



Софийски университет „Св. Кл. Охридски“

Факултет по математика и информатика

*Бакалавърска програма
„Софтуерно инженерство“*



Предмет: XML технологии за семантичен Уеб

Зимен семестър, 2017/2018 год.

Тема №01: „Каталог на хотели“

Курсов проект

Автори:

Кристина Димова, фак. номер 61886

Невена Гаджева, фак. номер 61938

януари, 2018

София

Съдържание

1	Въведение	3
2	Анализ на решението	3
2.1	Работен процес	3
2.2	Структура на съдържанието	3
2.3	Тип и представяне на съдържанието	9
3	Дизайн	10
4	Тестване	11
5	Заклучение и възможно бъдещо развитие	15
6	Разпределение на работата	15
7	Използвани литературни източници и Уеб сайтове	16

1 Въведение

Целта на проекта е да се създаде каталог на хотелите в България по региони. Реализиран е чрез XML документи с текстово и графично съдържание, описващо възможните характеристики на всеки един хотел в страната. Каталогът представя графичното съдържание чрез XML единици(entities). Връзките между региони и хотели, между вериги от хотели и хотел от веригата и обратното са описани чрез атрибути ID/IDREF. Представено е описанието на 8 хотела. Съдържанието на XML документа е валидирано чрез подходящо изграден за целта DTD документ, след което е представено в PDF документ чрез генерация посредством XSLT.

В настоящия документ са представени стъпките, които са следвани при създаване на каталога, неговата структура и дизайн.

2 Анализ на решението

2.1 Работен процес

Съдържанието на каталога е представено под формата на XML документ, като графичните елементи са добавени като частни външни единици (entities). Валидирането на структурата и съдържанието на XML документа е извършено чрез подходящо изграден за целта DTD документ. Представянето на XML съдържанието под формата на PDF документ е направено с помощта на XSL документ, описващ трансформацията на информацията, както и средата Altova XMLSpy 2018, част от която са XSLT-процесорът и Apache FOP.

2.2 Структура на съдържанието

0. **catalogue** – коренов елемент на XML документа. Той съдържа елементите: **regions**, **chains**, **hotels**. Според DTD схемата е задължително да има поне по един екземпляр от тези три елемента в XML файла
1. **regions** – пряк наследник на **catalogue** елемента. Негов поделемент е **region**. Според DTD схемата, елементът **regions** се състои от един или няколко елемента **region**.
 - 1.1. **region** - пряк наследник на **regions** елемента. Негов поделемент е **region_name**. Според DTD схемата, елементът **region** се състои от един елемент **region_name**, който носи като информация името на съответния регион. Елементът **region** има два атрибута: **regionID** и **hotelRef**.
 - 1.1.1. **regionID** – атрибут на елемента **region**. Според DTD схемата този атрибут е от тип ID и е означен като задължителен. Всеки регион ще има уникално **regionID**, което ще се реферира от хотелите, които се намират на територията на съответния регион.
 - 1.1.2. **hotelRef** – атрибут на елемента **region**. Според DTD схемата този атрибут е от тип IDREF и е означен като незадължителен. Всеки хотел ще има уникален **hotelID**, който ще се реферира от атрибута **hotelRef**. По този начин се изгражда връзка между региона и хотелите, които се намират в него.

- 1.1.3.**region_name** - пряк наследник на **region** елемента. Според DTD схемата, елементът **region_name** е от тип PCDATA. Той съдържа името на съответния регион.
2. **chains** – пряк наследник на **catalogue** елемента. Негов подлемент е **chain**. Според DTD схемата, елементът **chains** се състои от един или няколко елемента **chain**.
- 2.1. **chain** - пряк наследник на **chains** елемента. Негови поделементи са: **chain_name**, **chain_description**, **chain_logo**. Според DTD схемата, елементът **chain** се състои от един елемент **chain_name**, който носи като информация името на съответната верига от хотели. Елементите **chain_description**, **chain_logo** не са задължителни, може да присъства описание и лого на хотела, но могат да бъдат пропуснати. Елементът **chain** има два атрибута: **chainID** и **hotelRef**.
- 2.1.1.**chainID** – атрибут на елемента **chain**. Според DTD схемата този атрибут е от тип ID и е означен като задължителен. Всяка хотелска верига ще има уникално **chainID**, което ще се реферира от хотелите, които са част от тази верига.
- 2.1.2.**hotelRef** – атрибут на елемента **chain**. Според DTD схемата този атрибут е от тип IDREF и е означен като незадължителен. Всеки хотел ще има уникален **hotelID**, който ще се реферира от атрибута **hotelRef**. По този начин се изгражда връзка между съответната верига от хотели и хотелите, които се включват в нея.
- 2.1.3.**chain_name** - пряк наследник на **chain** елемента. Според DTD схемата, елементът **chain_name** е от тип PCDATA. Той съдържа името на съответната хотелска верига.
- 2.1.4.**chain_description** - пряк наследник на **chain** елемента. Според DTD схемата, елементът **chain_description** е от тип PCDATA. Той съдържа обща информация за съответната хотелска верига.
- 2.1.5.**chain_logo** - пряк наследник на **chain** елемента. Той съдържа елемента **logo**. Според DTD схемата, елементът **chain_logo** не е задължително да съдържа поделемента **logo**.
- 2.1.5.1.**logo** - пряк наследник на **chain_logo** елемента. Според DTD схемата, елементът **logo** е от тип EMPTY. Той притежава един атрибут **href**.
- 2.1.5.2.**href** – атрибут на елемента **logo**. Според DTD схемата този атрибут е от тип ENTITY и е означен като задължителен. Тази референция е към ресурс, който не е в XML формат, затова е използвана ключова дума, за тип на атрибута, ENTITY. Тази връзка позволява използването на изображения в проекта.
3. **hotels** – пряк наследник на **catalogue** елемента. Негов поделемент е **hotel**. Според DTD схемата, елементът **hotels** се състои от един или няколко елемента **hotel**.
- 3.1. **hotel** - пряк наследник на **hotels** елемента. Негови поделементи са: **hotel_name**, **category**, **hotel_type**, **address**, **phone**, **email**, **site**, **description**, **more_info**, **rooms**, **image**. Според DTD схемата, елементите: **hotel_name**, **category**, **hotel_type**, **address** е задължително да има поне по един екземпляр от тях в XML документа. Елементите: **phone**, **email**, **site**, **description**, **more_info**, **rooms**, **image** не са задължителни. Елементът **hotel** има три атрибута: **hotelID**, **chainRef**, **regionRef**.
- 3.1.1.**hotelID** – атрибут на елемента **hotel**. Според DTD схемата този атрибут е от тип ID и е означен като задължителен. Всеки хотел ще има уникално **hotelID**, което ще се реферира от хотелската верига, ако даденият хотел принадлежи към такава, и от региона, в който се намира.
- 3.1.2.**chainRef** – атрибут на елемента **hotel**. Според DTD схемата този атрибут е от тип IDREF и е означен като незадължителен. Всяка хотелска верига ще има уникален **chainID**, който ще се реферира от атрибута **chainRef**. По този начин се изгражда

връзка между даден хотел и хотелската верига, към която принадлежи, ако има такава.

- 3.1.3. **regionRef** – атрибут на елемента **hotel**. Според DTD схемата този атрибут е от тип IDREF и е означен като незадължителен. Всеки регион ще има уникален **regionID**, който ще се реферира от атрибута **regionRef**. По този начин се изгражда връзка между даден хотел и региона, в който се намира.
- 3.1.4. **hotel_name** - пряк наследник на **hotel** елемента. Според DTD схемата, елементът **hotel_name** е от тип PCDATA. Той съдържа името на съответния хотел.
- 3.1.5. **category** - пряк наследник на **hotel** елемента. Според DTD схемата, елементът **category** е от тип PCDATA. Той съдържа информация за броя на звездите на съответния хотел.
- 3.1.6. **hotel_type** - пряк наследник на **hotel** елемента. Според DTD схемата, елементът **hotel_type** е от тип PCDATA. Носи като информация типа на съответния хотел.
- 3.1.7. **address** - пряк наследник на **hotel** елемента. Той съдържа елементите: **city** и **street**. Според DTD схемата е задължително да има поне един екземпляр на елемента **city** в XML документа, а **street** може да присъства но не е задължително.
 - 3.1.7.1. **city** - пряк наследник на **address** елемента. Според DTD схемата, елементът **city** е от тип PCDATA. Той съдържа информация за града, в който се намира даден хотел.
 - 3.1.7.2. **street** - пряк наследник на **address** елемента. Според DTD схемата, елементът **street** е от тип PCDATA. Той съдържа информация за улицата, на която се намира съответния хотел.
- 3.1.8. **phone** - пряк наследник на **hotel** елемента. Според DTD схемата, елементът **phone** е от тип PCDATA. Предоставя номер за връзка с хотела.
- 3.1.9. **email** - пряк наследник на **hotel** елемента. Според DTD схемата, елементът **email** е от тип PCDATA. Предоставя email-а на хотела, за да се правят запитвания по важни въпроси.
- 3.1.10. **site** - пряк наследник на **hotel** елемента. Според DTD схемата, елементът **site** е от тип PCDATA. Предоставя сайта на съответния хотел, което позволява на клиентите да получат повече информация за него.
- 3.1.11. **description** - пряк наследник на **hotel** елемента. Според DTD схемата, елементът **description** е от тип PCDATA. Предоставя кратка информация за хотела, която може да създаде обща представа на клиентите за възможностите и удобствата му.
- 3.1.12. **more_info** - пряк наследник на **hotel** елемента. Той съдържа елемента **service**. Според DTD схемата не е задължително да има екземпляр на елемента **service** в XML файла.
 - 3.1.12.1. **service** - пряк наследник на **more_info** елемента. Според DTD схемата, елементът **service** има смесено съдържание. Той може да съдържа текст (PCDATA) или поделемента **characteristic** от тип PCDATA. Елемент **service** може да е изграден от няколко последователности от текст и/или елементи **characteristic**. Предоставя информация за допълнителни услуги, предлагани от хотела.
 - 3.1.12.1.1. **characteristic** - пряк наследник на **service** елемента. Според DTD схемата, елементът **characteristic** е от тип PCDATA. Предоставя информация за характеристиките на съответната услуга. Този елемент има един атрибут – **type**.

- 3.1.12.1.1. **type** - атрибут на елемента **characteristic**. Според DTD схемата този атрибут е от тип CDATA и е означен като незадължителен. Показва дали дадена характеристика на услуга е забранена или безплатна.
- 3.1.13. **rooms** – пряк наследник на **hotels** елемента. Негови поделемени са: **rooms_description**, **rooms_images**, **room**. Според DTD схемата, елементите **rooms_description**, **rooms_images** могат да не присъстват като екземпляри в XML документа, но елементът **rooms** задължително трябва да има поне един елемент **room**.
- 3.1.13.1. **rooms_description** - пряк наследник на **rooms** елемента. Според DTD схемата, елементът **rooms_description** е от тип PCDATA. Той съдържа допълнителна, обобщена информация за стаите на хотела.
- 3.1.13.2. **rooms_images** - пряк наследник на **rooms** елемента. Той се състои от елемента **room_image**. Според DTD схемата е задължително да има поне един екземпляр на елемента **room_image** в XML файла.
- 3.1.13.2.1. **room_image** - пряк наследник на **rooms_images** елемента. Той се състои от елемента **image_src**. Според DTD схемата има точно един поделемени **image_src** във всеки елемент **room_image** в XML файла. Този елемент има един атрибут – **type**.
- 3.1.13.2.1.1. **type** – атрибут на елемента **room_image**. Според DTD схемата този атрибут е от тип CDATA и е означен като незадължителен. Използва се, когато имаме повече от един елемент **room_image**, за да може да се прави разлика между тях.
- 3.1.13.2.1.2. **image_src** - пряк наследник на **room_image** елемента. Според DTD схемата, елементът **image_src** е от тип EMPTY. Той притежава един атрибут **href**.
- 3.1.13.2.1.2.1. **href** – атрибут на елемента **image_src**. Според DTD схемата този атрибут е от тип ENTITY и е означен като задължителен. Тази референция е към ресурс, който не е в XML формат, затова е използвана ключова дума, за тип на атрибута ENTITY. Тази връзка позволява използването на изображения в проекта, които да дадат визуална представа за стаите, с които разполага съответния хотел.
- 3.1.13.3. **room** - пряк наследник на **rooms** елемента. Той се състои от елементите: **type**, **price**. Според DTD схемата, двата поделемени трябва да присъстват в тази последователност и то по точно един път в XML документа.
- 3.1.13.3.1. **type** - пряк наследник на **room** елемента. Според DTD схемата, елементът **type** е от тип PCDATA. Информацията, която носи, определя типа на стаята.
- 3.1.13.3.2. **price** - пряк наследник на **room** елемента. Той се състои от елемента **season**. Според DTD схемата, поделемени трябва да присъства точно един път в съдържанието на елемента **room**.
- 3.1.13.3.2.1. **season** - пряк наследник на **price** елемента. Той се състои от елементите: **low**, **medium**, **high**, **all**. Според DTD схемата, поделемени: **low**, **high**, **all** могат да се срещнат веднъж или нито един път, докато **medium** може да се използва повече от един път.

- 3.1.13.3.2.1.1. **low** - пряк наследник на **season** елемента. Според DTD схемата, елементът **low** е от тип PCDATA. Той предоставя цената за нощувка в даден хотел през най-ниския сезон.
- 3.1.13.3.2.1.2. **medium** - пряк наследник на **season** елемента. Според DTD схемата, елементът **medium** е от тип PCDATA. Той предоставя цената за нощувка, в даден хотел през междинните сезони. Този елемент притежава атрибут **type**.
- 3.1.13.3.2.1.2.1. **type** - атрибут на елемента **medium**. Според DTD схемата този атрибут е от тип CDATA и е означен като незадължителен. Използва се за различаване на междинните сезони, когато в ценоразписа на даден хотел има повече от един междинен сезон.
- 3.1.13.3.2.1.3. **high** - пряк наследник на **season** елемента. Според DTD схемата, елементът **high** е от тип PCDATA. Той предоставя цената за нощувка в даден хотел през най-високия сезон.
- 3.1.13.3.2.1.4. **all** - пряк наследник на **season** елемента. Според DTD схемата, елементът **all** е от тип PCDATA. Той предоставя цената за нощувка в даден хотел, която е една и съща всички сезони.
- 3.1.14. **image** - пряк наследник на **hotel** елемента. Той съдържа елемента **src**. Според DTD схемата е задължително да има точно един елемент **src**, поделен на **image**.
- 3.1.14.1. **src** - пряк наследник на **image** елемента. Според DTD схемата, елементът **src** е от тип EMPTY. Той притежава един атрибут - **href**.
- 3.1.14.1.1. **href** – атрибут на елемента **src**. Според DTD схемата този атрибут е от тип ENTITY и е означен като задължителен. Тази референция е към ресурс, който не е в XML формат, затова е използвана ключова дума за тип на атрибута ENTITY. Тази връзка позволява използването на изображения в проекта, които да дадат визуална представа за хотела и неговия облик.

```
<catalogue>
  <regions>
    <region regionID hotelRef>
      <region_name></region_name>
    </region>
  </regions>
  <chains>
    <chain chainID hotelRef>
      <chain_name></chain_name>
      <chain_description></chain_descripti
      <chain_logo>
        <logo href/>
      </chain_logo>
    </chain>
  </chains>
```


2.3 Тип и представяне на съдържанието

Съдържанието на каталога е представено графично в PDF формат. То се състои от текст, таблици и списъци, създадени чрез XSL шаблони, и графични изображения (общо 27 на брой). Снимковият материал е в JPG формат и е свален от уеб сайтовете на описаните хотели (Източници 2-18).

- Заглавна страница – bulgaria.jpg (images)
- Верига A Hotels
 - Лого – aHotelsLogo.jpg (images/logos)
- Верига Lion Hotels
 - Лого – lionLogo.jpg (images/logos)
- Hotel Aronia Beach – AroniaBeach.jpg (images/aroniaBeach)
 - Стаи
 - room1.jpg (images/aroniaBeach)
 - room2.jpg (images/aroniaBeach)
- Hotel Astera – Astera.jpg (images/astera)
 - Стаи
 - room1.jpg (images/astera)
 - room2.jpg (images/astera)
- Hotel Atlas – Astera.jpg (images/atlas)
 - Стаи
 - room1.jpg (images/atlas)
 - room2.jpg (images/atlas)
- Hotel Restaurant History – history.jpg (images/history)
 - Стаи
 - room1.jpg (images/history)
 - room2.jpg (images/history)
- Hotel Lion Bansko – lionBansko.jpg (images/lionBansko)
 - Стаи

- room1.jpg (images/lionBansko)
- room2.jpg (images/lionBansko)
- Hotel Lion Sofia – lionSofia.jpg (images/lionSofia)
 - Стаи
 - room1.jpg (images/lionSofia)
 - room2.jpg (images/lionSofia)
- Hotel Lion Sunny Beach – lionSunnyBeach.jpg (images/lionSunnyBeach)
 - Стаи
 - room1.jpg (images/lionSunnyBeach)
 - room2.jpg (images/lionSunnyBeach)
- Hotel Rostov – rostov.jpg (images/rostov)
 - Стаи
 - room1.jpg (images/rostov)
 - room2.jpg (images/rostov)

3 Дизайн

За създаването на каталога са използвани следните технологии:

- XML version 1.0
- DTD version 1.0
- XSLT version 1.0
- XSL-FO
- Apache-FO

Съдържанието на каталога е представено под формата на XML документ. Графичните изображения са представени в XML документа чрез използването на частни външни единици (entities), които реферират към изображенията. Всеки един регион, верига и хотел притежава уникален идентификатор под формата на атрибут ID (**regionID** – идентификатор на регион, **chainID** – идентификатор на верига от хотели, **hotelID** – идентификатор на хотел). Този идентификатор служи за осъществяване на връзка между регион и хотели (**hotelRef**), намиращи се в този регион, верига от хотели и хотели от веригата (**hotelRef**) и обратното – между хотел и верига (**chainRef**), както и между хотел и региона (**regionRef**), в който се намира.

Това става с помощта на референции към идентификаторите – атрибут IDREF (**hotelRef**, **chainRef**, **regionRef**).

Съдържанието на XML документа е валидирано с помощта на DTD документ и онлайн средата XML Validator, предоставена от www.xmlvalidation.com.

Елементите на XML документа са валидирани чрез декларации на елементи в DTD документа и посочване на типа текстово съдържание, което могат да приемат, както и на честотата им на срещане. Напр.:

```
<!ELEMENT hotel (hotel_name, category, hotel_type, address, phone?, email?, site?, description?,  
more_info?, rooms?, image?)>  
  
<!ELEMENT hotel_name (#PCDATA)>  
  
<!ELEMENT service (#PCDATA | characteristic)*>  
  
<!ELEMENT logo EMPTY>
```

Атрибутите на елементите в XML документа са валидирани чрез декларации на атрибути в DTD документа и посочване на типа съдържание, което могат да приемат, както и това, дали задължително трябва да присъстват, или не. Напр.:

```
<!ATTLIST region regionID ID #REQUIRED>  
  
<!ATTLIST region hotelRef IDREFS #IMPLIED>  
  
<!ATTLIST logo href ENTITY #REQUIRED>  
  
<!ATTLIST characteristic type CDATA #IMPLIED>
```

Идентификаторите и референциите към тях в XML документа са валидирани чрез атрибути от тип ID и IDREF в DTD документа:

```
<!ATTLIST region regionID ID #REQUIRED>  
  
<!ATTLIST region hotelRef IDREFS #IMPLIED>
```

Единиците (entities) в XML документа са валидирани чрез декларации на частни външни единици в DTD документа:

```
<!ENTITY history SYSTEM "images/history/history.jpg" NDATA jpg>
```

Всички DTD декларации са разположени във външен за XML документа DTD документ, като първият е свързан с втория. Единствено декларациите на единиците са вградени в XML документа.

Трансформацията на XML текстовото съдържание в PDF формат е реализирана чрез XSL документ и средата Altova XMLSpy 2018, част от която са XSLT-процесорът и Apache FOP, който служи за преобразуване на XML в PDF документи.

В XSL документа са създадени редица шаблони, с помощта на които се представя графично текстовото съдържание на XML документа. Всяка страница се състои от няколко блока, в които са разположени отделните елементи от съдържанието, представено в XML документа. Чрез шаблоните се извличат необходимата информация и графичните изображения, представени под формата на единици (entities), и се разполагат в съответните блокове. Използвайки XSL-FO са зададени размерите на страниците, отстоянията между отделните блокове и елементи, както и техните размери, цвета, шрифта, стила, размера и подравняването на текстовото съдържание и са създадени няколко таблици и списъци.

Самото генериране на PDF документа се извършва от Altova XMLSpy 2018, като за целта се подава XSL документа с описаните трансформации и XML документа, съдържащ текстовите данни.

4 Тестване

След създаване на XML документа, той бе валидиран чрез подходящо създаден за целта DTD документ. За реализация на самата валидация бе използван инструментът XML validator, намиращ се на следния адрес: www.xmlvalidation.com.

За тестване на графичното представяне на XML съдържанието в PDF документ бе използвана средата Altova XMLSpy 2018, която получава като вход XSL и XML документ, валидира съдържанието им и генерира PDF документ, който показва как са визуализирани данните.

The screenshot shows the 'Validate an XML file' page of the XML validation tool. At the top, there is a navigation bar with links: '<?xml?>', 'Documentation', 'An example', and 'About'. The main heading is 'Validate an XML file', followed by a subtext: 'Read here how to validate your XML files (including referenced DTDs) online with just a few mouse clicks.' Below this, there is a large text area with the prompt 'Please copy your XML document in here:'. To the right of this area is a black-bordered box containing the figure caption. Below the text area, there is a section 'Or upload it:' with a 'Choose File' button and a file name 'XML_Hotels..._61938.xml'. At the bottom, there is a checkbox labeled 'Validate against external XML schema' and a 'validate' button.

<?xml?> Documentation An example About

Validate an XML file

Read here how to validate your XML files (including referenced DTDs) online with just a few mouse clicks.

Please copy your XML document in here:

Or upload it:
Choose File XML_Hotels..._61938.xml

The validation check is performed against any XML schema or DTD declared inside the XML document.
If neither an XML schema nor a DTD is declared, only a syntax check is performed.
To validate the XML document against an external XML schema, click below.

☐ Validate against external XML schema

validate

Фиг. 2 – Валидиране на XML документа – избор на XML документ

<?xml?>

DocumentationAn exampleAbout

Validate an XML file

Read here how to [validate your XML files \(including referenced DTDs\)](#) online with just a few mouse clicks.

The file XML_HotelsCatalogue_1_61886_61938.dtd is being referenced. Please copy it in here, so that the validation can continue:

Or upload it:

Choose FileXML_Hotels..._61938.dtd

continue validation

Фиг. 3 – Валидиране на XML документа – избор на DTD документ

<?xml?>

DocumentationAn exampleAbout

Validate an XML file

Read here how to [validate your XML files \(including referenced DTDs\)](#) online with just a few mouse clicks.

No errors were found

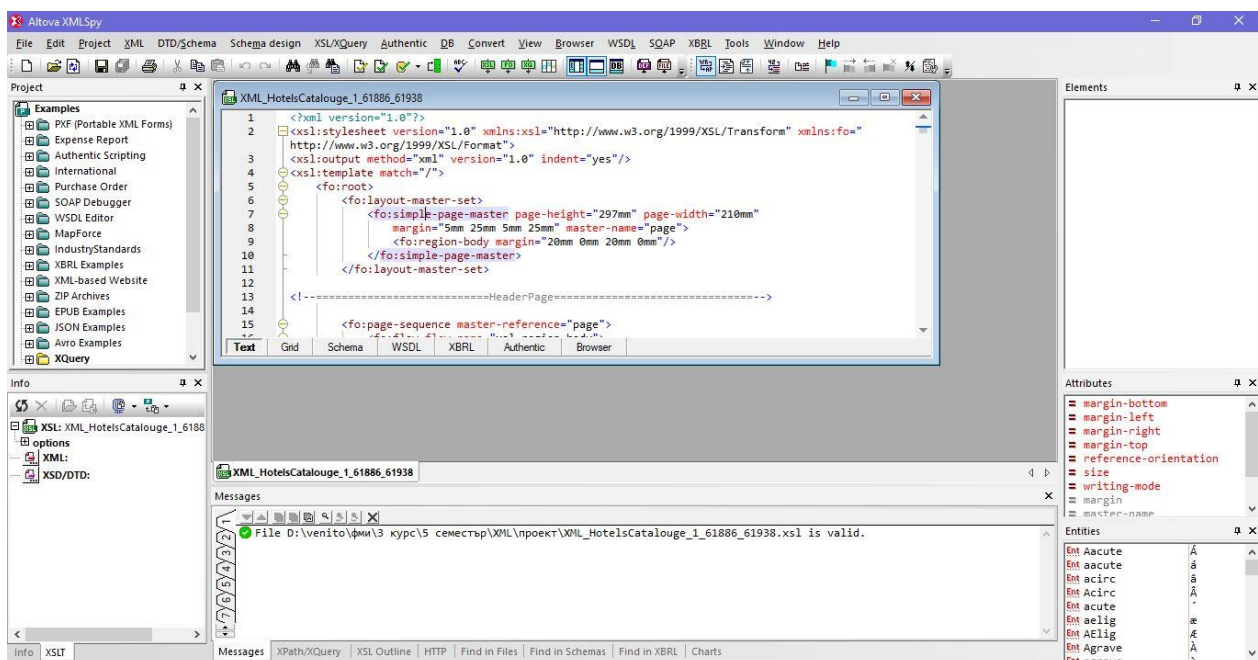
The following files have been uploaded so far:

XML document:⌵

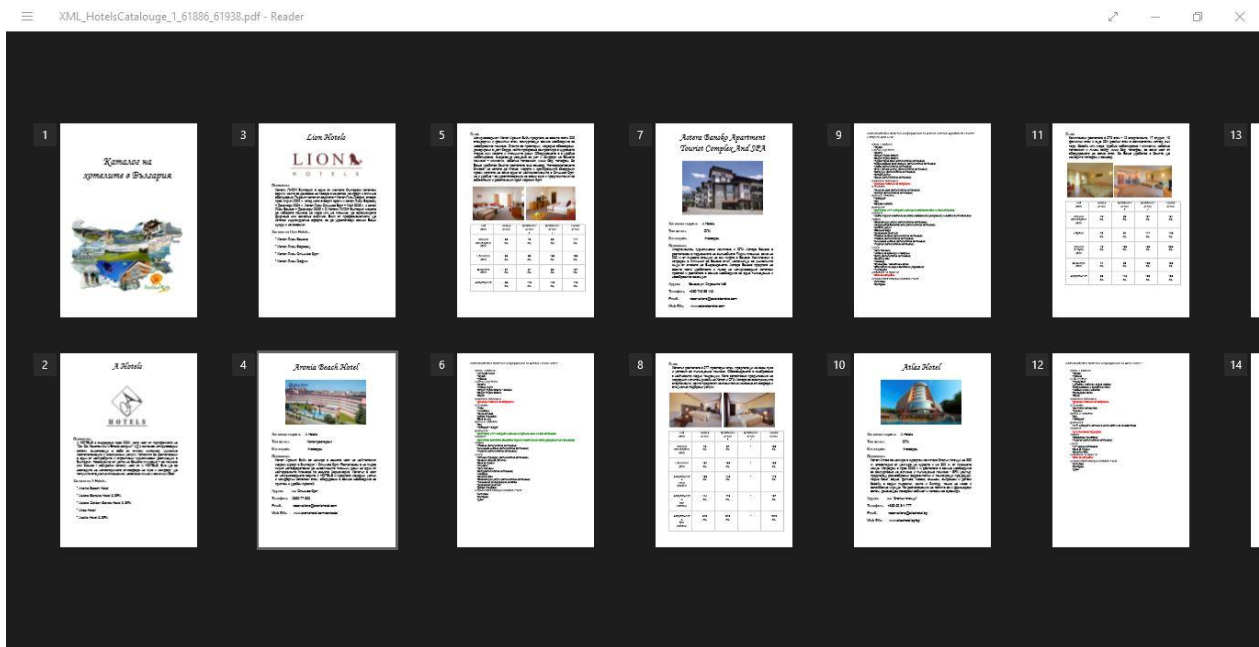
XML_HotelsCatalogue_1_61886_61938.dtd⌵

Click on any file name if you want to edit the file.

Фиг. 4 – Валидиране на XML документа – резултат от валидацията



Фиг. 5 – Трансформиране на XML съдържанието с помощта на XSLT



Фиг. 6 – Резултат от трансформацията

5 Заключение и възможно бъдещо развитие

Каталогът, който е представен като краен резултат, може да се използва като шаблон за представяне на информация за различни хотели и хотелски вериги. Дизайнът на каталога е стандартизиран и семпъл, което го прави лесно четим и разбираем. Самата информация е въведена ръчно в XML документа, като в следствие това може да се автоматизира и да се генерира от база данни, което би улеснило многократно създаването на каталог за голям брой хотелски вериги и хотели. Използвайки структурата на XSL документа, може да се добави разнообразни типове информация – като напр. за това, в кой град какви хотели има, какви обекти има в близост на хотелите, сортиране по някаква категория и др.

6 Разпределение на работата

Работата по проекта беше разпределена по следния начин:

- Кристина намери и представи информацията за хотелската верига A Hotels, като от нея са включени хотелите Aronia Beach, Astera Bansko и Atlas, както и за хотела Rostov
- Невена намери и представи информацията за хотелската верига Lion Hotels, като от нея са включени хотелит Lion Bansko, Lion Sofia и Lion Sunny Beach, както и за хотела History

Структурата на съдържанието на XML документа, DTD документа и дизайнът на PDF документа, описан в XSL документа бяха измислени и разработени съвместно.

7 Използвани литературни източници и Уеб сайтове

1. Бончев, Б., XML технологии, София, 2015
2. A Hotels – www.a-hotels.bg
3. Aronia Beach Hotel – www.aroniahotel.com
4. Aronia Beach Hotel (Booking.com) – www.booking.com/hotel/bg/aronia-beach.bg.html
5. Astera Bansko – www.asterabansko.com
6. Astera Bansko (Booking.com) – www.booking.com/hotel/bg/astera-bansko.bg.html
7. Atlas Hotel – www.atlashotel.bg
8. Atlas Hotel (Booking.com) – www.booking.com/hotel/bg/atlas.bg.html
9. Hotel Restaurant History (Booking.com) – www.booking.com/hotel/bg/history.bg.html
10. Lion Hotels – www.hotelslion.bg
11. Lion Hotel Bansko – www.bansko.hotelslion.bg
12. Lion Hotel Bansko (Booking.com) – www.booking.com/hotel/bg/lionbansko.bg.html
13. Lion Hotel Sunny Beach – www.sunnybeach.hotelslion.bg
14. Lion Hotel Sunny Beach (Booking.com) – www.booking.com/hotel/bg/lion-sunny-beach.bg.html
15. Lion Hotel Sofia – www.sofia.hotelslion.bg
16. Lion Hotel Sofia (Booking.com) – www.booking.com/hotel/bg/hotellionsof.bg.html
17. Hotel Rostov – www.rostov.bg
18. Hotel Rostov (Booking.com) – www.booking.com/hotel/bg/rostov.bg.html
19. Изображение Bulgaria.jpg – www.blackseavillas.net/image/About%20Bulgaria/bulgaria1.jpg
20. XML Validator – www.xmlvalidation.com