\_/Мурадов Н.Н. /

(подпись и ф.и.о. студента)

Руководитель: Матчин Василий Тимофеевич, ст. преподаватель

Работа допущена к защите\_\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_\_\_\_\_/ Матчин В.Т./

(подпись и ф.и.о. рук-ля)

Оценка по итогам защиты: \_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших защиту)

М. РТУ МИРЭА. 2021 г.



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«МИРЭА - Российский технологический университет»**

**РТУ МИРЭА**



**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине: Шаблоны программных платформ языка Джава

по профилю: Программная инженерия

Студент: Мурадов Натик Намигович

Группа: ИКБО-16-19

Срок представления к защите:

Руководитель: Старший преподаватель ИиППО Матчин Василий Тимофеевич

**Тема: «**Разработка приложения «Магазин компьютерных комплектующих»

**Исходные данные**: Документация и средства языка Java.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала**:

1. Провести анализ предметной области; 2. Провести анализ сравнения с существующими аналогами; 3. Разработать клиентскую часть приложения (принятие и обработка запросов);

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО: \_\_\_\_\_\_\_\_\_\_\_ / Болбаков Р. Г. /, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Задание на КР выдал: \_\_\_\_\_\_\_\_\_\_\_\_\_ /Матчин В.Т. /, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Задание на КР получил: \_\_\_\_\_\_\_\_\_\_\_ / Мурадов Н.Н. /, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

УДК 004.4

Мурадов Н.Н.Приложение «Магазин компьютерных комплектующих» / Курсовая работапо дисциплине «Шаблоны программных платформ языка Джава» профиля «Разработка программных продуктов и проектирование информационных систем» направления профессиональной подготовки бакалавриата 09.03.04. «Программная инженерия» (4ый семестр) / руководитель В.Т. Матчин, ст. преподаватель / кафедра ИиППО Института ИТ РТУ МИРЭА.

Целью работы является проведение анализа предметной области разрабатываемого ресурса, сравнения его с аналогами, разработать клиентскую часть приложения с применением JavaFX, также в качестве дополнительных целей была выбрана практика с 3D технологиями на языке Java, создание базы данных на PostgreSQL для дальнейшего его использования в разработанном приложении.

Muradov N.N. Application "Shop of computer components" / Course work on the discipline "Patterns of software platforms of the Java language" profile "Development of software products and design of information systems" of the direction of professional training of bachelor's degree 09.03.04. "Software Engineering" (4th semester) / supervisor V.Т. Matchin, Art. Lecturer / Department of IiPPO, Institute of IT RTU MIREA.

The aim of the work is to analyze the subject area of ​​the resource being developed, compare it with analogues, develop the client part of the application using JavaFX, and practice with 3D technologies in the Java language was chosen as additional goals, creating a database on PostgreSQL for its further use in the developed application.

М. МИРЭА. Ин-т ИТ. Каф. ИиППО. 2021 г. Мурадов Н.

# **Аннотация**

Целью данной курсовой работы является создание приложения для создания web-форм.

В разделе «Содержание» приводятся наименования основных разделов и подразделов с указанием номера страницы.

В разделе «Введение» передается общая характеристика КР, а именно актуальность избранной темы, предопределение цели работы и задач для её осуществления, описание предмета исследования.

Основная часть состоит из трёх разделов: общие сведения, функциональное назначение и описание логической структуры.

В разделе «Общие сведения» излагаются обозначение и наименование, а также программное обеспечение, необходимое для разработки и функционирования интернет-ресурса.

В разделе «Функциональное назначение» демонстрируются совокупность свойств интернет-ресурса, характеризуемых конкретными особенностями набора функций, способных исполнять заданным или подразумеваемым нуждам.

В разделе «Описание логической структуры» определяется перечень основных вопросов и подразделов, такие как анализ предметной области, ряд выбранных технологий для создания интернет-ресурса и межстраничной навигации, а также исполнение оптимизации веб-страниц для различных устройств.

В разделе «Заключение» формулируются теоретические выводы и предложения, описывающие успех в достижении ранее поставленной цели.

Курсовой проект содержит 23 страницы отчёта, 10 иллюстраций, 9 использованных источников литературы.

**Оглавление**

[Аннотация 4](#_Toc69779980)

[Введение 7](#_Toc69779981)

[1. Общие сведения 8](#_Toc69779982)

[1.1. Обозначение и наименование интернет-ресурса 8](#_Toc69779983)

[1.2. Прикладное программное обеспечение, необходимое для разработки и функционирования интернет-ресурса 8](#_Toc69779984)

[1.3. Языки и технологии, с помощью которых реализован интернет-ресурс 8](#_Toc69779985)

[2. Функциональное назначение 10](#_Toc69779986)

[3. Описание логической структуры 11](#_Toc69779987)

[3.1. Анализ предметной области разрабатываемого веб-приложения 11](#_Toc69779988)

[3.1.1. Систематика выбора предметной направленности интернет-ресурса 11](#_Toc69779989)

[3.1.2. Информационная составляющая видов выбираемого ресурса 12](#_Toc69779990)

[3.1.3. Выводы к первому подразделу 12](#_Toc69779991)

[3.2. Описание структуры данных с использованием UML диаграмм 12](#_Toc69779992)

[3.2.1. Формирование UML диаграммы 12](#_Toc69779993)

[3.2.2. Описание классов проекта 13](#_Toc69779994)

[3.2.3. Выводы ко второму подразделу 13](#_Toc69779995)

[3.3. Разработка серверной части программного приложения 13](#_Toc69779996)

[3.3.1. Описание структуры проекта 13](#_Toc69779997)

[3.3.2. Система сборки Maven 13](#_Toc69779998)

[3.3.3. Контроллер DownloadConroller 14](#_Toc69779999)

[3.3.4. Выводы к третьему подразделу 15](#_Toc69780000)

[3.4. Реализация алгоритма веб-приложения 15](#_Toc69780001)

[3.4.1. Исполнение алгоритма с помощью JavaScript 16](#_Toc69780002)

[3.4.2. Связь JavaScript и Java Spring 17](#_Toc69780003)

[3.4.3. Выводы к четвёртому подразделу 17](#_Toc69780004)

[3.5. Тестирование работоспособности приложения 17](#_Toc69780005)

[3.5.1. Приложение Postman 17](#_Toc69780006)

[3.5.2. Тестирование с помощью Postman 17](#_Toc69780007)

[3.5.3. Выводы к пятому подразделу 18](#_Toc69780008)

[3.6. Графическая схема логики приложения 19](#_Toc69780009)

[3.6.1. Общая схема работы приложения 19](#_Toc69780010)

[3.6.2. Выводы к шестому подразделу 20](#_Toc69780011)

[Заключение 21](#_Toc69780012)

[Список информационных источников 22](#_Toc69780013)

[Приложения 23](#_Toc69780014)

[Приложение об исходном коде веб-приложения 23](#_Toc69780015)

# **Введение**

В современном обществе Интернет испытывает сильный взлёт и колоссальную востребованность. Бесчисленное количество веб-страниц хранят в себе массу практичной и интересной информации, а также безграничное количество вспомогательных программ. Обычно она всегда находится в публичном доступе, тем самым повышая популярность и рейтинг различных компаний.

Каждый веб-ресурс обычно ориентирован на определённую группу пользователей, которые имеют схожие интересы и потребности и ищут контент конкретного характера. Например, веб-сайт может представлять образовательную платформу, интернет-магазин, площадку для общения и прочее.

Целью данного курсового проекта является организация и разработка веб-приложения для создания web-форм. Для осуществления цели использованы знания и навыки, которые были получены в процессе посещения лекционных занятий, а также выполнения практический задач по настоящему курсу. Кроме того, будут применены технологии, изученные непосредственно из интернет-источников.

Для лучшего результата работу потребовалось разделить на несколько этапов: 1) проведение анализа предметной области, выбор требуемых технологий и установка программного обеспечения для их реализации; 2) разработка серверной части веб-приложения, используя JavaScript и Java Spring; 3) проведение тестирования приложения.

Веб-приложение, реализованное в течение представленного курсового проекта, имеет простую, но в то же время и комфортную клиентскую реализацию, а также соответствующую серверную часть.

1. **Общие сведения**
   1. **Обозначение и наименование интернет-ресурса**

Наименование разрабатываемого веб-приложения в настоящей курсовой работе: «Приложение для создания web-форм».

Данное название выбрано для того, чтобы абсолютно любой пользователь, заходящий на наш сайт, понимал, какая информация и функционал на нём представлены.

* 1. **Прикладное программное обеспечение, необходимое для разработки и функционирования интернет-ресурса**

Одним из первых этапов является установка программного обеспечения для проектирования и дальнейшей корректной работы веб-приложения.

Для создания интернет-ресурса было установлено программное обеспечение VS Code. Это редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации [1].

Для правильного функционирования веб-сайта необходимо установить любой веб-браузер, например Google Chrome, Mozilla, Opera или Microsoft Edge. В нашем случае установлен Microsoft Edge [2].

* 1. **Языки и технологии, с помощью которых реализован интернет-ресурс**

Клиентская часть веб-приложения написана на известном языке гипертекстовой разметки – HTML. Именно этот язык воспринимают все интернет-браузеры и переводят его на понятный интернет-пользователю внешний вид [3]. Вместе с языком HTML обычно используют CSS - формальный язык описания внешнего вида документа (веб-страницы), который позволяет задавать различные стили (цвета, шрифты, расположение блоков, тени, анимации и т.д.) для нашей разметки [4]. Для исполнения клиентской логики в нашем интернет-проекте используется JavaScript [5].

C целью осуществления серверной части приложения использовался фреймворк Java Spring. Это универсальный фреймворк с открытым исходным кодом для Java-платформы [6].

Фреймворк Spring MVC обеспечивает архитектуру паттерна Model — View — Controller (Модель — Отображение — Контроллер) при помощи готовых компонентов.

Model (Модель) объединяет данные приложения, в целом они будут состоять из Java-объектов.

View (Отображение, Вид) отвечает за отображение данных Модели, — как правило, генерируя HTML, которые мы видим в своём браузере.

Controller (Контроллер) обрабатывает запрос пользователя, создаёт соответствующую Модель и передаёт её для отображения в Вид [7].

1. **Функциональное назначение**

Текущее интернет-приложение, исполняющее функционал создания web-форм, предназначен для обеспечения интернет-пользователей более легким способом формирования форм для их веб-сайтов. Также приложение предоставляет собственную стилизацию для web-форм, что обеспечивает еще большую простоту в разработке других интернет-ресурсов.

Чтобы исключить все возможные ошибки в разработке серверной части приложения, были организованы тестирование соответствующих запросов и структура данных с помощью UML диаграмм. Также была построена графическая реализация логики приложения.

1. **Описание логической структуры**
   1. **Анализ предметной области разрабатываемого веб-приложения**

Реализация приложения для создания web-форм было выбрано мной прежде всего потому, что данный функционал помогает разработчикам быстро формировать различные формы для заполнения на своих сайтах.

* + 1. **Систематика выбора предметной направленности интернет-ресурса**

В настоящее время существует много дополнительных модулей и пакетов, которые помогают упростить разработку программ.

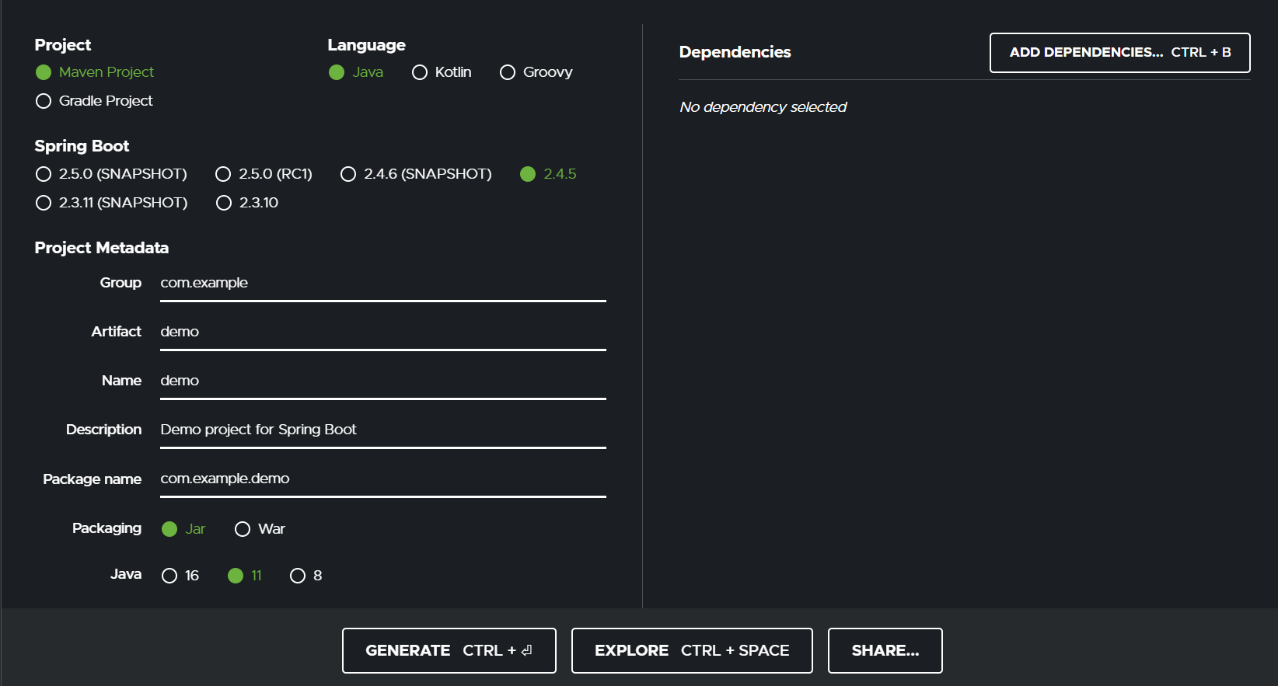
Мною был проанализирован список веб-сайтов для помощи разработчикам. Например, на сайте spring.io существует генерация Spring-приложения, а именно написание файла для системы сборки Maven или Gradle (см. Рисунок 3.1.1). Скачав данный файл и установив в свой проект, система сборки будет автоматически устанавливать необходимые зависимости, что поможет избежать дальнейших проблем в запуске приложения.

Рисунок 3.1.1 - Веб-страница сайта spring.io с генерацией файла системы сборки

Данный инструментарий будет очень полезен разработчикам, так как он позволяет сэкономить время и избежать непредвиденных ошибок в проекте.

Приложение для web-форм не нашлось в списке веб-сайтов, что еще раз дает понять о необходимости и актуальности создания такого веб-приложения.

* + 1. **Информационная составляющая видов выбираемого ресурса**

Заходя на любой похожий сайт, как spring.io, пользователю на глаз сразу представляется форма для заполнения данных, необходимых для генерации определенного файла. Данная форма может быть представлена в обычном виде с выбором вариантов и заполнением элементов форм, а также с помощью таблицы.

Также обязательным элементов интернет-приложения является кнопка «Сгенерировать». Она вызывает функцию для сборки всех данных, генерирует соответствующий файл и скачивает на компьютер пользователя.

* + 1. **Выводы к первому подразделу**

На основании рассмотренной выше информации можно сделать вывод, что интернет-приложение на данную тему будет очень востребован среди интернет-пользователей, особенно разработчиков.

* 1. **Описание структуры данных с использованием UML диаграмм**
     1. **Формирование UML диаграммы**

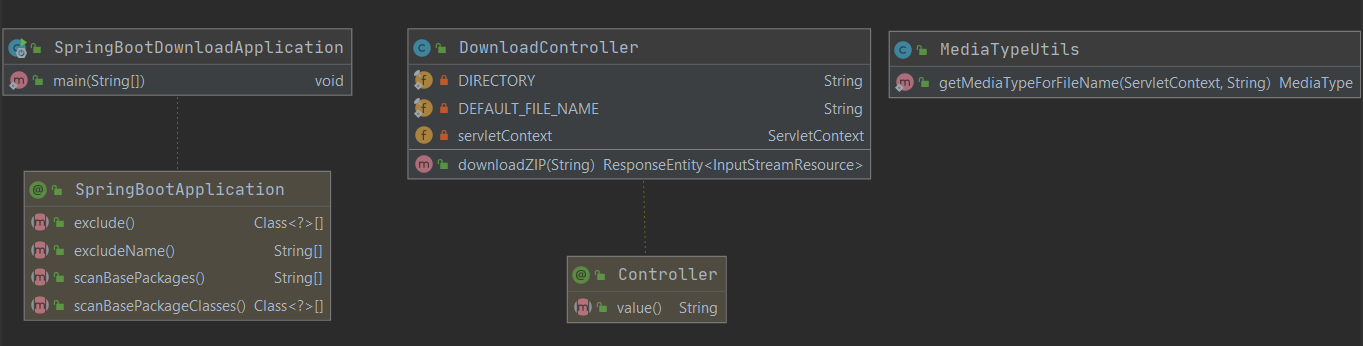
Для реализации структуры данных была спроектирована UML диаграмма приложения. Она автоматически сгенерировалась в среде разработки IntelliJ IDEA (см. Рисунок 3.2.1):

Рисунок 3.2.1 – UML диаграмма текущего проекта

* + 1. **Описание классов проекта**

Реализация курсового проекта по текущей теме содержит 3 класса:

1. DownloadController – класс-контроллер для скачивания необходимого файла, хранящегося на сервере;
2. MediaTypeUtils – класс для определения типа скачиваемого файла;
3. SpringBootDownloadApplication – класс для запуска Spring-приложения.

Класс DownloadController имеет 3 закрытых поля: директория хранения скачиваемого файла, название скачиваемого файла и сервлет для последующей отправки запроса на определение типа файла. Также данный класс имеет один метод, в котором прослушивается определенный запрос, и на основе полученного запроса выполняется основная логика программы, а именно скачивание сгенерированной web-формы.

Класс MediaTypeUtils содержит метод, получающий на вход сервлет (запрос) и скачиваемый файл, определяющий тип файла.

Класс SpringBootDownloadApplication является управляющим классом. Он вызывает предыдущие классы и запускает веб-приложение.

* + 1. **Выводы ко второму подразделу**

На основании рассмотренной выше информации можно сделать вывод, что структура данных текущего курсового проекта была сформирована. Были применены знания проектирования UML диаграмм.

* 1. **Разработка серверной части программного приложения**
     1. **Описание структуры проекта**

Spring-проект имеет определенную файловую систему. Он вмещает в себя файл конфигурации системы сборки Maven, директории «java» с разработанной серверной частью и «resources» со статическими элементами приложения, а именно клиентская часть.

* + 1. **Система сборки Maven**

Maven — механизм для автоматизации сборки проектов. Одна из главных особенностей фреймворка — декларативное описание проекта. Это значит, что разработчику не нужно уделять внимание каждому аспекту сборки — все необходимые параметры настроены по умолчанию. Изменения нужно вносить лишь в том объёме, в котором программист хочет отклониться от стандартных настроек.

Ещё одно достоинство проекта — гибкое управление зависимостями. Maven умеет подгружать в свой локальный репозиторий сторонние библиотеки, выбирать необходимую версию пакета, обрабатывать транзитивные зависимости [8].

* + 1. **Контроллер DownloadConroller**

Контроллер — это класс, предназначенный для непосредственной обработки запросов от клиента и возвращения результатов.

Чаще всего, в самом контроллере не описывается логика обработки данных. Создаются дополнительные, сервисные классы, которые выполняют все основные задачи по обработке данных. Задача методов в контроллере - вызов сервисных методов в нужном порядке и возвращение результатов клиенту.

Основная задача методов контроллера - определить требуемое действие, корректно принять данные, корректно возвратить результат.

В нашем случае был организован класс-контроллер для скачивания файла. Ниже представлен листинг кода (см. Рисунок 3.3.1):

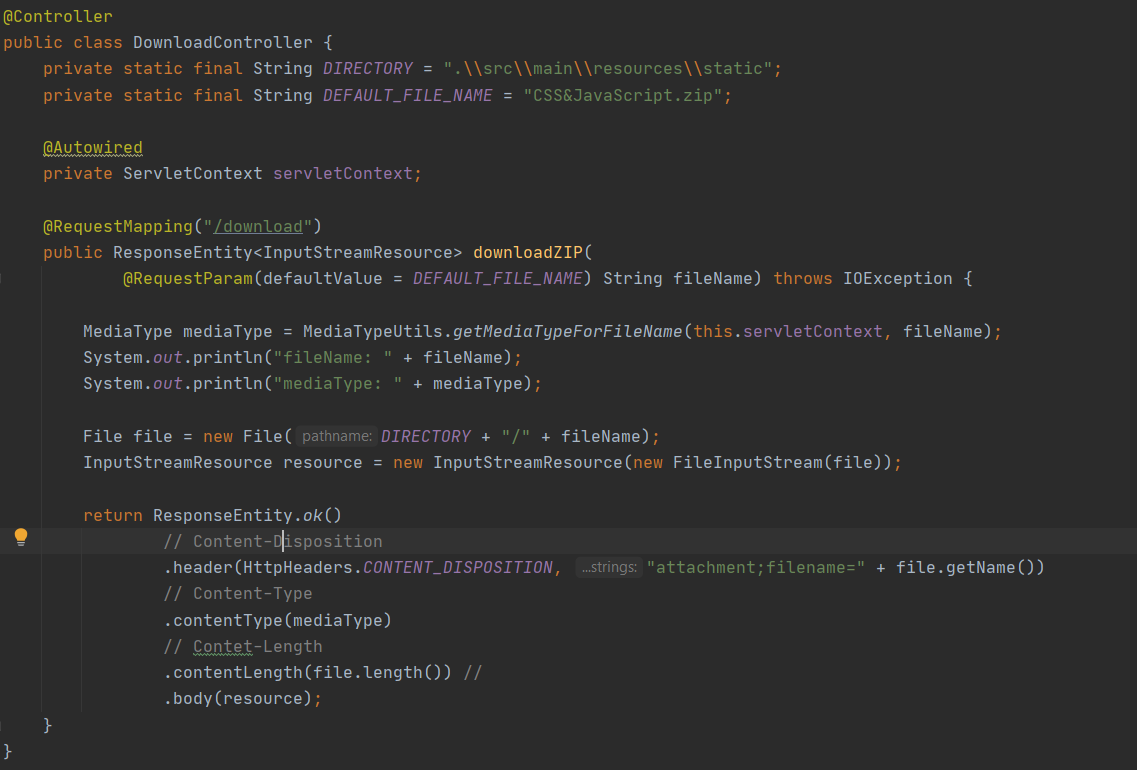
 Данный контроллер принимает запрос, относящийся к URL-адресу «/download» и вызывает соответствующий метод. Вызванный метод устанавливает и определяет данные Модели, а именно название файла, который будет скачиваться. В качестве возвращаемого значения выступает файл с определенными характеристиками: тип размещения отправленного содержимого для отображения в браузере, размер содержимого, тип содержимого и сам отправленный и подготовленный файл.

Рисунок 3.3.1 – Листинг кода контроллера DownloadController

* + 1. **Выводы к третьему подразделу**

На основании приведённой выше информации можно сделать вывод, что нам удалось выполнить еще один пункт реализации веб-приложения. Было выполнено условие организации серверной части приложения.

* 1. **Реализация алгоритма веб-приложения**

Организация алгоритма приложения была спроектирована с помощью JavaScript и соответствующей работы сервера Spring MVC.

* + 1. **Исполнение алгоритма с помощью JavaScript**

Большая часть курсовой работы написана на JavaScript. Скрипт генерирует почти все содержимое веб-страницы, в том числе таблицу, которая используется для формирования собственных web-форм, модальное окно для дополнительных настроек элементов формы, а также отдельную веб-страницу, которая в последствии скачивается на компьютер пользователя.

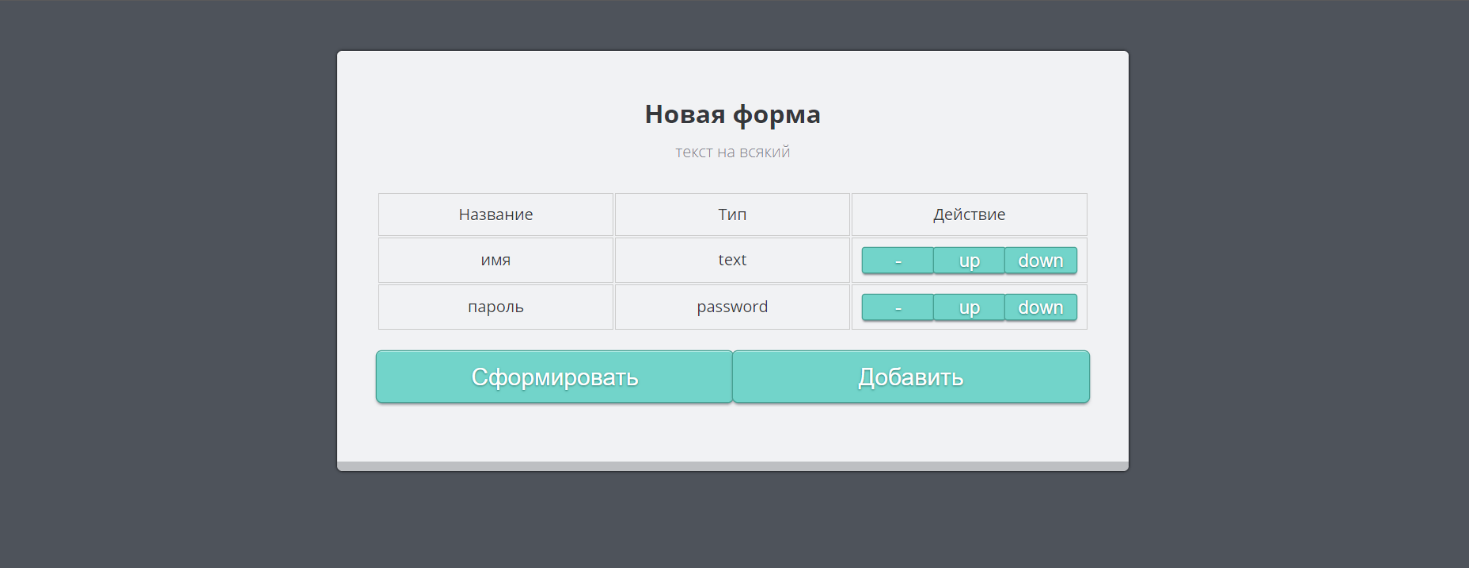
Ниже представлена веб-страница и ее функционал (см. Рисунок 3.4.1):

Рисунок 3.4.1 – Реализация главной веб-страницы приложения

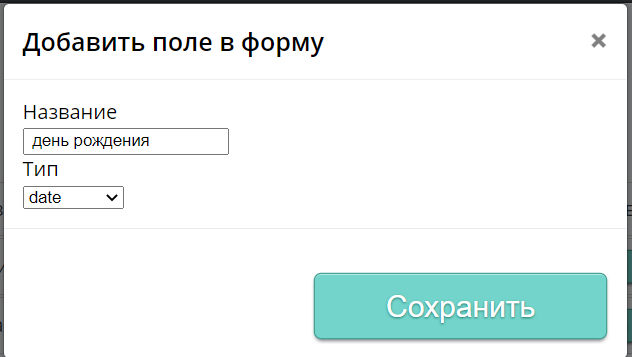
 Возможно перемещение элементов формы так, как захочет пользователь, а также и их удаление. Перед добавлением элемента в таблицу со всеми данными появляется модальное окно для дополнительной настройки (см. Рисунок 3.4.2):

Рисунок 3.4.2 – Реализация модального окна

После нажатия на кнопку «Сформировать», скрипт генерирует страницу и скачивает ее, а сервер в последствии скачивает подготовленные для нее стили и скрипты для проверки на валидность.

* + 1. **Связь JavaScript и Java Spring**

При нажатии на кнопку «Сформировать», она не только генерирует файл, но и перенаправляет на соответствующий запрос, который в URL-адресе включает «/download». Сервер получает данный запрос, вызывает должный контроллер, обрабатывает данные и отправляет содержимое браузеру.

На Рисунке 3.4.3 представлен листинг кода формирования веб-страницы с формой и ее скачивания, а также отправки запроса на сервер:

Рисунок 3.4.3 – Листинг кода генерации файла с формой и отправки запроса

* + 1. **Выводы к четвёртому подразделу**

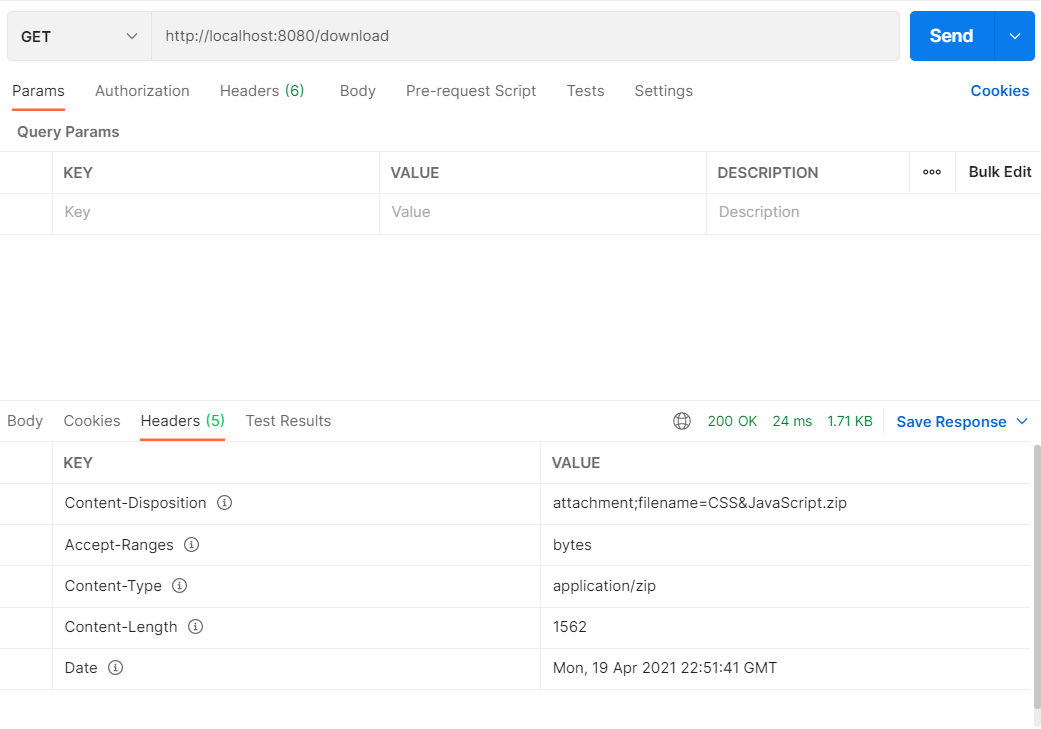
На основании представленной выше информации можно сделать вывод, что пункт с организацией алгоритмов на JavaScript и Java Spring был выполнен.

* 1. **Тестирование работоспособности приложения** 
     1. **Приложение Postman**

Postman предназначен для проверки запросов с клиента и получения ответа от сервера.

Основное предназначение приложения — создание коллекций с запросами к вашему API. Любой разработчик или тестировщик, открыв коллекцию, сможет с лёгкостью разобраться в работе вашего сервиса [9].

* + 1. **Тестирование с помощью Postman**

Было проведено тестирование разработанного веб-приложения. В нашем случае был отправлен запрос, включающий в себя «/download». Результат тестирования показан на Рисунке 3.5.1*:*

Тестирование показало, что запрос работает корректно. Также внизу написаны свойства, о которых говорилось выше и которые определялись перед скачиванием файла.

Рисунок 3.5.1 - Тестирование запроса с помощью Postman

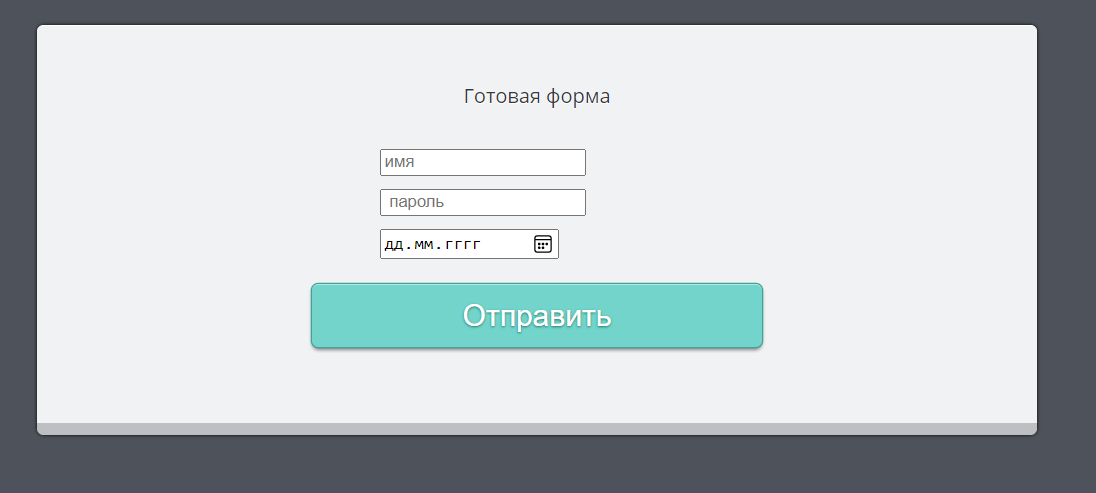
На Рисунке 3.5.2 показана сформированная web-форма:

Рисунок 3.5.2 – Веб-страница с сгенерированной web-формой

По умолчанию стоит стилизация страницы, как и основного приложения, но пользователь-разработчик может в любой момент изменить ее.

* + 1. **Выводы к пятому подразделу**

На основе приведённой выше информации можно сделать вывод, что веб-приложение успешно протестировано и работает без ошибок.

* 1. **Графическая схема логики приложения**
     1. **Общая схема работы приложения**

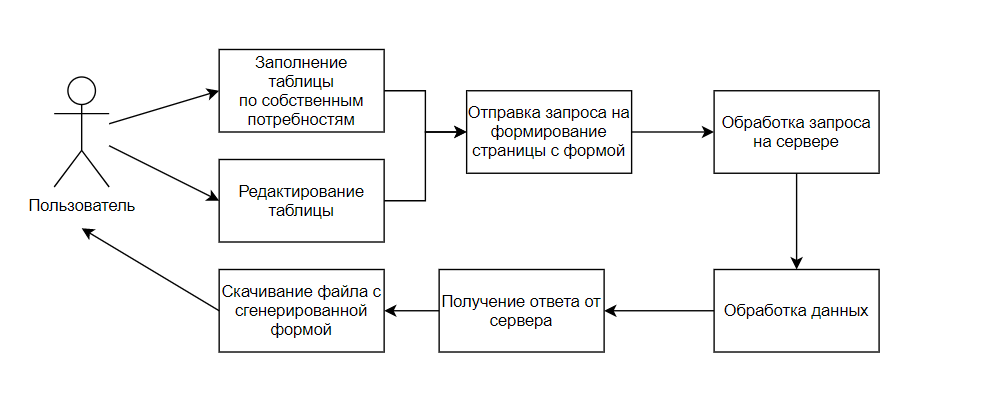
Ниже представлена общая схема работы приложения курсового проекта (см. Рисунок 3.6.1):

Рисунок 3.6.1 – Общая схема логики приложения

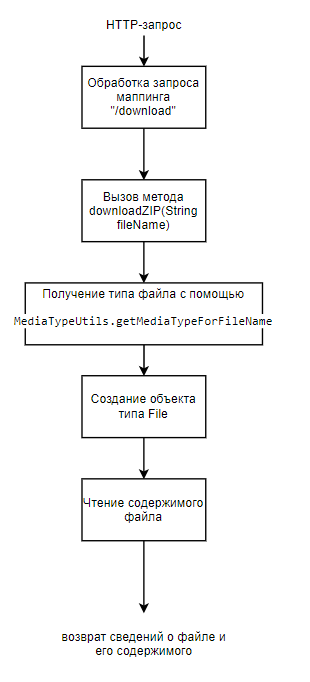
Далее представлена схема работы контроллера (см. Рисунок 3.6.2):

Рисунок 3.6.2 – Схема работы контроллера DownloadController

* + 1. **Выводы к шестому подразделу**

На основании приведённой выше информации можно сделать вывод, что пункт с построением графических схем логики приложения был выполнен.

# **Заключение**

Текущий электронный ресурс позволяет разработчикам сэкономить время и быстро сгенерировать страницу с web-формой.

В ходе выполнения курсового проекта были достигнуты следующие задачи:

1. Проведен анализ предметной области;
2. Была описана структура данных с помощью UML диаграмм;
3. Реализована серверная часть приложения на Java Spring MVC;
4. Были использованы большинство навыков, полученные в результате прохождения лекционных и практических занятий;
5. Были получены навыки в осуществлении связи между Java Spring и JavaScript;
6. Проведено тестирование запросов с помощью программы Postman;
7. Графически описана логика приложения.

Все пункты, указанные в техническом задании, работают корректно и в полной мере. Исходя из всего вышеперечисленного, можно сделать вывод, что данный курсовой проект выполнен успешно.

# **Список информационных источников**

1. Интегрированная среда разработки VS Code [Электронный ресурс]: Режим доступа: [Visual Studio Code - Code Editing. Redefined](https://code.visualstudio.com/), свободный (Дата последнего обращения: 10.04.2021)
2. Возможности браузера Microsoft Edge [Электронный ресурс]: Режим доступа: [Возможности браузера Microsoft Edge | Майкрософт](https://www.microsoft.com/ru-ru/edge/features), свободный (Дата последнего обращения: 10.04.2021)
3. Справочник по HTML [Электронный ресурс]: Режим доступа: [Справочник по HTML | htmlbook.ru](http://htmlbook.ru/html), свободный (Дата последнего обращения: 02.04.2021)
4. Справочник по HTML, CSS и JavaScript [Электронный ресурс]: Режим доступа: [CSS и CSS3. Свойства для форматирования html-элементов (html5book.ru)](https://html5book.ru/css-css3/), свободный (Дата последнего обращения: 02.04.2021)
5. Учебник по JavaScript [Электронный ресурс]: Режим доступа: <https://learn.javascript.ru/>, свободный (Дата последнего обращения: 03.04.2021)
6. Начало работы с Java Spring [Электронный ресурс]: Режим доступа: [Getting Started | Serving Web Content with Spring MVC](https://spring.io/guides/gs/serving-web-content/), свободный (Дата последнего обращения: 06.04.2021)
7. Основные принципы Spring MVC [Электронный ресурс]: Режим доступа: [Spring MVC — основные принципы / Хабр (habr.com)](https://habr.com/ru/post/336816/), свободный (Дата последнего обращения: 08.04.2021)
8. Краткое знакомство с Maven [Электронный ресурс]: Режим доступа: [Краткое знакомство с Maven (tproger.ru)](https://tproger.ru/articles/maven-short-intro/), свободный (Дата последнего обращения: 06.04.2021)
9. Введение в Postman [Электронный ресурс]: Режим доступа: [Введение в Postman / Блог компании Kolesa Group / Хабр (habr.com)](https://habr.com/ru/company/kolesa/blog/351250/), свободный (Дата последнего обращения: 10.04.2021)

# **Приложения**

## **Приложение об исходном коде веб-приложения**

Исходный код данного электронного ресурса, включающий в себя все файлы, является слишком объемным, чтобы его было уместно прикрепить в качестве приложения, поэтому он прилагается в папке, находящейся в одной директории со всеми файлами и документами данной курсовой работы.