

## СЕКЦИЯ 2. СИСТЕМЫ АВТОМАТИЗАЦИИ И ИНФОРМАТИЗАЦИИ В ОБРАЗОВАНИИ, НАУКЕ И ПРОИЗВОДСТВЕ

УДК 004.91

К. А. Асабин, И. А. Гурин, А. А. Куят

ФГАОУ ВО «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина», г. Екатеринбург, Россия

### РАЗРАБОТКА ВЕБ-СЕРВИСА ДЛЯ ЗАПОЛНЕНИЯ ШАБЛОНОВ MICROSOFT WORD НА ПЛАТФОРМЕ .NET CORE

**Аннотация.** *Представлено описание веб-сервиса, который позволяет создавать персонализированные документы на основе шаблонов документов Microsoft Word. Каждый шаблон может содержать несколько текстовых полей разных типов (текст, число, дата и др.). Веб-сервис разработан на языке программирования C# с использованием свободно-распространяемого кроссплатформенного фреймворка для создания веб-приложений с открытым исходным кодом ASP.NET Core MVC. Для создания текстовых документов на стороне сервера использована библиотека TemplateEngine.Docx. Преимуществом решения является то, что данные документов хранятся в базе данных, а при загрузке документа переносятся во временно создаваемый документ.*

**Ключевые слова:** *веб-сервис, язык программирования C#, TemplateEngine, Microsoft Word, автоматизация, текстовый документ, шаблонизация.*

**Abstract.** *A description of a web service that allows you to create personalized documents based on Microsoft Word document templates is presented. Each template can contain several text fields of different types (text, number, date, etc.). The web service is developed in the C # programming language using the free, cross-platform framework for building open source ASP.NET Core MVC web applications. The TemplateEngine.Docx library was used to create text documents on the server side. The advantage of the solution is that the document data is stored in the database, and when the document is loaded, it is transferred to the temporarily created document.*

**Key words:** *web service, C# programming language, TemplateEngine, Microsoft Word, automation, text document, templating.*

Развитие современных информационных технологий характеризуется переходом к облачным вычислениям и автоматизированной обработке данных. Если рассмотреть данный процесс с точки зрения работы с электронными документами, то он предполагает переход от работы с офисными приложениями на локальной машине пользователя к работе с облачными веб-сервисами, в которых происходит заполнение, обработка и создание готовых текстовых документов, электронных таблиц и др. Документы, которые раньше создавались, редактировались и хранились локально, сейчас сохраняются на удаленных сетевых дисках. В работе [1] рассмотрен круг решаемых задач и необходимость автоматизированной обработки данных электронных документов.

В данной работе выполнен обзор технологий, используемых при разработке веб-сервиса, предназначенного для заполнения шаблонов текстовых документов.

Редактор Microsoft Word является одним из наиболее популярных текстовых процессоров на сегодняшний день, поэтому для автоматизированной

обработки использовались создаваемые в этом программном продукте документы.

Веб-сервис разработан на языке программирования C# с использованием свободно-распространяемого кроссплатформенного фреймворка для создания веб-приложений с открытым исходным кодом ASP.NET Core MVC. Фреймворк ASP.NET Core MVC работает поверх платформы ASP.NET Core и предназначен для того, чтобы упростить создание приложения [2]. Платформа ASP.NET Core разрабатывается компанией Microsoft совместно с сообществом и имеет большую производительность по сравнению с ASP.NET [3].

Для обработки текстовых документов на стороне сервера используется библиотека TemplateEngine.Docx. Это открытая библиотека для работы с документами Microsoft Word с расширением DOCX. Она позволяет создавать новые документы на основе подготовленных шаблонов Microsoft Word и включает следующие возможности [4]:

- заполнение простых полей;
- заполнение таблиц;
- заполнение списков и вложенных списков;
- заполнение таблиц со списками;
- заполнение списков с таблицами;
- работа с изображениями.

Шаблон документа создается в Microsoft Word с соблюдением определённого стиля, при этом сразу видно, на что будет похож результирующий документ. Результирующий документ после скачивания содержит все необходимые данные, не подтягивая их с внешних источников.

Редактирование шаблона не требует наличия специальных навыков работы с документами. Например, шаблон можно доверить секретарю, чтобы он мог сменить логотип, реквизиты компании или изменить форматирование текста.

В качестве шаблона используются документы с расставленными элементами управления содержимым или «Content Controls», которые связаны с данными через свойство «Тег». Content Controls добавляются достаточно легко, при этом их достаточно сложно испортить при дальнейшей эксплуатации, а при отключенном режиме конструктора спец-обозначений «контролов» и вовсе не видно. На рисунке 1 показаны шаги создания таких элементов.

Для передачи информации из приложения в документ используются объекты *Content* и *FieldContent* библиотеки TemplateEngine.docx.

```
var content = new Content();
var alias = "field";
var value = "value";
content.Fields.Add(new FieldContent(alias, value));
```

В примере *alias* – это название созданных полей, элементов управления содержимым, а *value* – это данные, которые были введены в приложении на форме заполнения.

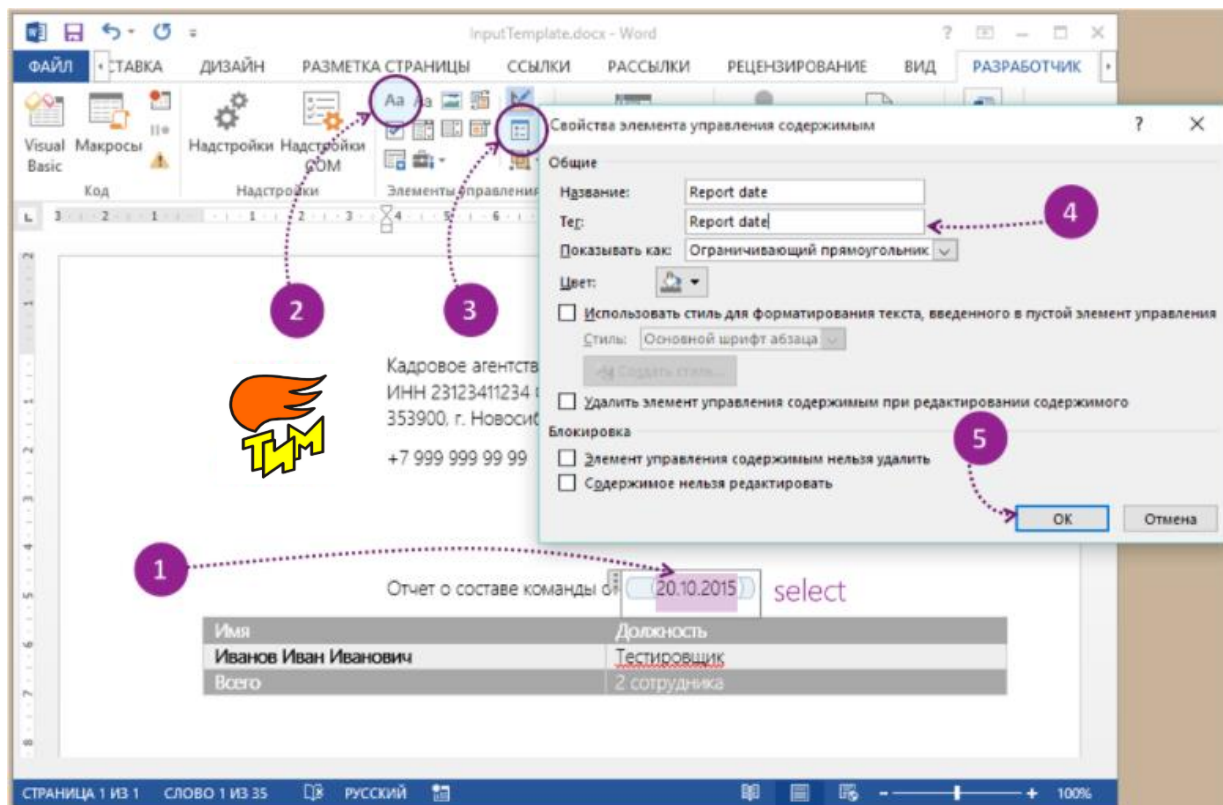


Рис. 1. Порядок создания элементов управления содержимым в Microsoft Word

Чтобы перенести данные, внесенные в документ вышеописанным кодом, в финальный документ, который можно скачать, необходимо создать новый объект с типом `TemplateProcessor`.

```
using (var outputDocument = new
TemplateProcessor("tempFileName.docx").SetRemoveContentControls(true))
{
    outputDocument.FillContent(content);
    outputDocument.SaveChanges();
}
```

Метод `SetRemoveContentControls` предназначен для удаления элементов управления содержимым, так как они чаще всего уже не нужны в результирующем документе. В метод `FillContent` передается в качестве параметра объект типа `Content`, созданный на предыдущем шаге.

Хранение данных в веб-сервисе реализовано с использованием СУБД `MariaDB`, распространяемая под лицензией `GNU GPL`. Схема базы данных приложения представлена на рисунке 2.

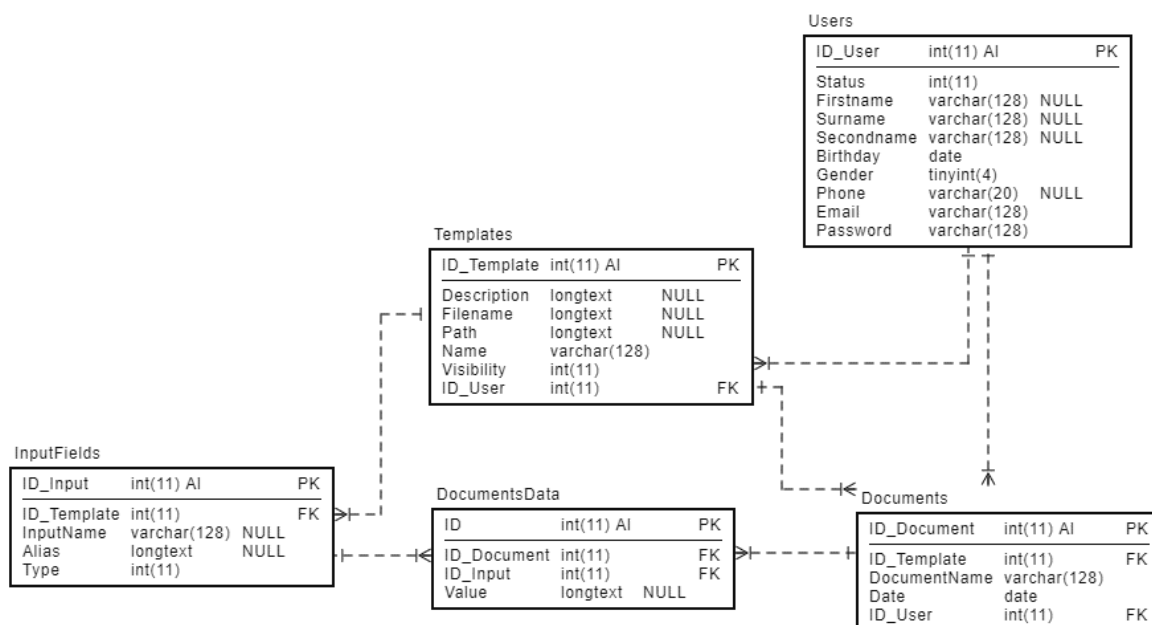


Рис. 2. Схема базы данных веб-сервиса

На рисунке 3 показаны шаблон документа и документ с данными, сформированный с помощью разработанного веб-приложения.

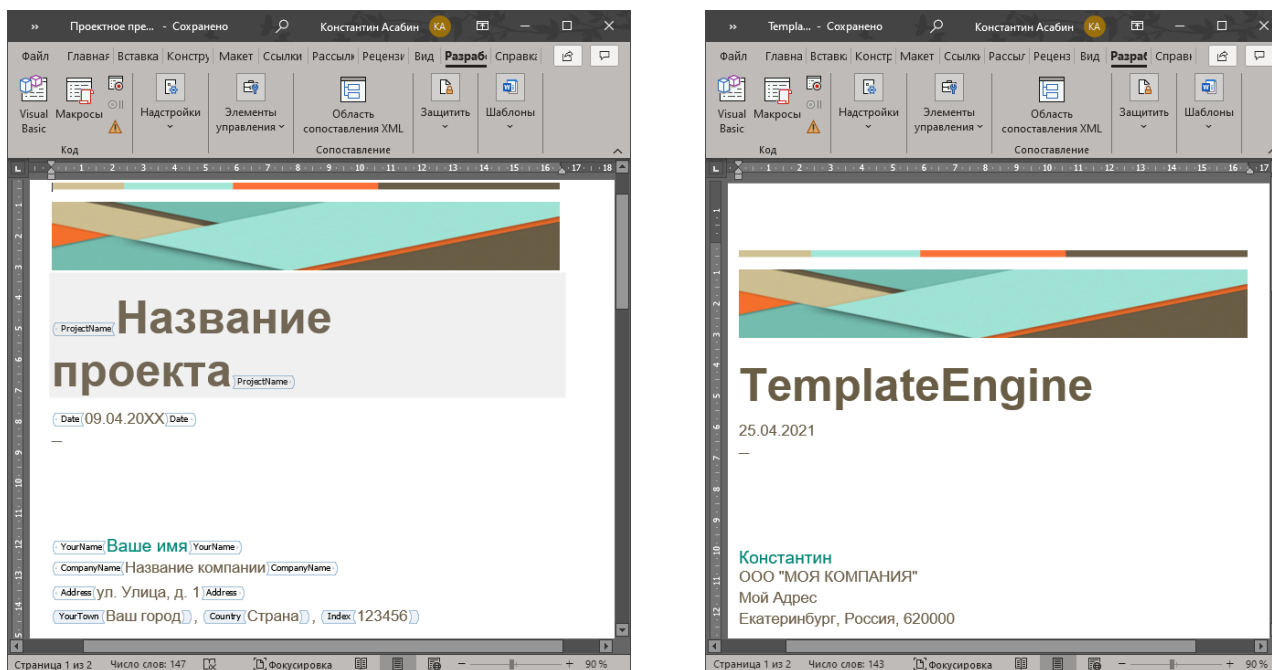


Рис. 3. Шаблон (слева) и документ с данными (справа)

Таким образом, с использованием свободного программного обеспечения разработан веб-сервис, предназначенный для заполнения шаблонов текстовых документов. Разработанный веб-сервис предоставляет возможность регистрации пользователя и загрузки шаблонов в личный кабинет. Шаблон имеет два варианта доступа: открытый и персональный. Открытые шаблоны доступны для использования всем пользователям сервиса, а персональные только владельцу.

Каждый шаблон может содержать несколько текстовых полей разных типов (текст, число, дата и др.). На основе шаблонов с помощью заполняемых на веб-интерфейсе форм создаются документы. Данные документов хранятся в базе данных, а при скачивании документа переносятся во временно создаваемый документ с помощью библиотеки TemplateEngine.docx.

### Список использованных источников

1. Методы работы с документами Microsoft Word при разработке веб-сервисов / И.А. Гурин, Н.А. Спиринов, В.В. Лавров, Н.И. Декун // Теплотехника и информатика в образовании, науке и производстве: сборник докладов VI Всероссийской научно-практической конференции студентов, аспирантов и молодых учёных (ТИМ'2017) с международным участием (11–12 мая 2017 г.). – Екатеринбург: УрФУ, 2017. С. 199-203.

2. Начало работы с MVC ASP.NET Core [Электронный ресурс] // Microsoft: [web-сайт]. – Режим доступа: <https://docs.microsoft.com/ru-ru/aspnet/core/tutorials/first-mvc-app> (дата обращения 25.04.2021).

3. Документация по ASP.NET [Электронный ресурс] // Microsoft: [web-сайт]. – Режим доступа: <https://docs.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-3.1> (дата обращения 25.04.2021).

4. Открытая библиотека по работе с шаблонами документов «TemplateEngine.Docx — OpenSource .NET» [Репозиторий GitHub] // Режим доступа: <https://github.com/UNIT6-open/TemplateEngine.Docx> (дата обращения 25.04.2021).

УДК 504.064.36

**А. О. Басистюк**

ФГБОУ ВО «Череповецкий государственный университет», г. Череповец,  
Россия

### ОПТИКО-ЭЛЕКТРОННАЯ СИСТЕМА КОНТРОЛЯ КАЧЕСТВА СМОТКИ ГОРЯЧЕКАТАНЫХ РУЛОНОВ

*Аннотация. Решение задачи показывает, что использование «ручного» способа осмотра горячекатаных рулонов отделом технического контроля (ОТК) на дефекты смотки, является более затратным и менее эффективным, чем применение оптико-электронной системы контроля качества смотки, которая использует камеры для получения кадров с рулонами, а также современные нейросетевые технологии машинного обучения, позволяющие решить проблемы обнаружения и классификации основных видов дефектов в режиме реального времени. В процессе производственного тестирования выявлено, что при осмотре рулонов ОТК, было зафиксировано на 23 % меньше дефектов, чем разработанной системой, однако суммарная точность решения составляет 96.7 %. Ошибки или же «ложные срабатывания системы», связаны с такими факторами как, большое параметрическое разнообразие выпускаемой продукции (высота, ширина и толщина полотна*