



**Софийски университет „Св. Кл.
Охридски“**

Факултет по математика и информатика



*Бакалавърска програма
„Софтуерно инженерство“*

Предмет: XML технологии за семантичен Уеб

Зимен семестър, 2022/2023 год.

Тема №01: „Каталог на хотели - 1“

Курсов проект

Автори:

Петя Петрова, фак. номер 62630

Радослав Велков, фак. номер 62528

януари, 2023 г.

София

Съдържание

1	Въведение	3
1.1	Избор на тема.....	3
1.2	Актуалност на избраната тема.....	3
1.3	Какъв точно проблем решава проекта и в какъв контекст е този проблем.....	3
1.4	Кратко описание на решението на този проблем	4
2	Анализ на решението.....	5
2.1	Работен процес	5
2.2	Структура на съдържанието.....	8
2.3	Тип и представяне на съдържанието.....	12
3	Дизайн	14
4	Тестване.....	23
5	Заклучение и възможно бъдещо развитие.....	26
6	Разпределение на работата	27
7	Използвани литературни източници и Уеб сайтове	28

1 Въведение

1.1 Избор на тема

Ние избрахме да създадем каталог на хотелите в света по региони (държави), базиран на XML документи с текстово и графично съдържание, описващо възможните характеристики на всеки един хотел в света спрямо различни предпочитания . За улеснение на потребителите най-важната информация и критерии за избор на хотел са събрани в един файл, вместо за всеки хотел да се проверява самостоятелно.







Чрез формулирането на темата по такъв начин, ние извършваме лека промяна в точно зададеното условие. То изисква хотелите да са само в рамките на България. Въпреки това, ние надграждаме това изискване, като освен хотели в България, представяме такива от целия свят.

1.2 Актуалност на избраната тема

В днешно време хотелиерството и туризма са едни от най-развитите сектори, защото голяма част от работните позиции предлагат така наречената „home office” работа, която позволява на работниците да работят от различни части на света. От друга страна пътуването в 21-ви век е много достъпно, това е причина да има различни типове търсене на хотели. Някои хора имат интерес към луксозния стандарт, други се интересуват от история и култура, трети предпочитат морски или зимен курорт и т.н.

1.3 Какъв точно проблем решава проекта и в какъв контекст е този проблем

Преглеждането на спецификациите и предимствата на всеки хотел по отделно, прави избора на хотел бавен процес. Намирането на най-съществената информация също не е лесно, когато всеки хотел се опитва да изтъкне екстрите, които предоставя. Избраната от нас тема има за цел да предостави сбито най-важната информация за всеки хотел, както следва:

-  Име;
-  Адрес, който в себе си съдържа информация за град и улица;
-  Контакти, които дават информация за телефон, факс, email, сайт на хотела за по-подробна информация и възможност за резервация, Facebook и Instagram;
-  Звезди;
-  Кратко описание, което дава представа за екстрите, които предоставя хотела;
-  Снимка.

1.4 Кратко описание на решението на този проблем

Каталогът представя графичното съдържание чрез XML Entities. Връзките в каталога – тези между региони и хотели, както и между вериги от хотели и хотел от веригата се описват чрез атрибути ID и IDREF.

За решението на този проблем, избрахме да представим 8 изцяло различни хотела, като за всеки има представено малко количество информация. Така избраните хотели ще покрият максимално всички възможни нужди и потребности на различните клиенти. Чрез този каталог, всеки клиент ще успее да открие типа хотел, който най-много се доближава до това, от което има нужда той в момента.

Каталогът е сравнително сбит – информацията за всеки хотел е представена в до една страница. В тази една страница за всеки от осемте хотела се включва неговото име, снимка, описание, място и основни контакти за бърза връзка. Така няма излишно инфо, което да проточи процеса на избиране на хотел и да отегчи четящия.

Цялата информация за хотелите, регионите и веригите е вписана в един основен .xml файл, с подходящи и прости имена на елементите (таговете) и атрибутите им, така че да става ясно само с четенето му каква функция изпълнява.

Създадохме подходящ DTD валидиращ документ, който е основа на .xml документа и го валидира. Накрая, създадохме и XSLT документ, който генерира PDF документ, където вече нагледно се представя цялото съществено съдържание, изцяло подходящо за четене от всеки потребител.

След края на техническата част, всичко е описано подробно в документация под формата на .docx файл. В следващите параграфи нагледно е описан всеки процес от решението на проблема, дизайна на крайния продукт, тестването му, възможни бъдещи подобрения, както и разпределението на работата по проекта.

2 Анализ на решението

2.1 Работен процес

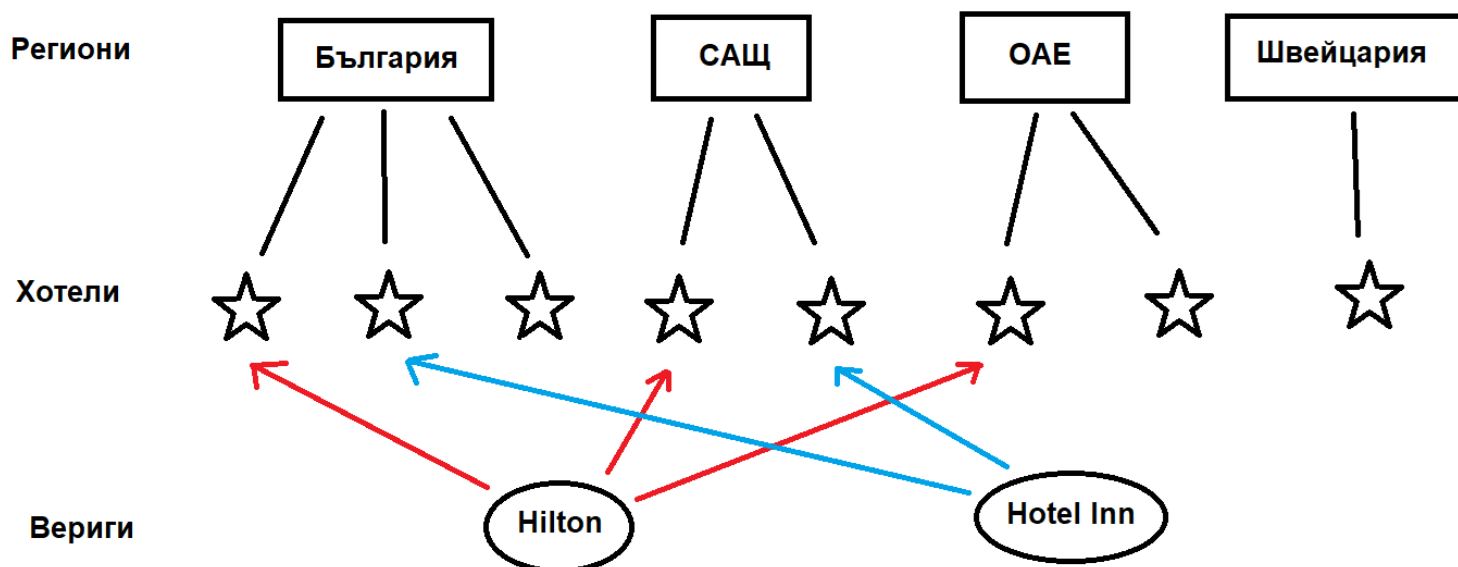
Спрямо изискването самият проект да включва региони, вериги от хотели и конкретни хотели. Също така, те трябва да бъдат свързани по между си чрез връзки. Регионите (които в нашия случай представляват държави) се свързват с конкретни хотели. От друга страна, част от хотелите са свързани с някоя от веригите. Всеки хотел е свързан към точно един регион, и може да бъде част от верига, което вече не е задължително. Регионите и веригите са свързани само косвено, чрез хотелите – те нямат директна връзка помежду си.

След анализиране на данните, преценихме за най-подходящо броят на отделните хотели, към момента, да бъде 8. Тези осем хотела да бъдат разпределени в 4 региона (държави) – България, САЩ, ОАЕ и Швейцария. Избрахме и две от най-известните вериги хотели в света (Хилтън и Хотел Инн).

Избрахме за най-удачно най-много хотели (три на брой) да бъдат разположени на територията на България. По два да са от САЩ и ОАЕ, и само един (планински) от Швейцария.

Веригата “Hilton”, в нашия каталог, е представявана от три хотела (в България, в САЩ и в ОАЕ). А другата верига – с два (в България и в САЩ). Останалите три от осемте хотела не са част от никоя верига.

За нагледно представяне на описаното и за по-лесна ориентация за реализирането на проекта и подбирането на конкретната информация, изработихме следната опростена схема:



Първо избрахме кои да са двете вериги, след което избрахме съответния брой хотели от тях и кои точно да са те. След избора на петте хотела от двете вериги, избрахме и останалите три самостоятелни, които да допълнят широката гама от разнообразни видове хотели.

След като вече знаем кои са осемте хотела, започнахме да набираме необходимата информация за всеки един. Текстовата нужна информация е следната: име, адрес (град, улица), контакти (телефон, факс, email, сайт, Facebook, Instagram), брой звезди и описание. След което започнахме да разглеждаме множество от снимки за всеки хотел, за да изберем най-подходящата, която ще пасва на целия облик на каталога и в същото време ще дава най-реална представа за хотела.

След като необходимата информация е изцяло събрана, започва същинската работа по проекта. Първото и най-основно нещо е да се изгради подходящ и добре-структуриран .xml документ. Този процес започва с изграждането на йерархичната структура от данни. Тя се състои от един елемент корен и няколко поделементи, които само по себе си могат да съдържат както много така и малко на брой свои поделементи. За всеки елемент се избира подходящо име, което едновременно не е твърде дълго и дава достатъчна информация, каква роля изпълнява. Почти всички елементи имат и атрибути (един или повече). Те, аналогично, имат кратки идентификатори и множество от стойности, които могат да притежават. Когато йерархията вече е напълно изградена, се въвеждат конкретните данни, които сме събрали, за хотелите, регионите и веригите.

Вторият основен документ е DTD файла. В него, всеки един елемент се декларира, че съществува по начина, по който съществува. Обозначават се и съответните му поделементи (само преки такива). Обозначаването става с точните имена на елементите. Декларират се също и точните атрибути, отново с конкретните им идентификатори, както и колко пъти да се срещат. Могат да бъдат задължителни или незадължителни при вписването на всеки един елемент. За поделементите на даден елемент, може да се обозначи, колко пъти да се съдържа, както следва:

? – за точно 0 или 1 срещане;

+ – за точно 1 или повече от едно срещания;

* – за 0 или повече срещания.

```
<!ELEMENT contacts (site, email+, phone+, fax*, facebook?, instagram?)>
<!ELEMENT site (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT fax (#PCDATA)>
<!ELEMENT facebook (#PCDATA)>
<!ELEMENT instagram (#PCDATA)>
```

Нотацията за ползване на изображения като външни единици (entities) е декларирана накрая на документа, и има по-различен синтаксис:

```
<!ELEMENT image EMPTY>  
<!ATTLIST image imgRef ENTITY #REQUIRED>  
<!NOTATION jpg PUBLIC "jpg">
```

Последният нужен файл за цялостното осъществяване на проекта и достигане до краен продукт (PDF файл) е .xsl файла. XSLT частта му е тази, която трансформира XML документа с помощта на различни стилове до HTML или до PDF. В нашия случай е нужен PDF и затова генерираме такъв. XSL-FO е език за описание на самото форматиране на данните в XML документа, за да бъдат подходящи за визуален изглед.

След като и трите необходими документа са изцяло завършени, използваме сайта www.xmlvalidation.com, за да валидираме нашия XML документ, по така дефинираното DTD и установяваме, че файлът е валиден. За изходен продукт, използваме приложението Altova XMLSpy 2023, с помощта на което от XML документа и XSL форматирането му, генерираме PDF файл, който представлява финалната версия на нашия каталог, която е предназначена за потребителите.

Крайният продукт – „Каталог за хотели по света“ е визуализиран в един PDF файл от 9 страници. Навигацията през него става чрез скролване надолу и нагоре. Започва с една заглавна страница, на която с текст е изписано какъв е следният каталог, съпроводен от илюстрираща образна снимка, както и имената на двамата създатели. Следващите 8 страници са напълно идентични като структура. Всяка от тях се състои от име на хотела, негова снимка, и пълната нужна информация за придобиване на прилична представа за съответния хотел (описана вече по-горе).

2.2 Структура на съдържанието

За реализирането на проекта използваме множество от елементи, групирани по подходящи начини, така че да се формира максимално подреден и логически свързан добре-структуриран и валиден XML документ. Някои елементи съдържат един или повече атрибути, чиято основна функция е да реферират други елементи, изграждайки връзки между двойки елементи от тип едно-към-едно/едно-към-много.

Структурата на крайния ни XML документ е следната:

Най-отгоре на текстовия файл се намира прологът:

```
<?xml version="1.0" encoding="utf-8"?>  
<?xml-stylesheet type="text/xsl" imgRef="XML2023_ProjectNo01.xsl"?>
```

Чрез него е деклариран самия xml документ, символното му кодиране – UTF-8, стилът и типа на документа, както и името на атрибута, който прави връзката между основния документ и XSL документа му (imgRef, тъй като се ползва за достъп и форматиране на външни изображения).

Следващия ключов текстов параграф е зададен със специфичен синтаксис. Ключовата дума е DOCTYPE, последвана от името на елемента-корен на йерархията, кл. дума SYSTEM и името на DTD документа в кавички. Чрез тази секция се прави връзката с .dtd файла, изграден, за да играе роля на „скелет“/„основа“ на главния документ и като цяло да осигурява валидността му. Вътре в тази секция, също така, са декларирани и 8 идентични „единици“, служещи за връзка с външни изображения. Синтаксисът за всяка отделна единица е:

```
<!ENTITY име_на_единица SYSTEM относителен_път_на_изображението NDATA jpg>
```

За по-добра читаемост всеки идентификатор на единица започва с „е_“. Релативният/относителният път служи за достъпване на изображението. За да може да се ползва относителен, вместо абсолютен път за извличане на снимките е нужно файловете да се разполагат в една и съща директория. Така е достатъчно да се зададат, разделени с „/“, единствено директорията със снимките и името на конкретната снимка с разширението й, вместо пълната й локация в системата. Ключовата дума NDATA е за единици, които не са XML данни (в случая изображения) и е обвързана с NOTATION от DTD документа, където от своя страна се намира декларацията на идентификаторът „jpg“.

Следва същинската структура на документа:

Коренът на йерархията е представен от елемента hotelCatalog, понеже целият проект има за цел да представи именно каталог на хотели.

hotelCatalog има три поделементи – regions, chains и hotels.


```

<hotelCatalog>
  <regions>
  </regions>

  <chains>
  </chains>

  <hotels>
  </hotels>
</hotelCatalog>

```

Тези три поделемента са елементи сами по себе си и са трите основни аспекта от цялата ни разработка.

Елементът `regions` съдържа в себе си само 4 поделемента, които споделят общо име – `region`. Те са напълно идентични, като идеята в нашия конкретен случай е че представляват 4 различни държави, в които се разполагат нашите хотели. Тези държави са: България, Съединените Американски Щати, Обединените Арабски Емирства и Швейцария. Имената на самите държави са изписани на кирилица като съдържание/стойност на самите елементи `region`. Освен това, елементите съдържат и по два задължителни атрибута – `regID` и `hotelRef`. Първият е собственият уникален идентификатор на елемента – възможни стойности: „BG“; „USA“; „UAE“; „CH“. Вторият атрибут играе роля на външен ключ към един или няколко от осемте хотела, изброени по-долу, имайки стойност самите ID-та на нужните хотели, изброени разделени с интервал.

```

<regions>
  <region regID="BG" hotelRef="HilSof InnBG Djog">
    България
  </region>
  <region regID="USA" hotelRef="InnUSA Waikiki">
    Съединени Американски Щати
  </region>
  <region regID="UAE" hotelRef="AtlDubai Jumeirah">
    Обединени Арабски Емирства
  </region>
  <region regID="CH" hotelRef="Omnia">
    Швейцария
  </region>
</regions>

```

Вторият главен поделемент на корена е елементът – `chains`. Той, аналогично на `regions`, съдържа два поделемента с еднакви имена – `chain`. Те илюстрират двете вериги хотели, които избрахме за нашия проект – “Hilton” и “Hotel Inn”. Отново, самите имена на веригите са съдържанието на съответния елемент, от който са представлявани. И двете

вериги имат по два атрибута – chainID и вече познатият ни hotelRef. chainID е от тип ID и е кратък идентификатор на самата верика, докато hotelRef е от тип IDREF и реферира някой/някои hotel елементи (като външен ключ).

```
<chains>
  <chain chainID="Hil" hotelRef="HilSof Waikiki Jumeirah">
    Hilton
  </chain>
  <chain chainID="Inn" hotelRef="InnUSA InnBG">
    Hotel Inn
  </chain>
</chains>
```

Третият и най-основен градивен елемент е hotels. Той съдържа 8 отделни hotel елемента – всеки един представляващ точно един хотел.

hotel елемент съдържа два или три атрибута.

- hotelID – стойността му е уникален идентификатор (първичен ключ) за съответния хотел.
- regRef – стойността му е референция към някой от 4-те региона, т.е. стойността му е идентификатора на някой от 4-те държави. Показва хотелът в коя държава се намира.
- chainRef – стойността му е препратка към някоя от 2-те вериги от хотели, т.е. стойността му е идентификатора на някоя от 2-те вериги. Този атрибут е опционален! Тоест той може да не съществува за даден елемент hotel. Липсата му ще показва, че съответният хотел не принадлежи към никоя верига.

Чрез така описаните атрибути става ясно, че даден хотел задължително има свой собствен идентификатор. Задължително е и от някоя от 4-те държави. Това което не е задължително е да е част от верига хотели.

```
<hotel hotelID="HilSof" regRef="BG" chainRef="Hil">
</hotel>
<hotel hotelID="AtlDubai" regRef="UAE">
</hotel>
<hotel hotelID="InnUSA" regRef="USA" chainRef="Inn">
</hotel>
```

Последното и най-задълбочено описание е на вътрешността на един hotel елемент.

Поделементите на hotel са:

- name – съдържанието му е точното наименование на хотела.
- address – адресът на хотела. Пояснява се от два поделемента: city, street, чийто съдържание са съответно градът, в който се намира и улицата, на която е разположен.
- contacts – структура, събираща в себе си 6 характеристики, които в съвкупност представляват начините за връзка с този хотел, а именно неговите „контакти“. Те са: site, email, phone, fax, Facebook, Instagram. Само първите три от тях са задължителни да присъстват за всеки един хотел (уебсайт, имейл и телефон). Няма как хотел с липсваща/и характеристика от тези три основни да бъде вписан в нашия каталог. По-специфично – сайтът трябва да е точно един, докато имейлите и телефоните за връзка могат да бъдат повече от един. Останалите три поделемента (факс, Фейсбук и Инстаграм) остават незадължителни за вписване. Тоест за някои хотели те няма да фигурират и съответно не са валиден инструмент за връзка, но по наша преценка те са от второстепенно значение и затова хотел не е задължен да ги предоставя.
- stars – стойността на елемента stars показва колко звезди притежава хотела.
- description – елемент със съдържание от тип PCDATA, целящ да събере в себе си цялата нужна информация относно „описанието“ на даден хотел. При някои то е по-кратко и информативно, при други е дълго и подробно, онагледяващо всички основни аспекти на хотела. Дългите описание са със специално намален шрифт от стандартния (14pt) във финалната визуализация с цел всичко за хотела да бъде събрано в една страница.
- image – самозатварящ се таг, който върши ролята да извлече нужната снимка за хотела. Няма стойност. Съдържа единствен задължителен атрибут imgRef, чиято стойност е точния идентификатор на съответното entity, отговарящо за дадената снимка.

```
<hotel hotelID="Omnia" regRef="CH">
  <name>
    The Omnia Hotel - Zermatt
  </name>
  <address>
    <city>Zermatt</city>
    <street>Auf dem Fels, 3920</street>
  </address>
  <contacts>
    <site>www.the-omnia.com</site>
    <email>info@the-omnia.com</email>
    <phone>+41 27 966 71 71</phone>
    <fax>+41 27 966 71 00</fax>
  </contacts>
</hotel>
```

```

        <facebook>THE OMNIA</facebook>
        <instagram>@omnia.zermatt</instagram>
    </contacts>
    <stars>5-star</stars>
    <description>
        Повечето стаи осигуряват изглед навън, благодарение на прилежащия
        балкон. Стаите притежават мини-хладилник, сейф и безжичен интернет. Хотелът
        предлага фамилна стая и стаи за непущачи. Гостите ще се почувстват комфортно във
        всяка една от наличните 30 стаи. Удобствата включват ресторант, зала за хранене,
        бар, обслужване по стаите и конферентна зала.
    </description>
    <image imgRef="e_omnia"/>
</hotel>

```

2.3 Тип и представяне на съдържанието

За цялостно оформяне на проекта сме използвали следните 6 файла:

„XML2023_ProjectNo01.xml“

„XML2023_ProjectNo01.dtd“

„XML2023_ProjectNo01.xsl“

„XML2023_ProjectNo01.pdf“

„XML2023_ProjectNo01.doc“

Най-общо казано, .xml файла ни служи за въвеждане и структуриране на информацията, .dtd го валидира и потвърждава, че вписаната информация е коректна. XSL файла форматира файла до pdf формат, който е и желаният изходен файл за клиента. В .doc файла подробно е описано предназначението на проекта, както и подробна информация за етапа на разработката му.

В следващата точка подробно е описано функцията и изгледа на всеки един от използваните файлове.

За по-добра ориентация в крайния продукт, използвахме и 9 графични файла, както следва:

За първата и начална страница използвахме файлът „headerHotel.jpg“, който е с размер 66511 bytes. Източникът му е: depositphotos.com

За следващите 8 страници, използвахме следните 8 графични файла, във формат JPG:

- „waikiki.jpg“, с размер 78 264 bytes и източник: www.costcotravel.com
- „omnia.jpg“, с размер 500 153 bytes и източник: www.usm.com
- „jumeir.jpg“, с размер 110 994 bytes и източник: www.theprocchefme.com
- „innUSA.jpg“, с размер 106 998 bytes и източник: www.weddingwire.com
- „innBG.jpg“, с размер 81 593 bytes и източник: rad-festa.com
- „hiltonSofia.jpg“, с размер 89 130 bytes и източник: galaxy-bg.com
- „hDjo.jpg“, с размер 649 243 bytes и източник: grupovo.bg
- „atlDubai.jpg“, с размер 58 475 bytes и източник: www.traveldailymedia.com

Общият размер на всички графични файлове е 1,66 MB, като за перфектно визуализиране в крайният pdf файл, всички изображения индивидуално са преформатирани.

3 Дизайн

За цялостното решение на формулирания проблем използваме 3 базови файла и още един финален, който е автоматично генериран продукт от тях. Събират се и общо 9 изображения (1 заглавно и 8 за конкретни хотели), които се съхраняват в една директория (поддиректория за директорията на целия проект) и се достъпват като външни единици, за да може във финалния продукт да има и визуално репрезентиране на всичко нужно. Накрая, след като цялостния процес е финализиран и изпълнен, пишем дълга и подробна документация, където всичко извършено за целия проект, бива описано с текст и онагледено със схеми, диаграми, както и екранни заснемания на файловете и финалния продукт.

Използваните технически файлове са:

- XML
- DTD
- XSL

XML файла е най-основният за цялата разработка. В секция 2.2 Структура на съдържанието е описан максимално подробно. Всичко свързано с него, кое каква функция изпълнява и защо логически е такова каквото е.

Вторият изискван и създаден файл е DTD файла. Той се ползва основно за валидация на основния файл. В него под формата на „правила“ са записани всички елементи с точните им имена, брой поделементи (вкл. по колко пъти е разрешения брой срещания), атрибути (вкл. имената им, типа им, и дали са задължителен компонент или не).

DTD документът има следното съдържание:

```
<!ELEMENT hotelCatalog (regions, chains, hotels)>

<!ELEMENT regions (region+)>
<!ELEMENT region (#PCDATA)>
<!ATTLIST region regID ID #REQUIRED>
<!ATTLIST region hotelRef IDREFS #REQUIRED>

<!ELEMENT chains (chain+)>
<!ELEMENT chain (#PCDATA)>
<!ATTLIST chain chainID ID #REQUIRED>
<!ATTLIST chain hotelRef IDREFS #REQUIRED>

<!ELEMENT hotels (hotel+)>
<!ELEMENT hotel (name, address, contacts, stars, description, image)>
```

```

<!ATTLIST hotel hotelID ID #REQUIRED>
<!ATTLIST hotel regRef IDREFS #REQUIRED>
<!ATTLIST hotel chainRef IDREFS #IMPLIED>

<!ELEMENT name (#PCDATA)>

<!ELEMENT address (city, street)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT street (#PCDATA)>

<!ELEMENT contacts (site, email+, phone+, fax*, facebook?, instagram?)>
<!ELEMENT site (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT fax (#PCDATA)>
<!ELEMENT facebook (#PCDATA)>
<!ELEMENT instagram (#PCDATA)>

<!ELEMENT stars (#PCDATA)>

<!ELEMENT description (#PCDATA)>

<!ELEMENT image EMPTY>
<!ATTLIST image imgRef ENTITY #REQUIRED>
<!NOTATION jpeg PUBLIC "jpeg">

```

Всичко описано в секция 2.2 за основния XML документ, тук се онагледява чрез DTD документа.

Елементът-корен hotelCatalog съдържа три поделементи – regions, chains, hotels.

Елементът regions съдържа поне един подеlement region.

Елементът region има съдържание #PCDATA.

Атрибутите на region са: regID от тип ID (задължителен) и hotelRef от тип IDREF (задължителен).

Елементът chains съдържа поне един подеlement chain.

Елементът chain има стойност от тип #PCDATA.

Атрибутите на chain са: chainID от тип ID (задължителен) и hotelRef от тип IDREF (задължителен).

Елементът hotels има поне един елемент hotel.

Елементът hotel има 6 поделементи – name, address, contacts, stars, description, image.

Атрибутите на hotel са hotelID от тип ID (задължителен), regRef от тип IDREF (задължителен) и chainRef от тип IDREF (незадължителен). По-горе в секция 2.2 е описана разликата защо, според заданието, някои са задължителни, а други – не.

Елементът name е със съдържание от тип #PCDATA.

Елементът address има два задължителни поделементи – city, street.

Елементът city има #PCDATA съдържание.

Елементът street има #PCDATA съдържание.

Елементът contacts има 6 поделементи – site, email, phone, fax, facebook, instagram. Първите три, от които са задължителни, а последните три – не.

Елементът site има стойност от тип #PCDATA.

Елементът email има стойност от тип #PCDATA.

Елементът phone има стойност от тип #PCDATA.

Елементът fax има стойност от тип #PCDATA.

Елементът facebook има стойност от тип #PCDATA.

Елементът instagram има стойност от тип #PCDATA.

Елементът stars има стойност от тип #PCDATA.

Елементът description има стойност от тип #PCDATA.

Елементът image е празен – няма съдържание/стойност.

Единственият атрибут на image е: imgRef от тип ENTITY и е задължителен.

Единственото не-XML съдържание в цялата разработка са изображенията. Дефинирани от думата NOTATION и идентификаторът jpg. Единствената стойност, която могат да приемат е „jpg“ (такова разширение на файлове).

Чрез уебсайта www.xmlvalidation.com, валидирахме завършения XML файл с така изготвения DTD. Валидацията е успешна, на екрана се изписа съобщението “No errors were found”.

Третият нужен документ за цялостния облик на каталога е от тип XSL. Той се ползва с цел да преобразува XML в специфичен формат за представяне/визуализиране, в случая PDF. Преобразува данните от формат, разбираем за едно приложение, във формат, разбираем за друго приложение.

Заглавния облик на нашия .xsl файл е следният:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:output method="xml" version="1.0" indent="yes"/>
  <xsl:template match="/">
    <fo:root>
    </fo:root>
  </xsl:template>
</xsl:stylesheet>
```

`<xsl:stylesheet>` задава обхват на XSLT документа и осигурява конфигуриране на различни параметри.

`<xsl:output>` се разполага веднага след `stylesheet` и позволява контрол над изходния документ (в нашия случай основния .xml файл).

`<xsl:template>` специфицира кои елементи трябва да бъдат обхванати от обработката, както и дефинира какво ще бъде добавено в изходния документ. Изпълнява кода в шаблона всеки път, когато срещне елемент, който съответства на специфицирания в атрибута `match` (в случая `match = "/"`). Шаблонът се изпълнява, когато се срещне кореновият елемент на документа.

`<fo:root>` вече е кореновия елемент за XSL документа от XSL-FO (Formatting Objects) частта.

Опростеният изглед на елементът-корен и неговите поделементи е следният:

```
<fo:root>
  <fo:layout-master-set>
    <fo:simple-page-master page-height="297mm" page-width="210mm"
margin="5mm 25mm 5mm 25mm" master-name="template">
      <fo:region-body margin="20mm 0mm 20mm 0mm"/>
    </fo:simple-page-master>
  </fo:layout-master-set>

  <fo:page-sequence master-reference="template">
  </fo:page-sequence>

  <!-- 1-st -->
```

```

    <fo:page-sequence master-reference="template">
    </fo:page-sequence>
    <!-- 1-st -->

    <!-- 2-nd -->
    <fo:page-sequence master-reference="template">
    </fo:page-sequence>
    <!-- 2-nd -->

    И АНАЛОГИЧНО ЗА ОЩЕ 6 СТРАНИЦИ (page-sequence - a) НАДОЛУ. ОБЩО 8 ХОТЕЛА.
    .....
    .....

    <!-- 8-th -->
  </fo:root>
</xsl:template>
</xsl:stylesheet>

```

Тук първо се дефинира `<fo:layout-master-set>`, в който се конфигурират характеристиките на документа като цяло. Това става, чрез промяната на стойностите на дадени параметри в точно числа и последвани от мерна единица.

След това единично се дефинира заглавната страница `<fo:page-sequence>`:

```

<fo:page-sequence master-reference="template">
  <fo:flow flow-name="xsl-region-body">
    <fo:block position="absolute" text-align="center" font-weight="bold"
margin-bottom="50mm" font-family="Calibri" font-size="48pt" padding-before="25mm"
color="black">
      КАТАЛОГ НА ХОТЕЛИТЕ В СВЕТА
      <fo:external-graphic src="url('hotelPictures/headerHotel.jpg')"
text-align="center" padding-after="0mm" margin-bottom="10mm" content-
height="400" content-width="480" >
    </fo:external-graphic>
    </fo:block>
    <fo:block position="absolute" text-align="right" font-style="italic"
margin-bottom="0mm" font-family="Calibri" font-size="16pt" padding-before="0mm"
color="black">
      Петя Петрова - 62630, СИ
    </fo:block>
    <fo:block position="absolute" text-align="right" font-style="italic"
margin-bottom="0mm" font-family="Calibri" font-size="16pt" padding-before="0mm"
color="black">
      Радослав Велков - 62528, СИ
    </fo:block>
  </fo:flow>

```

```
</fo:page-sequence>
```

Основните блокове тук са 3. Първият ни изписва заглавието на целия каталог – „КАТАЛОГ НА ХОТЕЛИТЕ В СВЕТА“ и веднага отдолу изобразява, чрез external-graphic външно изображение (не под формата на единица за проекта като останалите) една заглавна снимка с относителен път – hotelPictures/headerHotel.jpg. Следват още два идентични блока – единият за името, факултетния номер и специалността на единия автор и другият – за другия автор.

В останалата най-дълга част от XSL файла и по-конкретно от йерархията на root елемента са разположени 8 аналогични блока, всеки от които отговаря за точно една страница от каталога.

Ще разгледаме как изглежда един page-sequence:

```
<!-- 4-th -->
<fo:page-sequence master-reference="template">
  <fo:flow flow-name="xsl-region-body">
    <fo:block-container position="absolute" top="-2.5cm" left="-2.5cm">
      <fo:block position="absolute" text-align="center" padding-before="2cm" margin-left="2.5cm"
        font-family="Calibri" font-size="40pt" color="black">
        <xsl:value-of select="//hotel[@hotelID='Omnia']/name"/>
      </fo:block>

      <fo:block position="absolute" text-align="center" padding-before="7mm" margin-left="2.5cm">
        <fo:external-graphic src="{unparsed-entity-uri(//hotel[@hotelID='Omnia']/image/@imgRef)}" >
        </fo:external-graphic>
      </fo:block>

      <fo:block position="relative" font-family="Calibri" font-weight="bold" font-size="15pt"
        color="black" padding-before="5mm" margin-left="2cm">
        Адрес:
        <fo:inline position="relative" padding-left="31mm" font-weight="normal" font-size="14pt">
          <xsl:value-of select="//hotel[@hotelID='Omnia']/address"/>
        </fo:inline>
      </fo:block>

      <fo:block position="relative" font-family="Calibri" font-weight="bold" font-size="15pt"
        color="black" padding-before="5mm" margin-left="2cm">
        Сайт:
        <fo:inline position="relative" padding-left="36mm" font-weight="normal" font-size="14pt">
          <xsl:value-of select="//hotel[@hotelID='Omnia']/contacts/site"/>
        </fo:inline>
      </fo:block/>

      Имейл:
```

```

<fo:inline position="relative" padding-left="31mm" font-weight="normal" font-size="14pt">
  <xsl:value-of select="//hotel[@hotelID='Omnia']/contacts/email"/>
</fo:inline>
<fo:block/>

Телефон:
<fo:inline position="relative" padding-left="27mm" font-weight="normal" font-size="14pt">
  <xsl:value-of select="//hotel[@hotelID='Omnia']/contacts/phone"/>
</fo:inline>
<fo:block/>

Факс:
<fo:inline position="relative" padding-left="35mm" font-weight="normal" font-size="14pt">
  <xsl:value-of select="//hotel[@hotelID='Omnia']/contacts/fax"/>
</fo:inline>
<fo:block/>

Фейсбук:
<fo:inline position="relative" padding-left="27mm" font-weight="normal" font-size="14pt">
  <xsl:value-of select="//hotel[@hotelID='Omnia']/contacts/facebook"/>
</fo:inline>
<fo:block/>

Инстаграм:
<fo:inline position="relative" padding-left="22mm" font-weight="normal" font-size="14pt">
  <xsl:value-of select="//hotel[@hotelID='Omnia']/contacts/instagram"/>
</fo:inline>
<fo:block/>
</fo:block>

<fo:block position="relative" font-family="Calibri" font-weight="bold" font-size="15pt"
color="black" padding-before="5mm" margin-left="2cm">
  Звезды:
  <fo:inline position="relative" padding-left="31mm" font-weight="normal" font-size="14pt">
    <xsl:value-of select="//hotel[@hotelID='Omnia']/stars"/>
  </fo:inline>
</fo:block>

<fo:block position="relative" font-family="Calibri" font-weight="bold" font-size="15pt"
color="black" padding-before="5mm" margin-left="2cm">
  Описание:
  <fo:block/>
  <fo:inline position="relative" padding-left="10mm" font-weight="normal" font-size="14pt">
    <xsl:value-of select="//hotel[@hotelID='Omnia']/description"/>
  </fo:inline>
</fo:block>

```

```

        </fo:block-container>

    </fo:flow>

</fo:page-sequence>


<!-- 4-th -->

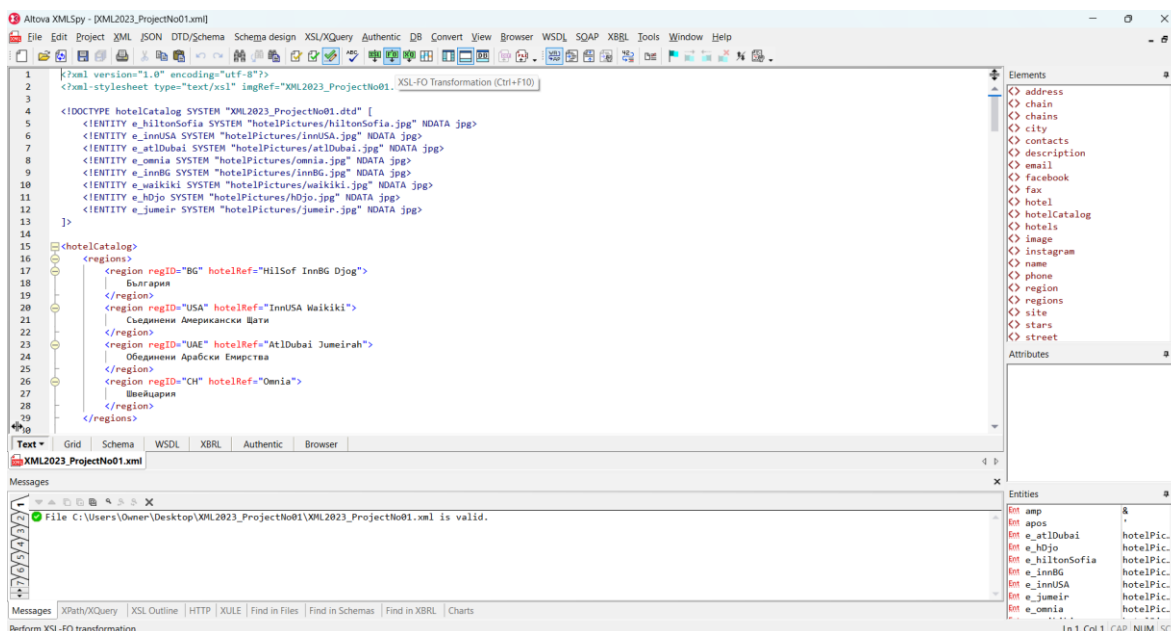
```


Като цяло, всяка от 8-те страници е изградена от множество отделни блока. Всеки има собствено форматиране, според точната дължина/обем на текста. За описанието (description) на хотел, например, ако целият текст е твърде дълъг и излиза от страницата, ръчно му променяме размера на шрифта на по-малък, така че да се вмести. Ако пък обратното – твърде кратък е, го уголемяваме, за да не се получава твърде голямо бяло/празно пространство до края на страницата. Друг пример е отстоянието на съдържанието на конкретен елемент от името му (което се явява и име на блока). Например, когато линкът за уебсайта на хотела е по-дълъг от обичайното, се измества това отстояние наляво, за да може този дълъг линк да се събере целият на един ред, а не да преминава на следващ и да нарушава цялата структура на документа надолу.

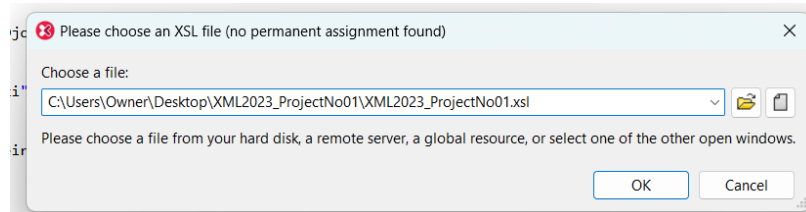
Конкретен хотел се достъпва като се вземе единият елемент от 8-те с име hotel, чието съдържание на hotelID е равно на това, на който хотел търсим. Например, искаме да достъпим съдържанието на хотела с hotelID = „Omnia“. По-конкретно от съдържанието му искаме описанието му. Това става със следния синтаксис: `//hotel[@hotelID='Omnia']/description`.

Всичките елементи, атрибути и техни стойности, които са вписани и ползвани в XSL документа имат за цел да форматира крайния продукт, който трябва да се получи от основния XML файл, моделирайки изгледа по търсения от нас начин за финалния резултат (каталог за хотели), който да се показва на клиента под формата на един PDF файл.

След като всички основни технически документи са завършени. Използваме приложението Altova XMLSpy 2023 . В него се отваря .xml файлът и изгледът е следният:



Чрез натискане на бутона XSL-FO Transformation  в секция XSL/XQuery и следване на стъпките, избирайки да се генерира PDF и задавайки абсолютния път на .xsl файла, чрез който да се генерира:



Се генерира PDF файл в избрана от нас директория, който в нашия случай представлява финалния продукт на целия проект – Каталогът на хотелите по света и изглежда по следния начин:

Заглавна страница:



Страница на един от 8-те хотела:



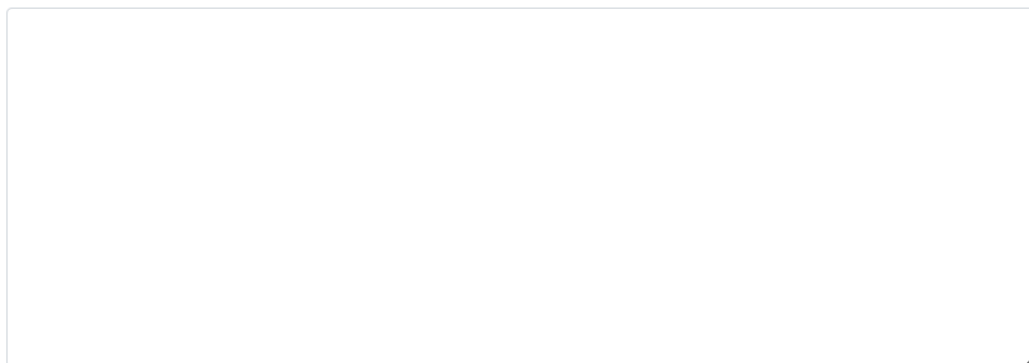
4 Тестване

Тъй като проектът съдържа множество компоненти и има определени изисквания за входните данни от голямо значение е валидацията на входни данни да е коректна.

Първият тест беше dtd валидацията, която ни даде информация, че проектът е валиден и добре форматиран. Тя се осъществи чрез .xml файла и .dtd файла, които стартирахме в сайтът <https://www.xmlvalidation.com>.

Първо стартирахме сайта, след което добавихме .xml файла, който искахме да валидираме.

Please copy your XML document in here:



Or upload it:

Избор на файл XML2023_...tNo01.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

Има възможност и за копиране на код, вместо прикачване на целия файл. Ние като качествени програмисти избрахме да не копираме съдържанието на файла, защото знаем че копирането на код не е добра практика, поради фактът, че е много лесно да се пропусне копирането на определен символ, който в последствие ще даде грешка в самата валидация.

Следващата стъпка беше да добавим валидиращия .dtd файл по същия начин.

The file XML2023_ProjectNo01.dtd is being referenced. Please copy it in here, so that the validation can continue:

Or upload it:

Избор на файл XML2023_...tNo01.dtd

continue validation

Финалният резултат, без грешки, който ни удовлетворяваше от първото тестване е следният:


No errors were found

The following files have been uploaded so far:

[XML document:](#) 

[XML2023_ProjectNo01.dtd](#) 

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

Крайното представяне на каталога е в PDF формат, който генерирахме чрез системата Altova XMLSpy 2023  и примерен изглед на крайния продукт от последната стъпка на тестване е следният:

КАТАЛОГ НА ХОТЕЛИТЕ В СВЕТА



Петя Петрова - 62630, СИ
Радослав Велков - 62528, СИ

5 Заключение и възможно бъдещо развитие

Проектът има изключително широка практичност за всички, които имат интерес към хотелиерството, както и за всички, които търсят бърз и лесен начин за намиране на перфектния хотел спрямо индивидуалните им изисквания и очаквания, защото целта е сбито и в един файл да се представи информация за различни хотели, където да има избор за всяко предпочитание.

Чрез XML езика създадохме един добре структуриран файл, с добре структурирана йерархия между различните елементи и също така ясно представяне на връзките помежду им. Чрез XSLT езика се преобразува и форматира информацията в лесно четим, за всеки потребител, PDF файл.

Поради подредената йерархия между различните елементи и разделянето на информацията в повече файлове, лесно може да се добави нова информация, която да разшири текущия проект.

Въвеждането на символа ,☆' за илюстриране на броя звезди за всеки от хотелите, ще бъде подобрение, което ще дава по-приятна визуализация на крайния файл, предназначен за потребителя. Това на този етап не успяхме да го реализираме, поради ограничения синтаксис на XML.

Друго възможно подобрение е да се добави по повече от една снимка за всеки един хотел, за придобиване на по-ясна представа за всеки от хотелите. Например може да се добавят няколко снимки, които илюстрират интериора на хотела, различните екстри, които се предлагат, ако има включено хранене в нощувката, да се добави снимка на примерна закуска и пр.

Добавянето на примерни цени за всеки от хотелите е възможно подобрение, чрез което различните потребители могат да се ориентират в избора си на хотел спрямо бюджета, с който разполагат.

Друго подобрение с цел бъдещо развитие на проекта е това да се добавят още хотели.

Изброените по-горе възможности като подобрения на проекта в себе си носят и недостатъкът от претрупване с информация на крайния продукт. За да се избегне това трябва да се подбере и представи отново най-сбито всяко предложено подобрение.

6 Разпределение на работата

От започването до самия му край, проектът беше разработван от двама души. Почти всеки компонент беше изграден взаимно и от двамата автора, работейки заедно и едновременно, и съгласувайки всяко нещо помежду си. Въпреки това, при някои компоненти преобладават повече идеите и личната работа на единия, а при други – на другия. Следващата таблица показва нагледно точното разпределение на работата по целия проект. Нека за улеснение наричаме двамата автори с техните първи букви от малките им имена, съответно „П“ за Петя Петрова и „Р“ за Радослав Велков.

ЧАСТ ОТ ПРОЕКТА	РАЗРАБОТВАЩ/И
НАЧАЛЕН АНАЛИЗ; ОПРЕДЕЛЯНЕ НА НЕОБХОДИМА ИНФОРМАЦИЯ; ИЗГОТВЯНЕ НА ПЛАН ЗА РАБОТА	П / Р
СЪБИРАНЕ НА ИНФОРМАЦИЯ ЗА ХОТЕЛИ И НАБАВЯНЕ НА ИЗОБРАЖЕНИЯ	П
ИЗГОТВЯНЕ НА .XML ФАЙЛ	П / Р
ИЗГОТВЯНЕ НА .DTD ФАЙЛ	П / Р
ИЗГОТВЯНЕ НА .XSL ФАЙЛ	Р
ГЕНЕРИРАНЕ НА .PDF ФАЙЛ	Р
ОПИСАНИЕ В ДОКУМЕНТАЦИЯТА:	П / Р
- ВЪВЕДЕНИЕ	П
- РАБОТЕН ПРОЦЕС	П / Р
- СТРУКТУРА НА СЪДЪРЖАНИЕТО	Р
- ТИП И ПРЕДСТАВЯНЕ НА СЪДЪРЖАНИЕТО	П
- ДИЗАЙН	Р
- ТЕСТВАНЕ	П
- ЗАКЛЮЧЕНИЕ	П
- РАЗПРЕДЕЛЕНИЕ НА РАБОТАТА	Р
- ИЗПОЛЗВАНИ ИЗТОЧНИЦИ	П / Р

7 Използвани литературни източници и Уеб сайтове

1. <https://learn.fmi.uni-sofia.bg/course/view.php?id=8554>
2. <https://www.google.com/>
3. <https://www.altova.com/xmlspy-xml-editor/download>
4. www.palmjumeirah.hilton.com
5. <https://www.facebook.com/HiltonDubai>
6. <https://www.instagram.com/hiltondubaipalm/?hl=bg>
7. <https://www.instagram.com/hiltonhawaiianvillage>
8. <https://www.facebook.com/hiltonsofia>
9. https://www.hiltonhotels.com/bg_BG/bulgarija/hilton-sofia
10. <https://www.instagram.com/hilton.sofia/?hl=bg>
11. <https://www.facebook.com/Hlcelebration>
12. <https://www.instagram.com/explore/locations/4506796>
13. www.ihg.com/holidayinn/hotels/us/en/kissimmee/mcoib/hoteldetail
14. <https://www.facebook.com/HolidayInnPlovdiv>
15. <https://www.instagram.com/holidayinnplovdiv>
16. <http://hiplovdivhotel.com>
17. <https://www.facebook.com/atlantisthepalm>
18. <https://www.instagram.com/atlantisthepalm/?hl=bg>
19. <https://www.atlantis.com/atlantis-the-palm>
20. <https://www.facebook.com/Omnia.Zermatt>
21. <https://www.instagram.com/omnia.zermatt/?hl=bg>
22. <https://www.the-omnia.com/en/hotel>
23. <https://www.gogolanovahouse.com/>
24. <https://www.facebook.com/gunatrendafil55>
25. <https://learn.fmi.uni-sofia.bg/course/view.php?id=8554>
26. <https://en.wikipedia.org/wiki/XML>
27. https://en.wikipedia.org/wiki/Document_type_definition
28. https://www.w3schools.com/xml/xml_dtd_intro.asp
29. https://www.w3schools.com/xml/xsl_intro.asp
30. <https://en.wikipedia.org/wiki/XSL>
31. <https://www.xmlvalidation.com>
32. <https://www.google.com>

33. <https://www.altova.com/xmlspy-xml-editor/download>
34. <https://galaxy-bg.com>
35. <https://grupovo.bg>
36. <https://www.traveldailymedia.com>
37. <https://www.xmlvalidation.com>
38. <https://www.hilton.com/en/hotels/hnlhvhh-hilton-hawaiian-village-waikiki-beach-resort>
39. <https://www.facebook.com/HyattRegencyWaikikiBeach>
40. <https://luxurycolumnist.com/the-most-luxurious-hotels-in-the-world>
41. <https://theluxurytravelexpert.com/2020/12/14/best-hotels-in-the-world>
42. www.costcotravel.com
43. www.usm.com
44. www.theprocchefme.com
45. www.weddingwire.com
46. <https://rad-festa.com/galaxy-bg.com>
47. www.traveldailymedia.com