

ДОДАТОК А
СПЕЦИФІКАЦІЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Software Requirements Specification

for

BarberDoc Project

Version 1.0 approved

Prepared by

Serhii Iordanov

Oleksii Budianskyi

NURE

04/07/2020

Revision History

Date	Description	Author	Comments
04/03/2020	v.0.1	Serhii Iordanov	Initial draft
04/05/2020	v.0.2	Oleksii Budianskyi	Review + enhancements
04/07/2020	v.1.0	Serhii Iordanov Oleksii Budianskyi	Final Revision

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Ilona Revenchuk	Product Owner	04/07/2020

1 INTRODUCTION

Nowadays, there are many institutions that work with a constant stream of documents that must meet certain well-defined formatting and layout requirements. Most of these organizations are faced with the problem of document clearance, which is a routine job and takes up a large part of the working hours of specialists. Similar problems can be encountered in almost every field, as document management is an integral part of the modern profession. Solution is based on the problem of automatic documents formatting verification.

Web-service devoted to automate the documents formatting verification process. System receives the input document and formatting rules and returns the list of validation errors to fix.

1.1 Purpose

The SRS document devoted to specify functional and non-functional requirements to cover the business needs in scope of the system described above.

1.2 Scope

The BarberDoc Project scope includes the next parts:

- Implement the library for document validation.
- Implement expandable set of validators.
- Implement an API to work with the library from the Web.
- Implement the UI application for end users.
- Implement a console application to test and develop the library.

1.3 Definitions, Acronyms, and Abbreviations

API – Application Programming Interface

XML – Extensible Markup Language

PDF – Portable Document Format

DFD – Data Flow Diagram

2 GENERAL DESCRIPTION

This section of the SRS describes the general factors that affect 'the product and its requirements.

2.1 Product Perspective

There huge clients' base. The system could be used by any person who works with documentation. There is an ability to enter a contact with universities or other organizations.

2.2 Product Functions

The system will consist of three main parts: frontend, backend and validation library. In production, the service should provide the following possibilities:

1) Backend

- a. Handles authentication;
- b. Performs preliminary request validation;

- c. Deserializes the document from the request;
- d. Calls validation library;
- e. Serializes validation result;
- f. Performs localization.

2) Frontend

- a. Displays validation errors;
- b. Provides inputs for files uploading;
- c. Provides an access to FAQ;
- d. Provides an access to formatting settings.

3) Validation Library

- a. Executes formatting validators concurrently;
- b. Combines validators results into a single list of errors.

2.3 User Characteristics

There are two main stakeholders:

- Students – need to have an ability to preliminary validate the formatting of their works which are in progress or ready to be finally verified.
- Formatting validation experts - need to automatically validate works.

2.4 General Constraints

Development limited by the following restrictions:

- development will be conducted only for NURE university;
- the trial license of Spire.Doc and Spire.Pdf is used.

2.5 Assumptions and Dependencies

The system supports only docx and pdf format for input documents and currently configured to work with “ДСТУ 3008-2015” requirements.

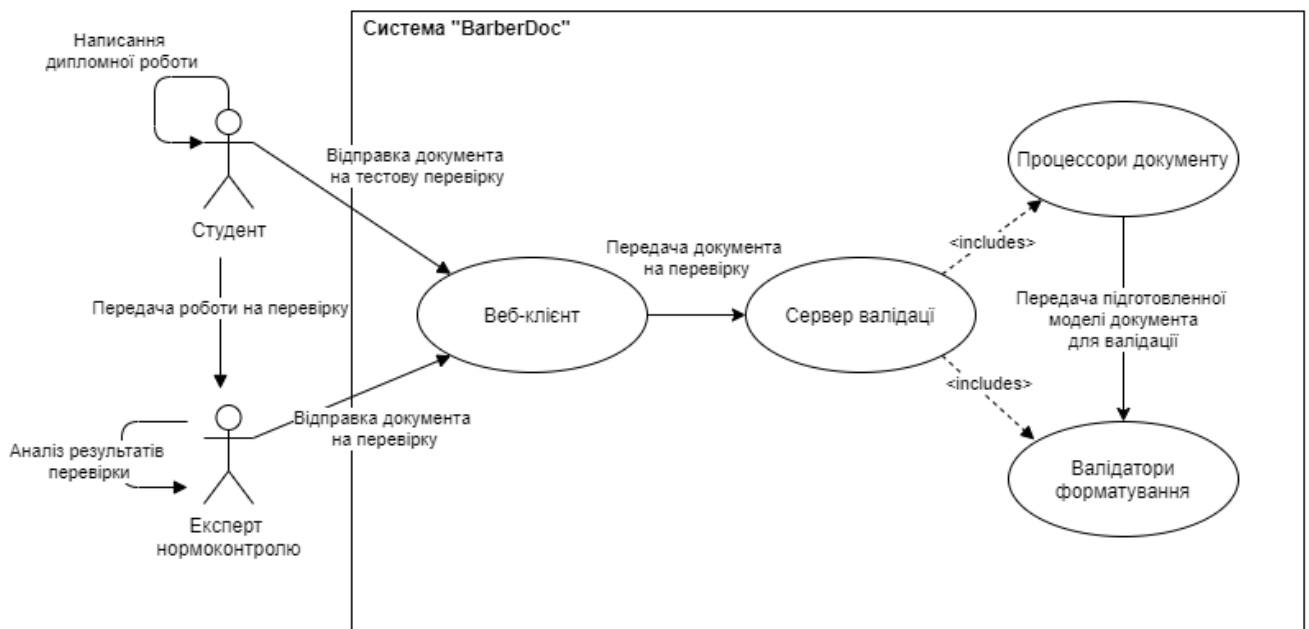
3 SPECIFIC REQUIREMENTS

3.1 User roles

BarberDoc solution addresses the requirements for the following group of users:

- Students
- Formatting validation experts
- System administrator

3.2 Use case scenarios



Picture 1 – Use case diagram

Main actors:

- Student;

- Formatting validation experts.

Pre-conditions

- Student has the prepared work in docx format;
- System administrator had configured the system to correctly validate the formatting;

Main flow

- Students sends the work to the system for preliminary validation;
- System provides validation errors;
- Student fixes the formatting of the work;
- Student sends the work to the formatting validation expert;
- Formatting validation expert validates the work via the system.

3.3 Functional requirements

Document - a file of the .docx format that meets the following requirements.

Sections and subsections are marked with the “Title” tool of MS Office.

Functional requirements:

- a) The system should have several ways of interacting with the user:
 - 1) Through a console application:
 - A Document is submitted to the input;
 - The output is a list of errors.
 - 2) Via the web client:
 - A Document is submitted to the input;
 - The output is a list of errors.
- a) The system should analyze the Document for a violation of the specified formatting rules:
 - 1) The document should not be blank
 - 2) The first page of the document should be the title page

- No page numbering;
- There is a hat "Міністерство ...";
- At the end of the page, the current year is centered.

3) There should be a page with the content

- The page contains a Table of Content element;
- Page numbering matches the real;
- Items are arranged in order and match the document headers;
- All items are written in the format of sentences (The first letter is large, then small).

4) Before the page with the content there is a page with the title "РЕФЕРАТ / ABSTRACT"

- The first paragraph of the text correctly indicates the number of figures, tables, diagrams, applications, sources

5) The indentation throughout the note is five characters or 1.25 cm. Fields:

- left - 2.5 cm.
- right - 1 cm
- upper and lower - 2 cm.

6) Formatting throughout the document

- Interval of the main text - one and a half;
- Text font - Times New Roman, black, 14;
- Throughout the Document, starting from the "VSTUP" section, there should be numbering (in the upper right corner);
- The page should not end with a title;
- Using the correct dash;
- The page cannot begin with a picture.

7) Sections:

- Each section starts on a new page;
- The last page of the section must have at least 10 lines;

- The section cannot end with a picture, table, diagram;
- Headings of the main sections - in capital letters;
- Headings and subheadings do not end with a period;
- Section title - indent or in the center;
- Section headings - with indentation;
- Two indents after the title or subtitle before the text (normal indentation between the title and subtitle);
- Numbering of headings and subheadings without a dot (not “1.1.”, But “1.1”);
- Partition numbering - in order of placement.

8) Applications:

- The appendices are at the end of the Document and are numbered A, B, C ...;
- Application formatting is ignored.

9) List of sources

- Numbered;
- References to sources are present in the text (at least one reference to the source), in the format “[1]”;
- Arranged in the order of mention in the text.

10) Lists

- Only letters, numbers, or dashes are used.

11) Pictures

- Numbering - end-to-end or section by section;
- In applications, numbering - A.1, A - application number;
- The signature of the figure is "Figure 1.1 - The name of the figure";
- Indentation before the drawing, after the drawing and before the text after the signature;
- The text should contain a link to the picture;

- The figure is in the center.
- 12) Tables:
- Table name - above the table;
 - The name format is “Table 1.1 - Table Name”;
 - The table column names are located in the center of the table cell;
 - Indentation before and after the table.
- 13) The result of the analysis should be a list of errors and warnings about found formatting violations.

4 NON-FUNCTIONAL REQUIREMENTS

4.1 Performance

The system must be interactive and the delays involved must be less than 1 minute. So, in every action-response of the system, there are no immediate delays. In case of opening pages, of popping error messages and saving the settings or sessions there is delay much below 2 seconds.

4.2 Reliability

As the system provide the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

4.3 Availability

If the internet service gets disrupted while sending information to the server, the information can be send again for verification.

4.4 Security

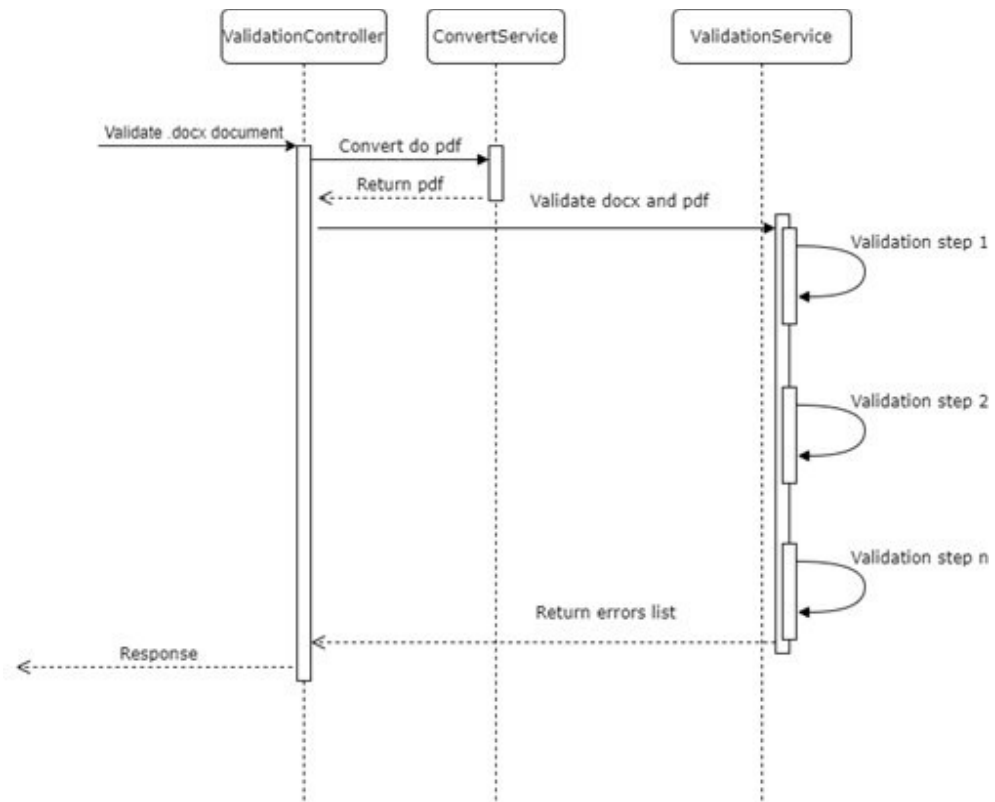
The main security concern is for users account hence proper login mechanism should be used to avoid hacking. The tablet id registration is way to spam check for increasing the security. Hence, security is provided from unwanted use of recognition software.

4.5 Usability

As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

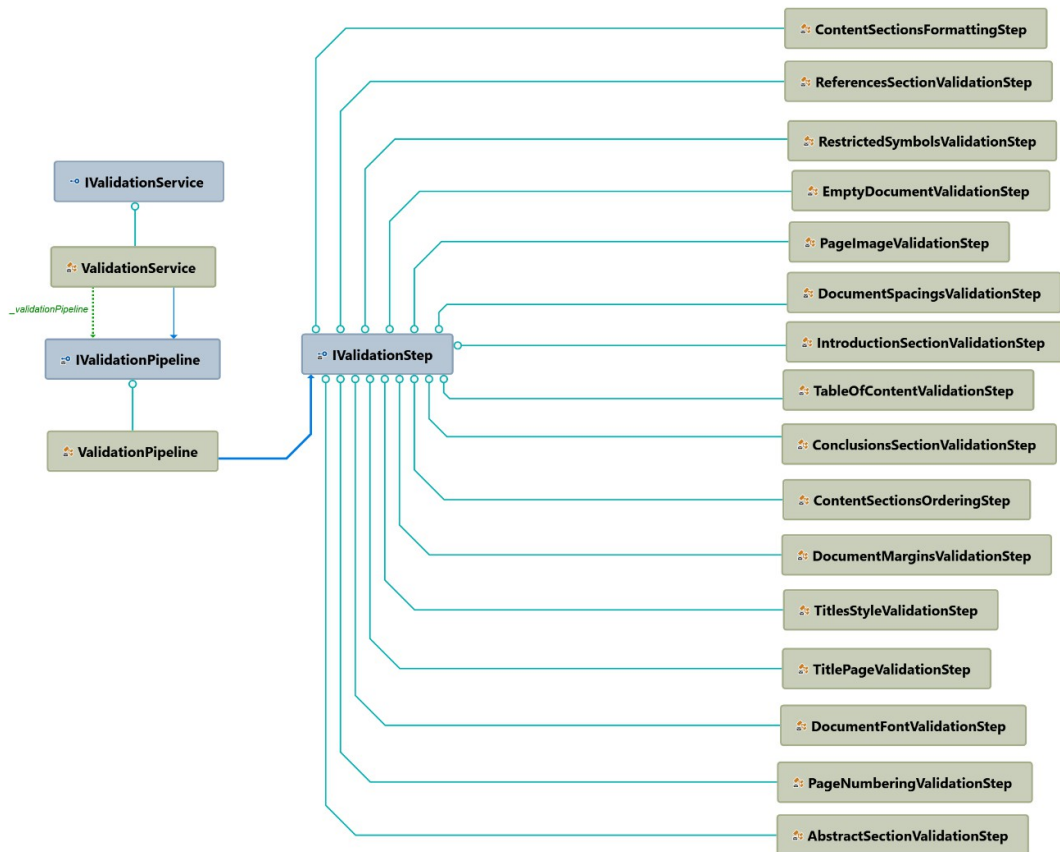
5 ANALYSIS MODELS

5.1 Sequence Diagrams



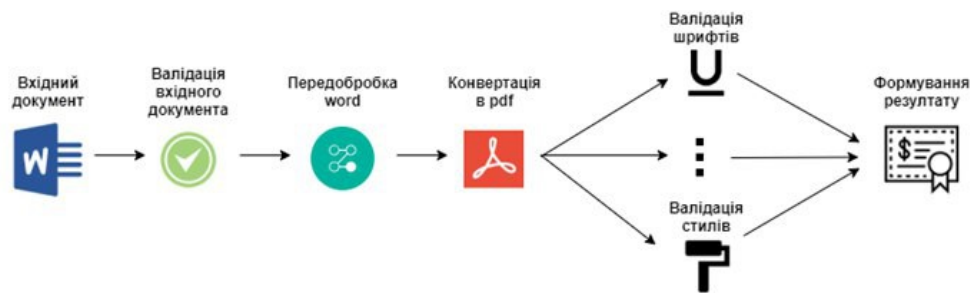
Picture 2 - Sequence Diagrams

5.2 Class diagram



Picture 4 – Class Diagram

5.3 Concurrency diagram



Picture 5 – Concurrency Diagram

6 CHANGE MANAGEMENT PROCESS

Change management process consists of the following steps:

1) Request for change is made.

Changes and issues are captured through Project Team meetings, a working group of external stakeholders including other government agencies, meetings of the Project Board, outcomes of approval stages and through monitoring by the Project Team and Project Sponsor.

2) Register and assess the change.

Change requests are captured on a central change log that enables monitoring of change levels by the Change Manager.

3) Review and submit RFC to Change Board.

If the change is valid the Project Manager assesses the impact of the change with the project team and submits the change via a Request for Change Form to the Change Manager. It is then submitted to the Change Board who assess the change and approve or reject it.

4) Change Board accept or reject the change.

The Change Board includes the client and senior managers who have a strategic overview of the contract. They assess whether the change is within program tolerances and are able to approve most changes.

5) Update plans & implement the change.

Once the change is approved the change log is updated and stakeholders are informed. The Project Manager then works with the project team to plan the implementation of the change; amending the project and stage plans, sourcing any additional equipment or personnel, updating the configuration items and completing the work package(s).

ДОДАТОК Б
ВІДГУК КЕРІВНИКА АТЕСТАЦІЙНОЇ РОБОТИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет комп'ютерних наук
ВІДГУК

на атестаційну роботу магістра
Іорданова Сергія Олексійовича, ІПЗМ-18-3
спеціальність 121 – *Інженерія програмного забезпечення*
освітньо-наукова програма «Інженерія програмного забезпечення»

Тема атестаційної роботи - Дослідження методів обробки даних у файлах docx для тестування контенту на виявлення помилок щодо вимог до оформлення тексту

В процесі дослідження магістрантом проведений аналіз предметної галузі, здійснено дослідження основних методів, моделей та алгоритмів аналізу відповідності електронного документу до вимог оформлення. В результаті аналізу виявлено декілька найбільш ефективних методів та досліджено можливість їх поєднання в процесі програмної реалізації.

Програмна реалізація функціональної частини системи являє собою набір бібліотек, написаних з використанням фреймворку .NET Core. Бібліотеки розроблено для максимального спрощення використання їх у різних типах систем. Використовуються консольний додаток для тестування, а також клієнт-серверна архітектура для інтеграції системи. Систему розгорнуто у середовищі Azure.

Результати роботи досить повно відображені в пояснювальній записці. Матеріал в атестаційній роботі викладено логічно і структурно. Зважаючи на вищевказане, атестаційну роботу виконано на хорошому рівні з використанням сучасних обчислювальних засобів.

Магістрант гр. ІПЗСМ-18-3 Іорданов С.О. готовий до самостійної інженерної діяльності. Атестаційну роботу можна подати до захисту в ЕК за спеціальністю 121-«Інженерія програмного забезпечення», освітньо-науковою програмою Інженерія програмного забезпечення.

26/04/2020

доц. каф ІІІ, к.т.н. Ревенчук І.А.

ДОДАТОК В
АПРОБАЦІЯ РЕЗУЛЬТАТІВ РОБОТИ

60 Jubileuszowa Konferencja Studenckich Kół Naukowych ● Pionu Górniczego AGH

MATERIAŁY KONFERENCYJNE

Kraków, 5 grudnia 2019 r.



Główny Partner

SPIS TREŚCI

Sekcja I	Górnictwo, Wiertnictwo, Nafta i Gaz	3
Sekcja II	Geologia	15
Sekcja III	Geoturystyka i Geofizyka	32
Sekcja IV	Geodezja, Kartografia i Geoinformacja	40
Sekcja V	Budownictwo	51
Sekcja VI	Chemia i Ochrona Środowiska	61
Sekcja VII	Wentylacja, Klimatyzacja i Ogrzewnictwo	79
Sekcja VIII	Inżynieria Mechaniczna	94
Sekcja IX	Energetyka	113
Sekcja X	Akustyka	120
Sekcja XI	Informatyka	138
Sekcja XIIa	Inżynieria Produkcji i Jakości	170
Sekcja XIIb	Inżynieria Produkcji i Jakości	188
Sekcja XIII	Zarządzanie i Marketing	206
Sekcja XIV	Przedsiębiorczość Zrównoważona i Innowacyjna .	222
Sekcja XV	Rachunkowość i Finanse	243
Sekcja XVI	Technology, Society and Language	251

Oleksii BUDIANSKYI

Serhii IORDANOV

Koło Naukowe Modelowania w Finansach

Software Engineering Department at Kharkiv
National University of Radio Electronics



Koło naukowe

Modelowanie w finansach

USING ELK STACK AND. NET CORE TECHNOLOGIES IN CASE ANALYSIS OF THE UKRAINIAN VEHICLES OPERATIONS DATASET

Recently Ukraine government has created a portal of open data with different types of data sets related to versatile operations inside the country for all citizens. One of the largest open dataset of this site is a register of vehicles and their owners which contains all information about car operations such as initial registration, registration of imported or re-registration of a vehicle to a new owner. All operations related to vehicles are listed in the dataset. Single data row contains main information about the car (color, registration plate, capacity, brand, etc.), the owner (registration address, individual registration type) and the operation information (name, code).

This paper contains results of format the data to one style, collecting it in one database, analyze and visualization with the most interesting findings. At first to analyze all the data and find any dependencies it's need to formatted them and collected in one database, to be able to compare different records. There were about 11 million data records that should be filtered and aggregated quickly with the ability to display the results of aggregation as charts. After investigation of the available tools and databases Elasticsearch was chosen as the main database.

Elasticsearch is a distributed, RESTful search and analytics engine capable of addressing a growing number of use cases, it lets perform and combine many types of searches – structured, unstructured, geo, metric, etc. So, it is perfect for storing a huge number of data and performing searches. Elasticsearch use of Lucene Core under the hood which provides Java-based indexing and search technology, as well as spellchecking, hit highlighting and advanced analysis/tokenization capabilities. In addition, there is a great visualization tool for Elasticsearch named Kibana, which is an open-source data visualization dashboard for the database. It provides visualization capabilities on top of the content indexed on an Elasticsearch cluster.

In the register, the information was kept in several tar.gz archives which contained csv files with the operation rows. It was not a convenient way to analyze the data and the chosen database was tuned for JSON format. So, the next challenge was converting and transferring the data to the local Elasticsearch cluster.

To perform the operation we have created a C# tool, which parses csv data row, depending on a format, as files have a little bit different ways of naming, then converts data to standard view to be capable of making a comparison, converts the resulting entity to JSON format and sends to Elasticsearch. In order to perform that transformation, we have used CsvHelper and Json.NET (formally known as Newtonsoft JSON). They both are distributed under the licenses which are free for commercial use (MS-PL and Apache 2 for CsvHelper and MIT for Json.NET).

The created solution did the job correctly however the data processing speed was quite slow. The tool was able to process 20-25 data records per second on the local computer, which meant that 11 million data rows would have been processing for more than 150 hours. The most time-consuming operations were parsing the data from csv and converting the data to JSON. As all datafiles were approximately the same size, we decided to split the threads by data source.

The parallel solution worked much better and the throughput was about 100 data rows per second, so all the data were transferred to the Elasticsearch database in around 30-35 hours and became ready to be analyzed. The paper presents interest results of the Ukrainian Vehicles Operations Dataset analysis with many original illustrations.

Opiekun naukowy:

Associated professor, Volodymyr Kobziew PhD

ДОДАТОК Г
СЛІЙДИ ПРЕЗЕНТАЦІЇ

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Атестаційна робота магістра

Дослідження методів обробки даних у файлах для
тестування контенту на виявлення помилок щодо
оформлення тексту



1

Виконав:
ст. гр. ІПЗм-18-3
Іорданов С. О.

Науковий керівник:
к.т.н., доц.
Ревенчук



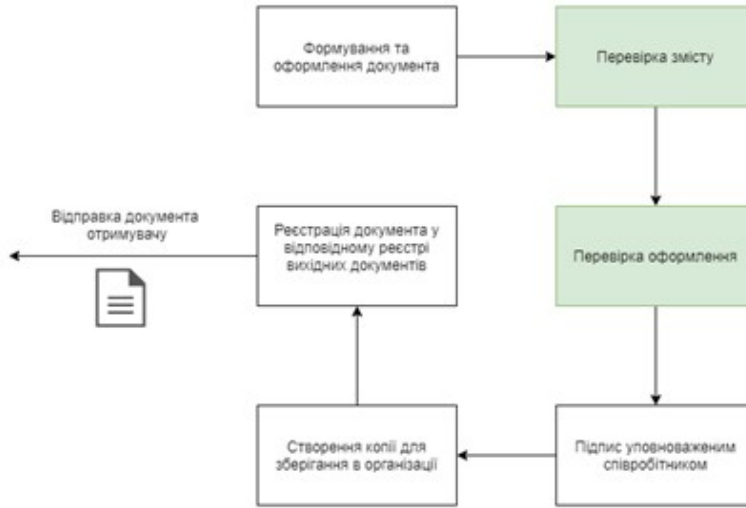
Мета роботи

- Аналіз процесу документообігу та перевірки форматування документів
- Аналіз існуючих рішень та особливостей документів формату docx
- Дослідження та розробка методів обробки docx документів
- Проектування та реалізація ПО з використанням досліджених методів
- Інтерфейс користувача та тестування системи
- Аналіз результатів роботи

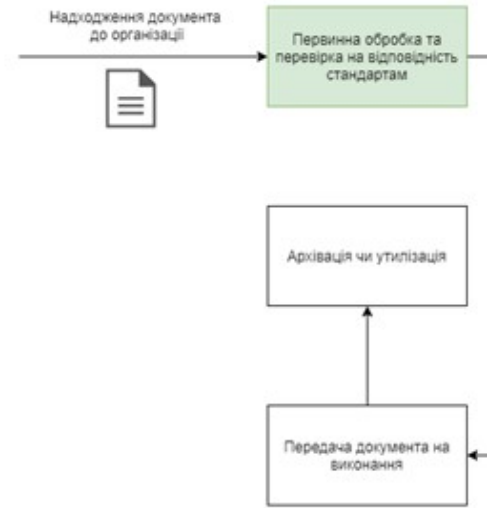


Аналіз проблеми

Документообіг вихідного документа






Документообіг вхідного документа





Аналіз існуючих рішень

	Рішення	Переваги	Недо
	Використання шаблонів docx	Вирішує проблему з форматуванням адже надає лише можливість занести до документу необхідну інформацію.	Можеуть бути викор невеликій кількості стандартні бланки, з
	FormatChecker від Star Group	Має досить гнучкі налаштування, дозволяє одразу виправляти знайденні помилки.	Працює лише з форм (зміна шрифту, вирів охоплює форматування цілому.
	Сервіс послуг форматування Proofers	Дозволяє описати усі необхідні правила форматування. Повертають виправлений документ	Кожний документ по виправляє людина. д часу, не підходить д та внутрішніх докум



Проблеми перевірки форматування docx

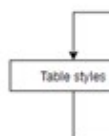
Відсутність інформації про розміщення тексту на сторінках

velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.	
>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut	3

Декларативне задання колонтитулів

```
<w:sectPr>  
  ...  
  <w:headerReference r:id="rId8" w:type="first"/>  
  <w:headerReference r:id="rId6" w:type="default"/>  
  ...  
</w:sectPr>
```

Складання таблиць





Існуючі методи обробки docx

- Редагування шаблонів (використання шаблонів Microsoft Word або заміна спецсимволів)
- Програмне створення/зміна XML структури документа (OpenXML з реалізацією WordprocessingML)
- Робота з COM (Component Object Model) Microsoft Word (програмне використання можливостей Microsoft Word)
- Конвертація документа у інші формати



Досліджені методи аналізу docx документів

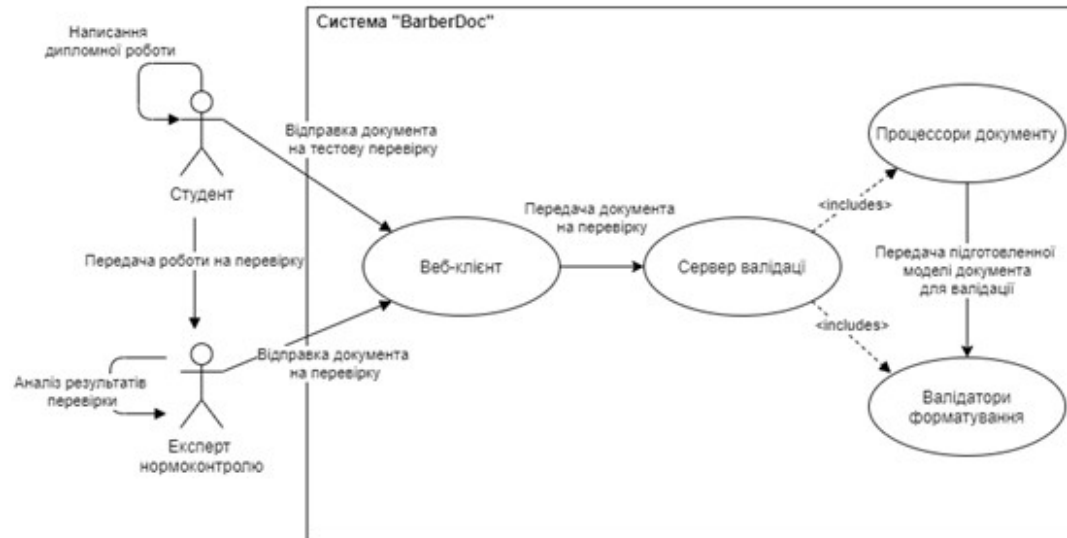
- Робота з XML структурою docx
- Комбінований аналіз документа у декількох форматах
- Використання проміжної структури даних





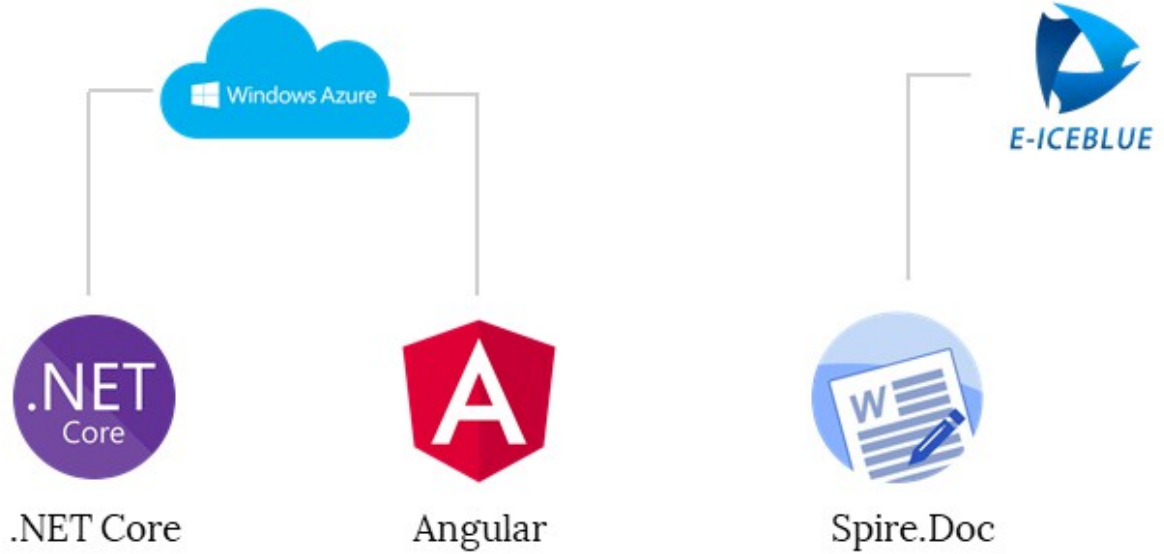
Постановка задачі та опис системи

Система для автоматизації процесу



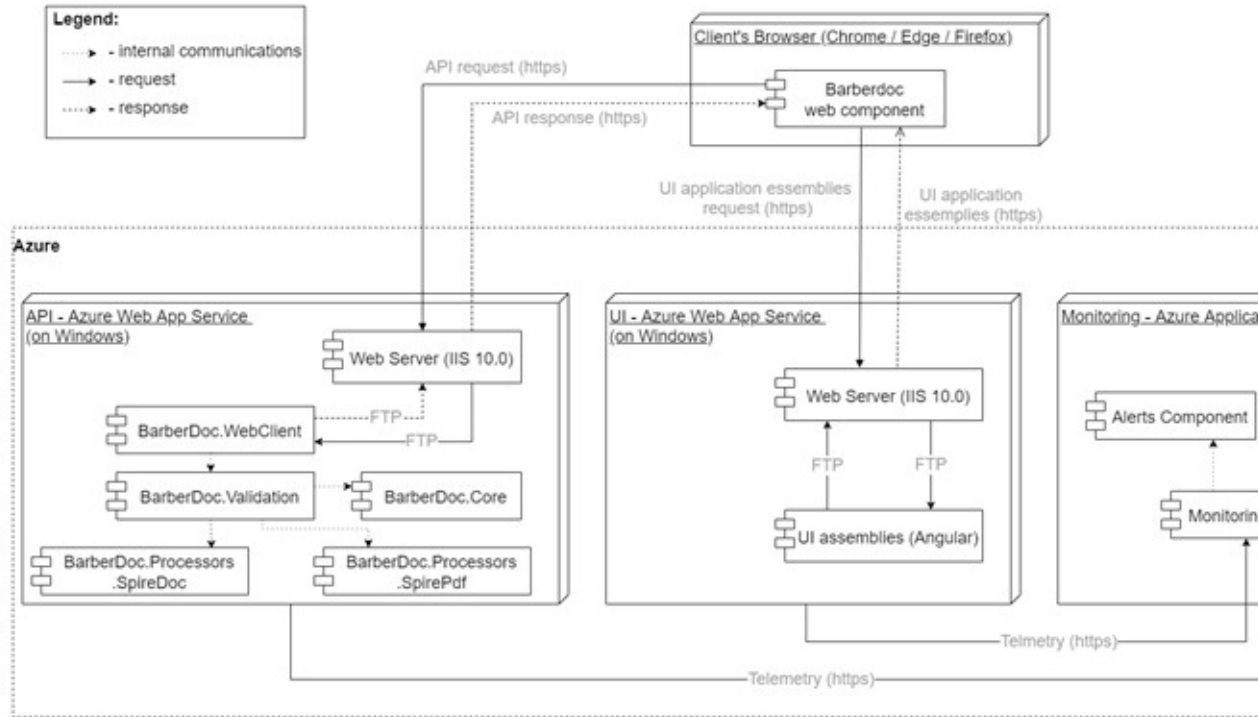


Вибір технологій розробки



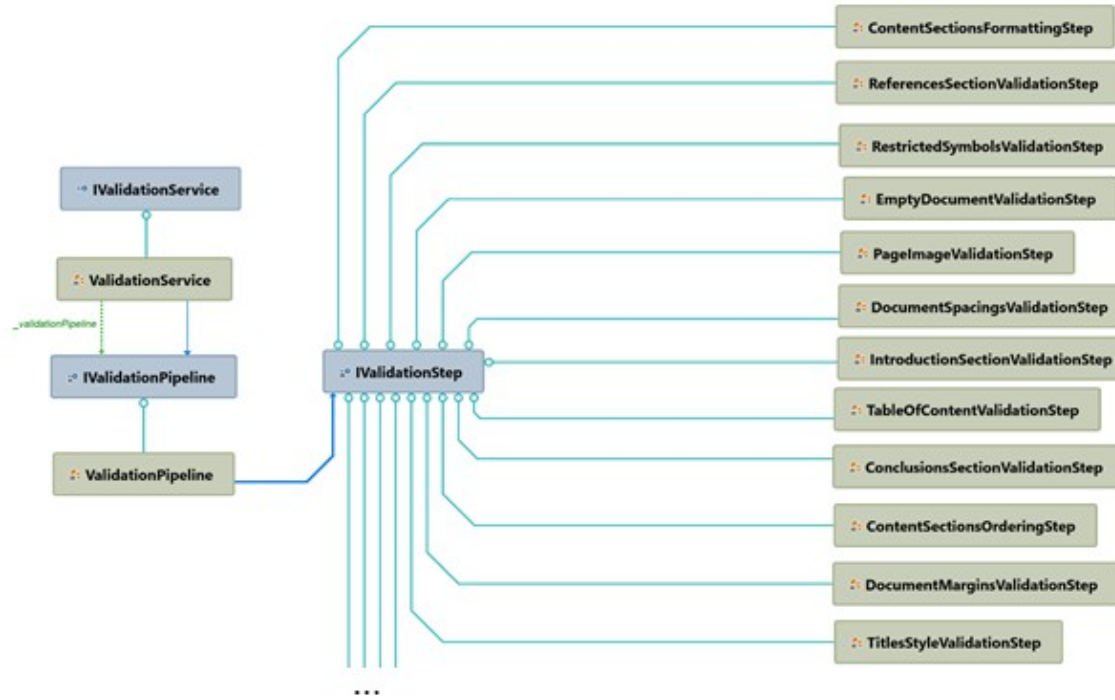


Дизайн системи





Дизайн системи





Приклад реалізації

```
// Ініціалізація масиву заголовків
var titles = new List<string>();

// Задання першої секції "ВСТУП"
var sectionTitle = introductionSectionTitle; // "ВСТУП"

// Цикл по секціях документа до моменту отримання секції з заголовком "ВИСНОВКИ"
do
{
    // Отримання останнього параграфу поточної секції
    if (GetSectionItems(document, sectionTitle).LastOrDefault() is Paragraph lastParagraph)
    {
        // Отримання назви наступної секції з останнього або наступного параграфа
        sectionTitle = GetNextSectionTitle(lastParagraph);

        // Збереження отриманої назви секції
        titles.Add(sectionTitle);
    }
    else
    {
        // Припинення циклу, якщо досягнуто кінця документа
        break;
    }
}
while (!EqualsIgnoringCase(sectionTitle, conclusionsSectionTitle)); // "ВИСНОВКИ"

// Вилучення секції "ВИСНОВКИ" та повернення результату роботи алгоритму
return titles.Where(title => !EqualsIgnoringCase(title, conclusionsSectionTitle));
```



Приклад реалізації

```
// Задання масиву помилок
var errors = new List<ValidationError>();

// Перевірка відповідності назви шрифту (наприклад: Times New Roman)
var isValidFontFamily = requiredFont.FontFamily == paragraph.BreakCharacterFormat.FontName;

// Перевірка розміру шрифту (використовується допустима похибка рівна 0.00001)
var isValidFontSize = Math.Abs(paragraph.BreakCharacterFormat.FontSize - requiredFont.Size) < SizeInfelicity;

// Перевірка кольору шрифту
var isValidFontColor = ColorRgb.ConvertToRgb(paragraph.BreakCharacterFormat.TextColor).Equals(requiredFont.Color);

// Висновок щодо правильності шрифту у параграфі
var isValidFont = isValidFontFamily && isValidFontSize && isValidFontColor;

// Формування помилки у разі неправильності шрифту
if (paragraph.Text.Length > 0 && !isValidFont)
{
    var errorType = isTableParagraph ? ErrorType.Warning : ErrorType.Error;

    var message =
        $"Invalid font: {paragraph.BreakCharacterFormat.FontName}" +
        $" {paragraph.BreakCharacterFormat.FontSize}" +
        $" {ColorRgb.ConvertToRgb(paragraph.BreakCharacterFormat.TextColor)}." +
        paragraph.GetTextSubstring(_settings.ErrorMessageNumberOfChars, isTableParagraph);

    errors.Add(new ValidationError(errorType, typeof(TContext), message));
}


return errors;
```





Інтерфейс користувача


Welcome back to BarberDoc

Step 1: Upload your work How does it work?


Drag and drop MS Word file here
or
[Browse for file](#)


Drag and drop PDF file here
or
[Browse for file](#)

 sample.docx
42.26 KB ✕

 sample.pdf
149.6 KB ✕

[Start validation](#) [Clear results](#)

Step 2: Analyze the results Show validation settings

#	Type	Message	Source
1	Error	Current text on the title page is invalid. Expected: "2020. n." but was: "n."	TitlePageValidationStep



Тестування

Test Case Examples							
ID	Description	Precondition	Test Steps	Expected Result	Actual result	Priority	Test Result
1	Paragraph font validation	There is a .docx document with valid font in all the paragraphs. The system is configured to validate Times New Roman font with 14 size. A: Actor (user) S: System	A: Changes a font for several paragraphs in the document A: Uploads .docx document to the system and starts validation S: Validates the documents, returns a list of validation errors A: Checks the result error list	All the changed paragraphs are present in the error list	-	High	PASS
2	Sections ordering validation	There is a .docx document with table of content and section titles in the correct order A: Actor (user) S: System	A: Changes the ordering of the document sections A: Uploads .docx document to the system and starts validation S: Validates the documents, returns a list of validation errors A: Checks the result error list	The error list contains the errors related to wrongly placed sections	-	High	PASS
3	Restricted symbols validation	There is a document without restricted symbols (see the list in SRC). A: Actor (user) S: System	A: Changes a few restricted symbols to the document A: Uploads .docx document to the system and start validation S: Validates the documents, returns a list of validation errors A: Checks the result error list	The error list contains the paragraphs with the inserted restricted symbols	-	Medium	PASS



Можливості використання

- Поточна реалізація: автоматизація процесу нормоконтролю в інших вищих навчальних закладах
- Перспективи: дженералізація системи для використання в будь-якій сфері, будь-якою організацією
- Використання досліджених методів для автоматичного виявлення типу документа

Дякую за увагу!

