



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 №013: „Сборник с готварски рецепти - 1“

Курсов проект

Автори:

Илиян Йорданов, фак. номер 62546

Александра Радева, фак. номер 62541

януари, 2023 г.

София

Съдържание

1	Въведение	3
2	Анализ на решението	3
2.1	Работен процес	3
2.2	Структура на съдържанието	4
2.3	Тип и представяне на съдържанието	6
3	Дизайн	9
4	Тестване	24
5	Заклучение и възможно бъдещо развитие	28
6	Разпределение на работата	28
7	Използвани литературни източници и Уеб сайтове	29

1 Въведение

Както сигурно много хора са запознати, кулинарията е едно от любимите хобита или работа на редица българи. Понякога търсенето на рецепти става сложно, поради незнание на името на самата или пък търсенето измежду множество сайтове, съдържащи различни информации за дадени рецепти, които могат да доведат до объркване или грешна направа на ястието, или пък повечето не са склонни да си закупят книгата на някой шеф-готвач поради прекалено високите цени. Целта на настоящия документ е да представи проект на тема „Сборник от български рецепти, разпределени по региони“, като той съдържа любими на много българи традиционни рецепти и цели да запознае читателя с едни от характерните ястия за всеки регион, както и да му помогне за направата на съответните ястия.

Представянето на каталога с тези рецепти става с помощта на XML документ, съдържащ информация за всяка една от изброените от нас рецепти. Чрез атрибути ID/IDREF са реализирани връзките между отделните региони и рецепти, както и между рецепти и използваните за тях техники. Проектираният сборник представя информация за, към момента, десет български рецепти, описвайки техния произход, съставки, начин на приготвяне, препоръки за сервиране, медицински параметри, алергени и други. За да сме сигурни, че дадения XML документът валиден, ние сме решили да го валидираме посредством създаден за него DTD документ, който подsigурява ще няма да възникнат грешки при добавяне на още рецепти или пък модифициране на текущите, а пък самото съдържанието е представено в PDF документ посредством XSLT.

В съдържанието на настоящия документ е описан всеки съществен компонент на проекта, като в детайлен прочит може да бъде разгледан подробен анализ на решението на възложеното задание, като е описан дизайнът и подхода на нашето решение. Освен това дадения документ включва насоки за бъдещо развитие и изброени източници, по които екипът се водеше по време на разработването на проекта.

2 Анализ на решението

2.1 Работен процес

За да може дадения сборник да може да бъде създаден, както и да бъдат добавяни към него елементи, ние трябва да можем да се сдобием от някъде с нужната информация. В рамките на текущия проект, информацията, с която сме се сдобили идва от специално селектирани интернет сайтове, които са описани в точка [7 Използвани литературни източници и Уеб сайтове](#). След детайлен прочит на всеки един от сайтовете ние избираме какво е важно и какво не е, за да можем да го включим в нашия документ, съответно от вече тази подбрана информация ние допълнително филтрираме съдържанието като извлечем данните за автора на рецептата, нейния произход, съставките нужни за направата и, стъпките за приготвяне на рецептата, как се сервира самото ястие, с какви напитки се сервира, медицински параметри на ястието и алергени, за да може да бъде добре изграден документа и да добие една цялостност.

След като имаме готово съдържание за всяка една от рецептите, които предварително сме направили проучване, за кои са от най-характерните за дадения регион, ние трябва да намерим и подходящи снимки за всяка една от тях, като те трябва да бъдат с достатъчно високо качество, за да могат после хората, които гледат този каталог да могат да добият добра представа как трябва накрая да се получи съответното ястие. Съответно тези снимки после се съхраняват в отделна папка в папката на дадения проект.

След като сме избрали тези снимки какви ще са, имаме и нужната информация за рецептите, ние вече можем да започнем изграждането на самия XML документ, като той трябва да бъде добре структуриран за да може следващата обработка на съответните данни да бъде лесна и, а и, естествено, да бъде добре направен. За да се подsigури неговата валидност на документа ние сме направили специален DTD файл, който го подsigурява това, а графичното представяне на рецептите (снимките) се случва чрез XML entities.

За самото създаване на pdf файла и неговата визуализация ние постигаме това с помощта на XSLT трансформация през програмата Altova XMLSpy 2023. Тъй като ние изготвяме каталог с рецепти идващи от различни краища на България, ние сме сметнали за нужно те да бъдат разделени в 7 на брой региони – северозападен, северен централен, североизточен, югозападен, южен централен, югоизточен и незнаен (този регион се отнася за рецепти, чиито регион е неизвестен, но се предават от уста на уста).

2.2 Структура на съдържанието

В тази точка ще опишем подробно структурата на съдържанието на XML документа.

1. **catalog** – елемент корен на XML документа. Съдържащите се в него елементи са *regions*, *techniques-to-prepare* и *recipes*.

1.1. **regions** – елемент наследник на *catalog*.

- 1.1.1. **region** – елемент наследник на *regions*. Елементът има два задължителни атрибута *name*(тип CDATA) и *id*(тип ID).

- 1.1.1.1. **name** – атрибут на елемента *region* от тип CDATA. Носи информация за името на даден български регион.

- 1.1.1.2. **id** – атрибут на елемента *region* от тип ID. Носи информация за инициалите на даден български регион.

1.2. **techniques-to-prepare** – елемент наследник на *catalog*.

- 1.2.1. **technique** – елемент наследник на *techniques-to-prepare*. Елементът има два задължителни атрибута *name*(тип CDATA) и *id*(тип ID).

- 1.2.1.1. **name** – атрибут на елемента *technique* от тип CDATA. Носи информация за името на дадена техника за приготвяне.

1.2.1.2. id – атрибут на елемента *technique* от тип ID. Носи информация за инициалите на дадена техника за приготвяне.

1.3. recipes – елемент наследник на *catalog*.

1.3.1. recipe – елемент наследник на *recipes*. Елементът има атрибути *region-ID* и *recipe-ID*, и поделементи, служещи са описание на неговите характеристики: *name*, *image*, *author*, *origin*, *ingredients*, *preparation-steps*, *how-to-serve*, *suitable-drinks*, *medical-parameters*, *allergen*.

1.3.1.1. recipe-ID – атрибут на *recipe*, който представлява идентификатор на съответната рецепта.

1.3.1.2. region-ID – атрибут на *recipe*, който реферира към атрибута *id* на елемента *region*.

1.3.1.3. name – наследник на елемента *recipe* от тип PCDATA. Има атрибут *food-type* от тип CDATA. Носи информация за името на ястието, описано в рецептата.

1.3.1.3.1. food-type – атрибут на елемента *name* от тип CDATA. Носи информация за вида на ястието, описано в рецептата (десерт, предястие, основно или др.)

1.3.1.4. image – наследник на елемента *recipe* от тип EMPTY. Има един атрибут *href*.

1.3.1.4.1. href – атрибут на елемента *image* от тип ENTITY. Атрибутът реферира към ресурс, който не е в XML формат. Използва се с цел визуализиране на изображения за рецептите в проекта.

1.3.1.5. author – наследник на елемента *recipe* от тип PCDATA. Има атрибут *nationality* от тип CDATA. Носи информация за автора на ястието, описано в рецептата (ако има такъв).

1.3.1.5.1. nationality – атрибут на елемента *image* от тип CDATA. Носи информация за националността на автора на ястието, описано в рецептата.

1.3.1.6. origin – наследник на елемента *recipe* от тип PCDATA. Носи информация за произхода на ястието, описано в рецептата.

1.3.1.7. ingredients – наследник на елемента *recipe*. Съдържа поделемент *list*.

1.3.1.8. list – поделемент на *ingredients*, чиято цел е да представи списък с необходимите съставки за ястието, описано в рецептата. Съдържа поне един поделемент *item*.

1.3.1.8.1.1.item – поделемент на *list*. Носи информация за една съставка от списъка с такива за съответното ястие, описано в рецептата.

1.3.1.9. preparation-steps – наследник на елемента *recipe* от тип PCDATA. Има един атрибут *technique-ID*. Носи информация за стъпките за приготвяне на ястието, описано в рецептата.

1.3.1.9.1. technique-ID – атрибут на *preparation-steps*, който реферира към атрибута *id* на елемента *technique*.

1.3.1.10. how-to-serve – наследник на елемента *recipe* от тип PCDATA. Носи информация за предложения за сервиране на ястието, описано в рецептата.

1.3.1.11. suitable-drinks – наследник на елемента *recipe* от тип PCDATA. Носи информация за подходящи напитки към ястието, описано в рецептата.

1.3.1.12. medical-parameters – наследник на елемента *recipe* от тип PCDATA. Носи информация за медицинските параметри на ястието, описано в рецептата.

1.3.1.13. allergens – наследник на елемента *recipe*, чиято цел е да представи списък с алергени в ястието, описано в рецептата. Съдържа поне един поделелемент *allergen*.

1.3.1.13.1. allergen – поделелемент на *allergens*. Носи информация за един алерген от списъка с такива за съответното ястие, описано в рецептата.

2.3 Тип и представяне на съдържанието

Сборникът от рецепти представлява PDF файл, генериран чрез Altova XMLSpy 2023. PDF файлът се състои от 16 страници, изпълнени с текст и изображения, генерирани чрез XSL файл и XML файл, които използват UTF-8 кодиране и xml version 1.0. Изображенията са в .jpg формат и са оразмерени в съответствие с дизайна на документа.

Използвани текстови файлове:

- XML2022_project_project13_62541_62546.xml – 37.5 KB (38,492 bytes)
- XML2022_project_project13_62541_62546.xsl - 9.29 KB (9,522 bytes)
- XML2022_project_project13_62541_62546.dtd - 1.15 KB (1,179 bytes)
- RecipeCatalog.pdf - 3.00 MB (3,156,659 bytes)

Използвани са 11 файла с изображения:

- bobchorba.jpg - 109KB
- byalmazh.jpg - 260KB
- chomlek.jpg - 370KB
- kapama.jpg - 229KB

- korica.jpg - 523KB
- kosachkokiselo.jpg - 321KB
- patatnik.jpg - 286KB
- ribenachorba.jpg - 182KB
- shopskasalata.jpg - 301KB
- taramahaiver.jpg - 173KB
- tortagarash.jpg - 258KB

Подробно съдържание на PDF файла:

1. Заглавна страница
 - a. Изображение – korica.jpg
2. Рецепти от Южен-централен регион
 - a. Пататник
 - i. Изображение – patatnik.jpg
 - ii. Текст
3. Рецепти от Северозападен регион
 - a. Косачко кисело
 - i. Изображение – kosachkokiselo.jpg
 - ii. Текст
 - b. Бял муж
 - i. Изображение – byalmazh.jpg
 - ii. Текст
4. Рецепти от Северен-централен регион
 - a. Торता „Гараш“
 - i. Изображение – tortagarash.jpg
 - ii. Текст
5. Рецепти от Североизточен регион
 - a. Тарама хайвер
 - i. Изображение – taramahaiver.jpg
 - ii. Текст
6. Рецепти от Югозападен регион
 - a. Капама
 - i. Изображение – капама.jpg

- ii. Текст
 - b. Бански чомлек
 - i. Изображение – chomlek.jpg
 - ii. Текст
- 7. Рецепти от Югоизточен регион
 - a. Бургаска рибена чорба
 - i. Изображение – ribenachorba.jpg
 - ii. Текст
- 8. Рецепти с неизвестен произход
 - a. Шопска салата
 - i. Изображение – shopskasalata.jpg
 - ii. Текст

Бележка: Източниците, от които са взети всички изображения и текстови съдържания, изброени по-горе, са описани в точка 7 „Използвани литературни източници и Уеб сайтове“ от текущия документ.

3 Дизайн

Самия документ започва с изброяването на регионите, от които ще идват ястията, като се подава само най-важната информация, която включва името на региона и ID към него, за да може по някакъв начин дадена рецепта да може да се реферира към него, което в последствие ще разберем как става. Тъй като всяко ястие си има различна техника на приготвяне като за пример: печене, варене и други, след регионите ние сме включили и съответни техники за приготвяне като те отново по подобен начин изглеждат като горния блок, тоест всяка техника си има своето име и съответно ID, към което дадено ястие може да бъде реферирано. Избрахме техниките да бъдат по този начин, отделени като отделен елемент, към които поделементи ще бъдат реферирани, защото то е нещо основополагащо и ключово за направата на едно ястие. И за регионите, и за техниките за приготвяне ние сме добавили по един елемент, който е за „незнаен“ регион, тъй като не винаги е ясно уточнено от къде произлиза дадено ястие и „Нито едно от изброените“ за съответната техника, като това като за бъдещо развитие могат да бъдат добавяни и още техники и да се добави такава, която ще отговаря на ястието и да бъде променено за съответното него. На показаната снимка отдолу първият ред е закоментиран, за да може да се визуализира в браузър, като за визуализация XML файла е ползван браузъра Google Chrome.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- <?xml-stylesheet type="text/xsl" href="XML2022_project_project13_62541_62546.xsl"?> -->
▼<catalog>
  ▼<regions>
    <region name="Северозападен" id="NW"/>
    <region name="Северен централен" id="NC"/>
    <region name="Североизточен" id="NE"/>
    <region name="Югозападен" id="SW"/>
    <region name="Южен централен" id="SC"/>
    <region name="Югоизточен" id="SE"/>
    <region name="Незнаен" id="NA"/>
  </regions>
  ▼<techniques-to-prepare>
    <technique name="Печене" id="PE"/>
    <technique name="Пържене" id="PR"/>
    <technique name="Варене" id="VA"/>
    <technique name="Пасиране" id="PA"/>
    <technique name="Нито едно от изброените" id="NN"/>
  </techniques-to-prepare>
  ►<recipes>
    ...
  </recipes>
</catalog>
```

След като бяха уточнени техниките и регионите, сега е време и за самите рецепти. Под тях слагаме вече един елемент "recipes", който ще съдържа поделементи "recipe" за всяка една от изброените рецепти. За да можем добре да подредим информацията за всяка една от тях, ние първо, естествено започваме с името на дадената рецепта и като атрибут на елемента име е сложен и типа ястие което е. За атрибут на елемента "recipe" ние също слагаме името на дадената рецепта, защото когато бъдат визуализирани в браузър и бъдат свити със стрелкичката от лявата страна, да може пак да се разпознае коя е дадената рецепта, а като друг атрибут е и от кой регион

е. Под името на рецептата слагаме, с помощта на XML ENTITIES, снимката, защото тя е външна за нашия документ единица, до която трябва да се стигне по някакъв начин, като това става със съответната референция. Пример от нашия код за достъпване на дадена снимка е:

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

което се намира още в началото, където се намира във вътрешното DTD в XML документа. За самото него ще опишем по-нататък в документа. За вкарване на самата снимка в XML документа ние ползваме елемента `<image href="kapama"/>` където с помощта на href ние реферираме към DTD вътрешната дефиниция. Всичкото това ще можем да видим на следващата снимка, откъс от документа. Важно нещо за една рецепта и нейния автор, като него сме го сложили под снимката с атрибут на елемента, съответната националност. За всяка една рецепта е хубаво и интересно да имаме кратка история за нейния произход, като нея също я включваме в документа. Няма как да направим някаква рецепта без да имаме нейните съставки. Тях сме ги включили като елемент „ingredients“ с поделелемент „list“ и в него сме изброили всичките съставки, нужни за направата. Като всеки един отделен елемент item е поделелемент на елемента list, което е PCDATA. Имайки съставките, вече трябва да покажем и как се приготвя самото ястие. Това се случва с елемента „preparation-steps“, който съдържа цялата информация под формата на PCDATA, за това как трябва да бъде направено даденото ястие. Малко по – нагоре беше споменато, че имаме „techniques-to-prepare“, към което ще бъде реферирано, тъй като даденото ястие може да има различен начин на приготвяне. Това го постигаме като слагаме „technique-ID“ като атрибут на дадения елемент и му даваме някоя от изброените стойности за ID в „techniques-to-prepare“, към която ще реферира. Още нещо изключително важно е и как самото ястие се поднася, като това го добавяме с елемента „how-to-serve“. Когато си приготвим дадено ястие е хубаво то да върви с някаква напитка. Тази напитка я добавяме като PCDATA на елемента „suitable-drinks“. Понякога много хора не желаят да ядат висококалорични храни, затова ние трябва да включим и някаква информация за даденото ястие. Това го правим с помощта на елемента „medical-parameters“. От друга страна и редица хора имат проблеми към даден алерген и затова ние сме длъжни да включим и кои са алергените, които се съдържат в даденото ястие. Те биват изброени като поделелементи „allergen“ на елемента „allergens“.

Впоследствие ние продължаваме по гореспоменатия начин да изброяваме и други такива рецепти до края на документа, за да може той да добие някаква пълнота, и естествено количество, тъй като нашата задача е да представим на потребителите добре подбран каталог с български рецепти.

Снимка от какво представлява и една рецепта:

```

▼<recipe ID="Капама" region-ID="SW">
  <name food-type="Основно ястие">Капама</name>
  <image href="капама"/>
  <author nationality="България">Unknown</author>
  <origin>Капама е традиционно българско национално ястие приготвяно от населението в котловината по поречието на река Места, наричана Разлог, в Югозападна България. Думата е с персийски произход. Оригиналната капама се готви в казан върху открит огън. Сега е по-известен друг начин на приготвяне - в запечатано с тесто глинено гърне (гювеч) се пече няколко часа във фурна.</origin>
  ▼<ingredients>
    ▼<list>
      <item>500 гр. свинско месо</item>
      <item>500 гр. телешко месо</item>
      <item>500 гр. пиле</item>
      <item>1 кисела зелка</item>
      <item>1 чаша ориз</item>
      <item>400 гр. суджук</item>
      <item>400 гр. кървавица</item>
      <item>1 чаша червено вино</item>
      <item>черен пипер</item>
      <item>червен пипер</item>
      <item>сол на вкус</item>
      <item>тесто за замазване</item>
    </list>
  </ingredients>
  <preparation-steps technique-ID="PE">Поставете един слой от предварително фино нарязаното кисело зеле в глинен съд. Добавете малко подправки. Поставете един слой от ориз. После наредете част от свинското, телешкото, пилешкото, както и парчета суджук и кървавица. Добавете още един слой от кисело зеле. Сложете подправките и поставете отново слой от всички видове месо. Продължете по същия начин, докато съдът се запълни. Затворете капака на съда с тесто. Поставете съда в студена фурна, печете на ниска температура (150 градуса). След като заври, печете най-малко 4 часа.</preparation-steps>
  <how-to-serve>Може да се сервира с картофено пюре или друга гарнитура по желание.</how-to-serve>
  <suitable-drinks>Ястието се сервира с червено вино.</suitable-drinks>
  <medical-parameters>Ястието е много калорично , да се консумира в умерено количество от хора на диета. Киселото зеле премахва стомашни проблеми, защото млечно-киселите бактерии в него, които действат противомикробно, влияят отлично върху чревната флора в полза на полезните чревни бактерии. Месото е едно от най-питателните храни и съдържа всички необходими аминокиселини за изграждане на мускули и кости.</medical-parameters>
  ▼<allergens>
    <allergen>бяло вино</allergen>
    <allergen>сулфити</allergen>
    <allergen>хистамин</allergen>
  </allergens>
</recipe>

```

А така изглежда целия XML документ

```

<!-- <?xml-stylesheet type="text/xsl" href="XML2022_project_project13_62541_62546.xsl"?> -->
▼<catalog>
  ►<regions>
    ...
  </regions>
  ►<techniques-to-prepare>
    ...
  </techniques-to-prepare>
  ▼<recipes>
    ►<recipe ID="Торта Гараш" region-ID="NC">
      ...
    </recipe>
    ►<recipe ID="Шопска салата" region-ID="NA">
      ...
    </recipe>
    ►<recipe ID="Капама" region-ID="SW">
      ...
    </recipe>
    ►<recipe ID="Боб чорба по манастирски" region-ID="NA">
      ...
    </recipe>
    ►<recipe ID="Бургаска рибена чорба" region-ID="SE">
      ...
    </recipe>
    ►<recipe ID="Пататник" region-ID="SC">
      ...
    </recipe>
    ►<recipe ID="Бански чомлек" region-ID="SW">
      ...
    </recipe>
    ►<recipe ID="Косачко кисело" region-ID="NW">
      ...
    </recipe>
    ►<recipe ID="Тарама хайвер" region-ID="NE">
      ...
    </recipe>
    ►<recipe ID="Бел муж" region-ID="NW">
      ...
    </recipe>
  </recipes>
</catalog>

```

Скелета на нашия XML документ изглежда така

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="XML2022_project_project13_62541_62546.xsl"?>
<!DOCTYPE catalog SYSTEM "XML2022_project_project13_62541_62546.dtd" [
    <!ENTITY ... (name) SYSTEM "....jpg" NDATA jpg>
    ...
]>

<catalog>
    <regions>
        <region name="..." id="..."/>
        <region name="..." id="..."/>
        ...
    </regions>
    <techniques-to-prepare>
        <technique name="..." id="..."/>
        <technique name="..." id="..."/>
        ...
    </techniques-to-prepare>
    <recipes>
        <recipe ID="..." region-ID="...">
            <name food-type="...">...</name>
            <image href="...">...</image>
            <author nationality="...">...</author>
            <origin>...</origin>
            <ingredients>
                <list>
                    <item>...</item>
                    <item>...</item>
                    ...
                </list>
            </ingredients>
            <preparation-steps technique-ID="...">...</preparation-steps>
            <how-to-serve>...</how-to-serve>
            <suitable-drinks>...</suitable-drinks>
            <medical-parameters>...</medical-parameters>
            <allergens>
                <allergen>...</allergen>
                <allergen>...</allergen>
                ...
            </allergens>
        </recipe>
        И още рецепти ...
    </recipes>
</catalog>
```

Където ... са празни полета, където може да бъде сложен текст или в някои от другите случаи означава и още поделементи (изброяване).

За да сме сигурни, че този документ е валиден, ние трябва да създадем един DTD файл, който да ни го подsigури това нещо. Както още в горните параграфи ние споменахме, и понятието вътрешно DTD в XML документа. Към момента (към сегашния вид на документа, който ще бъде разширяван и напред) нашето вътрешно DTD изглежда така

```
<!DOCTYPE catalog SYSTEM "XML2022_project_project13_62541_62546.dtd" [  
    <!ENTITY bobchorba SYSTEM "images/bobchorba.jpg" NDATA jpg>  
    <!ENTITY byalmazh SYSTEM "images/byalmazh.jpg" NDATA jpg>  
    <!ENTITY chomlek SYSTEM "images/chomlek.jpg" NDATA jpg>  
    <!ENTITY kapama SYSTEM "images/kapama.jpg" NDATA jpg>  
    <!ENTITY kosachkokiselo SYSTEM "images/kosachkokiselo.jpg" NDATA jpg>  
    <!ENTITY patatnik SYSTEM "images/patatnik.jpg" NDATA jpg>  
    <!ENTITY ribenachorba SYSTEM "images/ribenachorba.jpg" NDATA jpg>  
    <!ENTITY shopskasalata SYSTEM "images/shopskasalata.jpg" NDATA jpg>  
    <!ENTITY taramahaiver SYSTEM "images/taramahaiver.jpg" NDATA jpg>  
    <!ENTITY tortagarash SYSTEM "images/tortagarash.jpg" NDATA jpg>  
>]
```

Като със SYSTEM "XML2022_project_project13_62541_62546.dtd" ние извикваме нашето външно DTD, където ще се случва цялата валидация на документа, извън снимките. Още в горната точка за елемента <image> ние описахме как се представя графичното съдържание с помощта на тези XML entities.

За външния DTD документ ние трябва да валидираме всеки един от елементите дали отговаря правилно на изискванията за структурата на XML документа, както съответно и всеки един от поделементите дали отново правилно отговаря, съответно тяхната подредба, естествено, и бройка. Но не само елементите и поделементите ние трябва да валидираме, но и съответно техните атрибути ако имат такива. Не на последно място ние трябва и да подsigурим връзката между ID-тата (атрибути) на елементите <recipe ... region-ID="..."> и <preparation-steps technique-ID="..."> да с ID-тата на съответните региони и техники на приготвяне с помощта на атрибутите ID и IDREF в DTD документа. Нека онагледим написаното с код:

```
<!ELEMENT catalog (regions, techniques-to-prepare, recipes)>  
<!ELEMENT regions (region+)>  
<!ATTLIST region name CDATA #REQUIRED id ID #REQUIRED>  
....  
<!ATTLIST recipe ID CDATA #REQUIRED region-ID IDREF #REQUIRED>
```

Аналогично изглежда и за <!ATTLIST preparation-steps technique-ID IDREF #REQUIRED> спрямо <!ATTLIST technique name CDATA #REQUIRED id ID #REQUIRED>. По този начин ние вече сме подsigурили, че тези атрибути ще бъдат свързани правилно.

Към текущия момент нашия DTD документ има следния вид:

```
<!ELEMENT catalog (regions, techniques-to-prepare, recipes)>  
<!ELEMENT regions (region+)>  
<!ELEMENT region EMPTY>
```

```

<!ATTLIST region name CDATA #REQUIRED id ID #REQUIRED>
<!ELEMENT techniques-to-prepare (technique+)>
<!ELEMENT technique EMPTY>
<!ATTLIST technique name CDATA #REQUIRED id ID #REQUIRED>
<!ELEMENT recipes (recipe+)>
<!ELEMENT recipe (name, image, author, origin, ingredients, preparation-steps, how-to-serve, suitable-drinks, medical-parameters, allergens)>
<!ATTLIST recipe ID CDATA #REQUIRED region-ID IDREF #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ATTLIST name food-type CDATA #IMPLIED>
<!ELEMENT image EMPTY>
<!ATTLIST image href ENTITY #REQUIRED>
<!ELEMENT author (#PCDATA)>
<!ATTLIST author nationality CDATA #IMPLIED>
<!ELEMENT origin (#PCDATA)>
<!ELEMENT ingredients (list)>
<!ELEMENT list (item+)>
<!ELEMENT item (#PCDATA)>
<!ELEMENT preparation-steps (#PCDATA)>
<!ATTLIST preparation-steps technique-ID IDREF #REQUIRED>
<!ELEMENT how-to-serve (#PCDATA)>
<!ELEMENT suitable-drinks (#PCDATA)>
<!ELEMENT medical-parameters (#PCDATA)>
<!ELEMENT allergens (allergen+)>
<!ELEMENT allergen (#PCDATA)>
<!NOTATION jpg PUBLIC "jpg viewer">

```

Нека и накратко опишем какво се случва в следния код. Тъй като root елемента на нашия XML документ е catalog той съдържа следните поделемени – regions, techniques-to-prepare, recipes. Отдолу следва реда <!ELEMENT regions (region+)>, който оказва за първия поделемени на catalog, че може да съдържа веднъж или повече пъти поделемента region. В XML документа ние сме решили елемента region да е самозатварящ се елемент ние оказваме в dtd-то, че този елемент е <!ELEMENT region EMPTY>, но пак има атрибути, които трябва да ги валидираме като това се случва по следния начин - <!ATTLIST region name CDATA #REQUIRED id ID #REQUIRED>, като региона има име, което е CDATA, за не бъде разглеждано от XML парсера, като е задължително да присъства с #REQUIRED, и ID към което ще се реферира. Абсолютно същото нещо се случва и за techniques-to-prepare. Елемента recipes има поделемени recipe отново срещаш се веднъж или повече пъти като подредбата на последвалите поделемени на recipe е строго дефинирана. Той има и атрибути които ID (сложили сме го по този начин, защото той има вече елемент name и за да не се повтаря и като атрибут, ние сме избрали такова име), а самата цел на това ID както вече бяхме споменали е, за да можем да видим когато са collapse-нати в браузър елементите, коя рецепта е. Има и атрибут region-ID, който както вече описахме реферира към съответния регион. Вече следващите поделемени на recipe като name, author, origin, preparation-steps, how-to-serve, suitable-drinks, medical-parameters съдържат #PCDATA, която бива парсната от парсера. Name има и атрибут food-type CDATA #IMPLIED, защото не е задължително да бъде добавен, но е хубаво да се знае, какъв е типа на ястието. Същото се случва и за nationality на author елемента. Имаме елемента image, който е самозатварящ се и <!ELEMENT image EMPTY> обаче има атрибут <!ATTLIST image href ENTITY #REQUIRED>, който е задължителен и това е самата снимка, която е href entity. За ingredients той съдържа поделемени list, който е само един, а вече той има един или повече поделемента item. Същото е и за allergens. На последно място в документа се намира и <!NOTATION jpg PUBLIC "jpg viewer"> където NOTATION инструкцията към процесора указва тип данни, които не са тип XML

data, PUBLIC е за външен елемент, и може да бъде достъпен от всеки и XML процесора трябва да може да обработи, като това е файл от jpg с jpg viewer.

След като разгледахме самото DTD на нашия XML документ и сме подсигурили че е валиден и всички подавани данни ще са строго спазени, ние следва да видим по какъв начин ние ще предоставим цялата информация на потребителите, различна от суров формат (код). Съдържанието на нашия XML документ ще го представим като PDF документ чрез генерация посредством XSLT. За да бъде постигнато това нещо то трябва да бъде генерирано през XSL FO. Целия този код намира своето място в отделен файл с разширение .xsl, в който се съдържа цялата информация за това как ще бъде визуализиран pdf-а. Преди да преминем към този файл, XML документа трябва да бъде свързан с неко по някакъв начин. Точно тук идва и часта, където ще обясним какъв е онзи закоментиран ред в началото на XML документа с каталога.

```
<?xml-stylesheet type="text/xsl" href="XML2022_project_project13_62541_62546.xsl"?>
```

Този ред показва връзката между XML документа и XSL Style Sheet-а, като той е от тип "text/xsl" и самата връзка е с файла href="XML2022_project_project13_62541_62546.xsl". Вече в самия файл описваме как ще изглеждат страниците на pdf-а.

Тъй като това отново е XML тип документ ние трябва да започнем с реда <?xml version="1.0" encoding="UTF-8"?> и декларацията на файла <xsl:stylesheet version="1.0" xmlns:xsl=<http://www.w3.org/1999/XSL/Transform> xmlns:fo="http://www.w3.org/1999/XSL/Format">, като корен елемента stylesheet има атрибути указващи версията на style sheet-а, като показва официалното W3C XSLT пространство от имена. Същото е и за xmlns:fo което показва пространството от имена за fo. Първо нещо като поделелемент на stylesheet ние забелязваме <xsl:template match="/">, където match атрибута асоциира темплейт с даден елемент. В случая този темплейт ще се изпълнява за целия документ. Под него следва и fo:root елемента, който е налачен елемент и всички негови поделелементи трябва да са един обект <fo:layout-master-set>, съдържащ всички шаблони, които ще се използват в документа, и всички <fo:page-sequence>. Още в самото начало когато още нямаме нищо ние трябва да преценим как ще бъде оформени нашите страници, тоест колко ще е голяма, каква ширина ще има, какви margins ще има дадената страница (отгоре, отдолу, отляво и отдясно). Всичко както би изглеждало и в един MS word документ. Ние трябва да определим и как ще са ни header, footer, right and left side встъпление. Всичките тези изброени неща намират своето място като поделелементи на елемента layout-master-set. Пример за това от нашия документ за повече онагледеност е:

```
<fo:layout-master-set>
  <!-- All page templates go here -->
  <!--This is the part where we create the page and set its properties-->
  <!--The page width and height 297 by 210 because it is A4 page-->
  <!--Margins are set to be 0.5 cm-->
  <fo:simple-page-master master-name="RecipeCatalog"
    page-width="210mm" page-height="297mm"
    margin-top="0.5cm" margin-bottom="0.5cm"
    margin-left="0.5cm" margin-right="0.5cm">
    <fo:region-body margin="0mm"/>
  <!--This is header-->
```

```

<fo:region-before extent="0mm"/>
<!--This is footer-->
<fo:region-after extent="0mm"/>
<!--This is left side indentation-->
<fo:region-start extent="0mm"/>
<!--This is right side indentation-->
<fo:region-end extent="0mm"/>
</fo:simple-page-master>
</fo:layout-master-set>

```

Между редовете на кода има и коментари, за всеки един отделен ред какво прави и защо сме избрали да са така стойностите. Ние искаме нашия пдф да се визуализира като стандартна A4 страница, затова ние слагаме `page-width="210mm"` `page-height="297mm"` да са съответните стойности. Искаме всичките марджини да са 0.5, което би отговаряло на narrow page layout. В `</fo:simple-page-master>` ние дефинираме всичките тези стойности, защото там се дефинира размера и изгледа на страницата. Името на самия документ, към което всички следващи `page-sequence` ще реферират е зададено с атрибута „`master-name="RecipeCatalog"`“

След като сме определили как ще изглеждат страниците на документа, ние вече можем да вкарваме елементите, които ще се визуализират. Ние искаме каталога да започне с някаква заглавна страница, а не директно с елементите, затова ние я задаваме по следния начин:

```

<fo:page-sequence master-reference="RecipeCatalog">
  <fo:flow flow-name="xsl-region-body">
    <fo:block position="absolute" text-align="center">
      <fo:external-graphic src="url('images/korica.jpg')"
        text-align="center"
        margin-top="-1cm"
        margin-bottom="0mm" content-height="63%" content-
width="63%"></fo:external-graphic>
    </fo:block>
  </fo:flow>
</fo:page-sequence>

```

С `<fo:page-sequence master-reference="RecipeCatalog">` ние задаваме нова последователност от страници, за които даденото нещо което искаме да визуализираме, ще се изпълни. Тази последователност реферира `master-name` с атрибута `master-reference`. `Fo:flow` елемента съдържа всички елементи, които ще се показват на страниците, в случая ще е една – заглавната. Дадената „единица елемент“, която ще се показва на страницата ние я слагаме като поделемент на елемента `fo:block`, като можем да окажем и някои допълнителни данни за нея, като оформления. Заглавната ни страница ще съдържа специално създадена от нас графика (снимка) като корица. Позицията на тази корица ще е “absolute”, а `text-align` – “center” или казано центрирана. Корицата ще я вкараме с помощта на следния елемент

```

<fo:external-graphic src="url('images/korica.jpg')"
  text-align="center"
  margin-top="-1cm"
  margin-bottom="0mm" content-height="63%" content-width="63%"></fo:external-
graphic>

```

като добавяме и допълнителни атрибути за допълнително оформление за самата нея.

След като си зададохме началната страница, следва да направим и останалите страници. Започваме с нов page-sequence за визуализацията на тях, като отново master-reference="RecipeCatalog", сочи към мастер името. Отново задаваме един flow елемент, който ще съдържа елементите, които ще се визуализират на пдф страниците. Даваме му име с помощта на атрибута flow-name. Важно нещо, което ще важи за рецептите занапред, е че всичките рецепти ще бъдат филтрирани по региони. За да можем да постигнем това нещо ние трябва да ги разделим на, както в още частта за регионите от XML файла за 7 отделни части, всяка за отделния си регион. С цел избягване на прекалено повторение последвалия код и описание ще важи за всеки един регион.

```
<fo:page-sequence master-reference="RecipeCatalog">
  <fo:flow flow-name="xsl-region-body">
    <fo:block position="relative"
              text-align="center"
              font-family="Monotype Corsiva"
              font-style="normal"
              font-size="22pt"
              padding-before="10mm">
      Рецепти от Южен-централен регион
    </fo:block>
    <fo:block position="relative" padding-before="5mm">
      <xsl:for-each select="/catalog/recipes/recipe[@region-ID='SC']">
        <xsl:apply-templates select="."/>
      </xsl:for-each>
    </fo:block>
    ..... за другите региони
  </fo:flow>
</fo:page-sequence>
```

За всеки един отделен регион ние искаме да изкараме като заглавие на самия регион, а под това заглавие да започва изреждането на всяка една рецепта за дадения регион, като това ще го покажем малко по-надолу. Започваме с fo:block за заглавието (името на региона) като със стилистична цел ние му добавяме и атрибути за позиция, къде ще се намира на страницата текста, какъв ще е типа шрифт, стила на шрифта, естествено, големината на шрифта и даден падинг. След което следва и името на самия регион. Под името на региона ние отваряме нов блок, който ще ни помогне с визуализацията на всичките рецепти. За да можем да ги изредим, тъй като те са повече от една в XML документа ние трябва да използваме for-each цикъл на който му подаваме Xpath-a `"/catalog/recipes/recipe[@region-ID='Съответното ID на региона по който ще търсим']`. Сега за всеки една рецепта, която е съвпаднала ние трябва да обработим данните, които ще се визуализират (за всички елементи и поделелементи на recipe от XML документа) с реда `<xsl:apply-templates select="."/>`. Като надолу ще имаме специален отделен темплейт за рецептата. *Apply-templates* избира всички темплейти от входното дърво и съответно се прилага този темплейт, който отговаря.

На последно място преди да сме преминали към рецептите ние трябва да затворим root елемента и темплейта

```
</fo:root>
</xsl:template>
```

Сега следва и частта с рецептите като както споменахме вече по-нагоре, че самата рецепта се съдържа в отделен темплейт `<xsl:template match="recipe">`.

```

<xsl:template match="recipe">
  <fo:block position="relative"
    text-align="center"
    font-family="Monotype Corsiva"
    font-style="normal"
    color="black"
    font-size="20pt"
    padding-before="10mm">
    <xsl:value-of select="./name/text()"/>
  </fo:block>

  <fo:block position="absolute"
    text-align="center"
    padding-before="10mm"
    padding-after="5mm"
    content-height="25%"
    content-width="25%">
    <xsl:apply-templates select="./image"/>
  </fo:block>

  <fo:block position="relative" font-family="Book Antiqua" font-style="normal" font-size="14pt" color="black"
    padding-before="5mm" padding-after="2mm">
    Произход:
    <fo:block position="relative" padding-before="5mm" padding-left="5mm" font-size="12pt">
      <xsl:value-of select="./origin/text()"/>
    </fo:block>

  </fo:block>

  Необходими продукти:
  <fo:block position="relative" padding-before="5mm" padding-left="5mm" font-size="12pt">
    <xsl:for-each select="./ingredients/list/item">
      <fo:block start-indent="50pt">
        <xsl:apply-templates select="."/>
      </fo:block>
    </xsl:for-each>
  </fo:block>

  </fo:block>

  <fo:block position="relative" font-family="Book Antiqua" font-style="normal" font-size="14pt" color="black"
    padding-before="5mm" padding-after="2mm">
    Стъпки на приготвяне:
    <fo:block position="relative" padding-before="5mm" padding-left="5mm" font-size="12pt">
      <xsl:value-of select="./preparation-steps/text()"/>
    </fo:block>
  </fo:block>

  <fo:block position="relative" font-family="Book Antiqua" font-style="normal" font-size="14pt" color="black"
    padding-before="5mm">
    Начин на сервиране:
    <fo:block position="relative" padding-before="5mm" padding-left="5mm" font-size="12pt">
      <xsl:value-of select="./how-to-serve/text()"/>
    </fo:block>
  </fo:block>

  <fo:block position="relative" font-family="Book Antiqua" font-style="normal" font-size="14pt" color="black"
    padding-before="5mm" padding-after="2mm">
    Подходящи напитки:
    <fo:block position="relative" padding-before="5mm" padding-left="5mm" font-size="12pt">
      <xsl:value-of select="./suitable-drinks/text()"/>
    </fo:block>
  </fo:block>

```

```

<fo:block position="relative" font-family="Book Antiqua" font-style="normal" font-size="14pt" color="black"
padding-before="5mm" padding-after="2mm">
  Медицински параметри:
  <fo:block position="relative" padding-before="5mm" padding-left="5mm" font-size="12pt">
    <xsl:value-of select="./medical-parameters/text()"/>
  </fo:block>
</fo:block>

<fo:block position="relative" font-family="Book Antiqua" font-style="normal" font-size="14pt" color="black">
  Алергени:
  <fo:block position="relative" padding-before="5mm" padding-left="5mm" font-size="12pt">
    <xsl:for-each select="./allergens/allergen">
      <fo:block start-indent="50pt">
        <xsl:apply-templates select="."/>
      </fo:block>
    </xsl:for-each>
  </fo:block>
</fo:block>
</xsl:template>

```

Както знаем, че рецептата има следната [структура](#):

Ние трябва да направим така, че всеки един поделемента на рецептата да се визуализира по правилен начин, като всеки един поделемента ще седи в отделен блок, за да бъдат визуализирани поотделно. Първо започваме с името на рецептата. Отново ние си задаваме стил, и с `<xsl:value-of select="./name/text()"/>` ние изкарваме стойността на елемента от XML документа. Следващото нещо, което трябва да се появи под името на рецептата, е снимката. Тя от своя страна с `<xsl:apply-templates select="./image"/>` което съвпада с темплейта за снимката:

```

<xsl:template match="image">
  <fo:external-graphic src="{unparsed-entity-uri(@href)}" content-
height="40%" content-width="40%"/>
</xsl:template>

```

чиято цел е да визуализира снимката, като `<fo:external-graphic>` служи за извличане на снимката, външна за "XML result tree" на пространството от имена fo. `{unparsed-entity-uri(@href)}` Служи за даване на достъп до декларации на не парснати обекти в DTD дефиницията.

За произход, стъпки на приготвяне, начин на сервиране, подходящи напитки и медицински параметри блоковете са почти идентични, затова с цел избягване на повторение ние ще ги обобщим. Самите блокове започват със съответната стилизация на това как бихме искали да изглеждат подзаглавията, а във вътрешен блок (подблок) ние извличаме стойността на `(.text())` елемента от документа.

Специфични остават само „Необходими продукти“ и „Алергени“ понеже за тях ние искаме те да се визуализират като списъци. Да започнем с „Необходими продукти“ - изброяваме всеки един от поделемента ("items") на вътрешния елемент "list" с `for-each` и за всеки един поделемента ние слагаме встъпление навътре „50pt“ за да може да бъдат открити от останалия текст, като вече всеки един *item* си има свой собствен темплейт:

```

<xsl:template match="ingredients">
  <fo:block>
    <xsl:value-of select="./item/text()"/>
  </fo:block>
</xsl:template>

```

А що се отнася до „Алергени“ нещата се случват по почти подобен начин, само че всеки алерген си има свой собствен темплейт:

```
<xsl:template match="allergens">
  <fo:block>
    <xsl:value-of select="./allergen/text()" />
  </fo:block>
</xsl:template>
</xsl:stylesheet>
```

След като показахме как изглежда самия .xsl файл и как би трябвало да изглежда трансформацията, е време да покажем и финалния вариант на това как изглежда нашия каталог в pdf формат.

Началната страница



Страници с рецепти по региони

Рецепти от Южен-централен регион

Пататник



Произход:

Името му идва от „патати“ – така наричат картофите в родопския край. Запознати твърдят, че колкото са селата в Родопите, толкова са и различните рецепти за него. Рецептата се е предавала от поколение на поколение до наши дни. Пататникът е типично ястие от Родопския край. По оригинална рецепта се приготвя в тиган или в сач, под похлупак, като се използват картофи, яйца, лук и сирене.

Необходими продукти:

1 кг стари картофи
2 яйца
3-4 глави кром 3-4 глави кромид лук
200 гр. сирене
5 стръка джоджен
2 супени лъжици олио
сол на вкус

Стъпки на приготвяне:

Картофите се измиват, обелват се и се настъргват на едро. Слагат се в кухненска кърпа и се изцеждат от течността. Лукът се нарязва на ситно или се настъргва. В купа се смесват картофите и

лукът с яйцата, джоджена и сиренето. Всичко се посолява и се разбърква. В тигана или сача се сипва олиото и се загрева. Слага се сместа за пататника, като се заглажда с дървена лъжица и се притиска към дъното на съда. Ястието се пържи 10-15 минути похлупено, след което се обръща върху капака на съда за готвене и отново се връща за още 10 минути, докато се изпържи и от другата страна.

Начин на сервиране:

Сервирайте пататника топъл, като по желание можете да го поръсите с кашкавал или да го овкусите с чубрица и самардала.

Подходящи напитки:

Поднесете с кисело мляко, айран, бърканица.

Медицински параметри:

В състава на картофа има малко натрий и мазнина. Но това не са единствените плюсове. В състава на картофа се съдържа скорбяла, която помага да се понижи холестеролът. Има и огромно количество калий. Калият помага за профилактиката на сърдечните заболявания. Използва се в много диети.

Алергени:

яйца
сирене

Рецепти от Северозападен регион

Косачко кисело



Произход:

Сиромашка и постна, но вкусна и здравословна е кухнята на балканджиите, която се е запазила и до днес. Странните рецепти е измисляла турлашката домакиня от немотия. Изключително вкусните постни гозби са емблематични, а самобитният им език е оставил отпечатък и в наименованията, използвани в кухнята им - скроб, косачко кисело, компировица, коприва цоловарка, бело пръжено шльопаво, керкеле, сиромашко пръжено, сиромашка закуска, сиромашка лютеница. Торлашката кухня използва малко подправки - зелен джоджен лете за чушките с боб, мащерка, мерудия (магданоз), бабина душица и най-вече копър. Без него нито косачкото кисело, нито зелената лютеница биха били така вкусни и пикантни. Чесънът е на почит.

Необходими продукти:

сухи чушки
краставица
чесън
копър
вода
сол
оцет
захар

Стъпки на приготвяне:

Приготвя се от нарязани сварени предварително сухи чушки, нарязана на ситно краставица, чесън и копър. Всичко това се залива със студена вода и се овкусява със сол, оцет и захар. Трябва да престои поне час – два, за да проникнат съставките една в друга. Поднася се като предястие.

Начин на сервиране:

Сервира се студена.

Подходящи напитки:

Ястие, което по традиция се приготвя само през лятото.

Медицински параметри:

То не засища, но охлажда в жегите и възбужда апетита.

Алергени:

няма

Бел муж



Произход:

Бел муж е традиционно жетварско ястие. Легендата разказва, че селяните на полето гощавали Свети Петър със сирене, но било толкова горещо, че то се разтопило. Тогава светецът го поръсил с брашно и се получила гозба за чудо и приказ. Легендата за ястието Бял муж, което е характерно за жителите на селата високо в Балкана, е завладяваща. Интерес обаче буди и фактът, че това е единственото ястие в българската фолклорна традиция, което се приготвя от мъж.

Необходими продукти:

1 кг прясно меко сирене
200 гр. брашно
1 ч.л. сол
200 мл овче прясно мляко

Стъпки на приготвяне:

Сиренето се поставя в дълбок съд и отгоре се сипва прясното мляко, солта и разтвореното във вода брашно. Готви се на силен огън, като непрекъснато се разбърква с дървена лъжица, докато кашата се стъсти.

Начин на сервиране:

Сложете в чиния за сервиране и поднесете по желание с мед или сладко по избор. Може да се намаже на филия хляб.

Подходящи напитки:

Консумира се с бяло вино.

Медицински параметри:

Сиренето е продукт, който предлага богатство от ползи за здравето. Ползните свойства на този деликатес се дължат на набор от минерални вещества (калций, магнезий, фосфор, йод, желязо, селен, цинк, мед) и такива витамини като В, В1, В2, В12, РР, С, D, Е, F. Този комплект осигурява пълен витаминен комплекс за пълното развитие на тялото.

Алергени:

мляко
казеин
лактоглобулин

Рецепти от Северен-централен регион

Торта "Гараш"



Произход:

Легендарната торта "Гараш" започва славния си път от... България! Оказва се, че вкусният шоколадов деликатес за пръв път е представен у нас преди близо 120 години! През 1885 г. в Русе пристига прочут сладкар – Коста Гараш от Австро-Унгария. Сладкото изкушение, което той измисля, носи именно неговото име. Но как започва всичко? В края на XIX век в Русе е построен първият за България гранд хотел – „Ислях хане“. Мебелите, приборите, завесите са специално доставени от Виена и блясъкът на тогавашна Австро-Унгария се пренася в Русе. Когато пристига в града, Коста Гараш поема управлението на „Ислях хане“. Краят на XIX век, основния салон на хотел "Ислях хане" в Русе в салоните на хотела княз Александър Батенберг посреща крале. Именно за един от тези приеми Гараш създава тортата „Гараш“. След това я пренася в София в един от най-люксовите хотели за онова време – „Пахах“. Около 1900 година Коста Гараш се премества в София, за да поеме управлението на хотела. На партера се е намирала и сладкарницата, където столичани са могли да опитат парче „Гараш“.

Необходими продукти:

И други, които могат да се видят в генерирания pdf

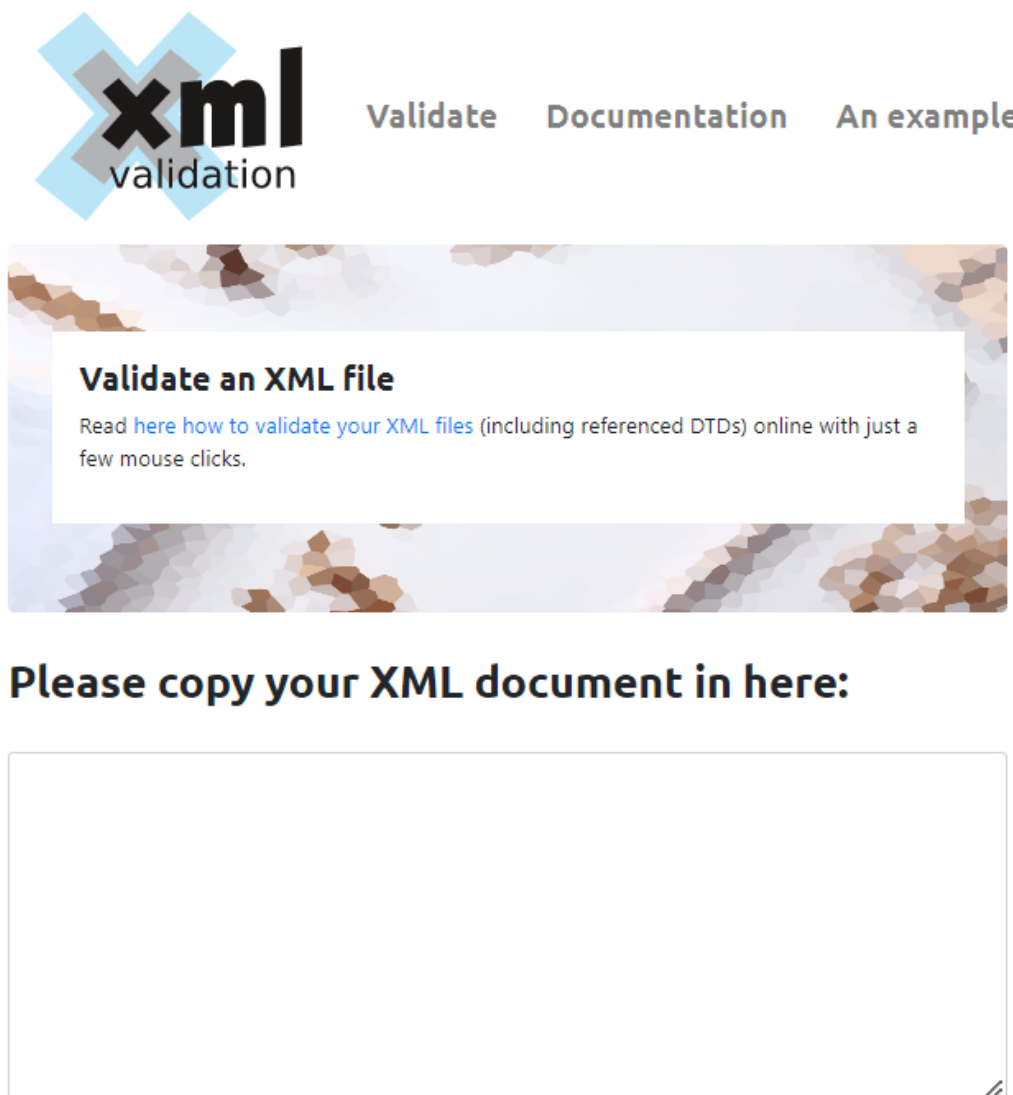
4 Тестване

<http://www.xmlvalidation.com> – за валидация на XML документа спрямо създадената от нас DTD схема.

<https://www.altova.com/xmlspy-xml-editor> - за генерация на PDF файлът, използвайки като входни данни създадените от нас XML, DTD и XSL файлове.

Стъпките на валидация и генерация чрез използваните инструменти е представена на следващите няколко фигури.

В XML валидатора чрез последователност от стъпки се дава възможността да се валидира XML документ спрямо дадена DTD схема. Първата стъпка е качване на XML файл, след това валидатора засича необходимия DTD файл и изписва неговото качване във валидатора. След като и двата файла са качени, чрез кликуването на бутона „Continue validation“ се извършва нужната валидация. На фигурите по-долу ясно се виждат изброените стъпки, както и че валидацията на нашите изходни файлове е минала без грешки.



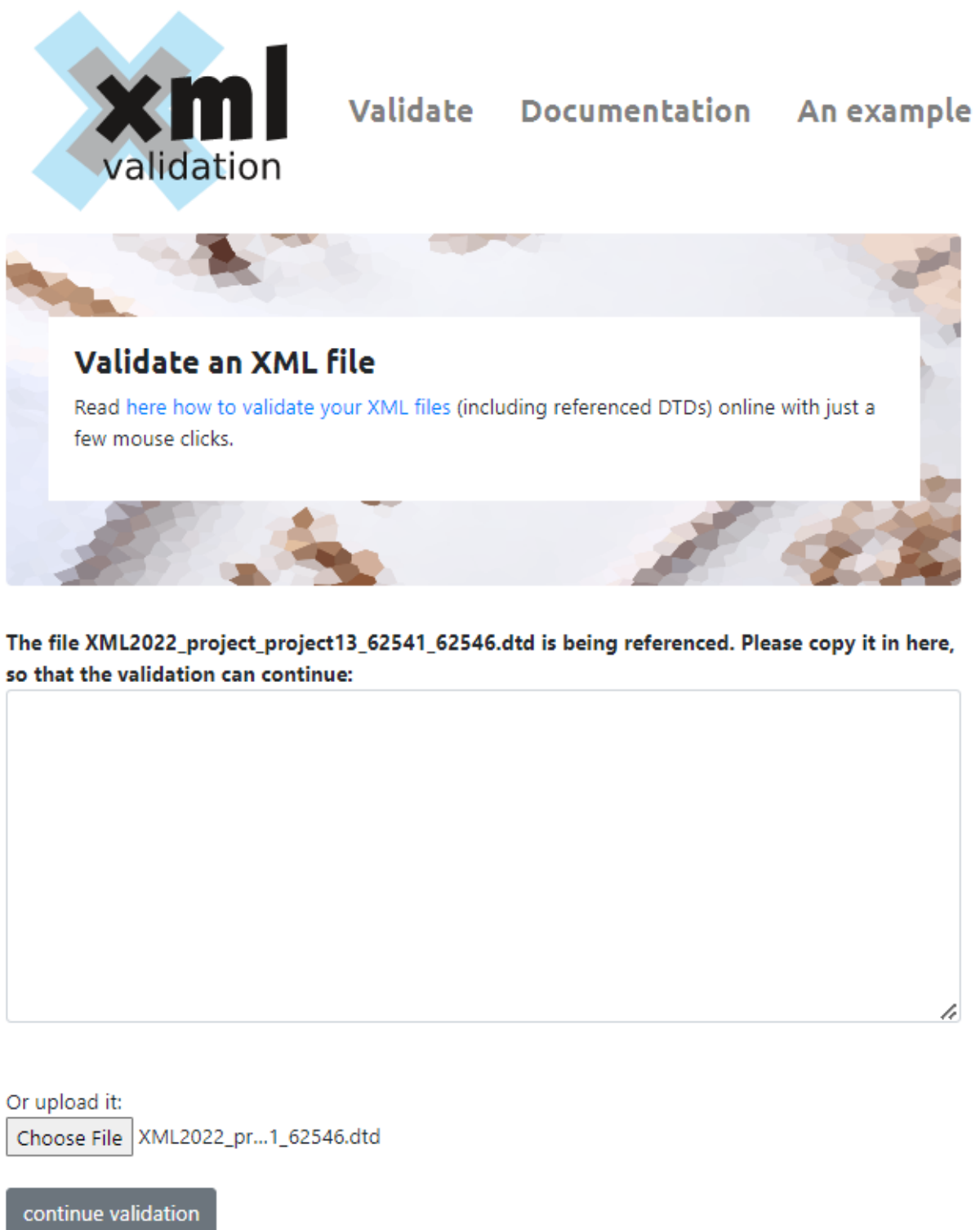
xml validation Validate Documentation An example

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:
 XML2022_pr...1_62546.xml

Фиг 1. Избор на XML документ за валидиране.



The image shows the homepage of an XML validation website. At the top left is a logo with a large 'x' and 'ml' in a blue and grey color scheme, with the word 'validation' underneath. To the right of the logo are three navigation links: 'Validate', 'Documentation', and 'An example'. Below the navigation links is a large banner with a background image of a mountain range. In the center of the banner is a white box with the text 'Validate an XML file' and a link to 'Read here how to validate your XML files'. Below the banner, there is a message stating that a specific DTD file is being referenced and needs to be copied into a provided text area. Below the text area, there is a section for uploading the file, with a 'Choose File' button and a text input field containing the file name 'XML2022_pr...1_62546.dtd'. At the bottom of this section is a 'continue validation' button.

xml
validation

[Validate](#) [Documentation](#) [An example](#)

Validate an XML file

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

The file XML2022_project_project13_62541_62546.dtd is being referenced. Please copy it in here, so that the validation can continue:

Or upload it:

Choose File

XML2022_pr...1_62546.dtd

continue validation

Фиг 2. Избор на DTD.



Validate

Documentation

An example

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:

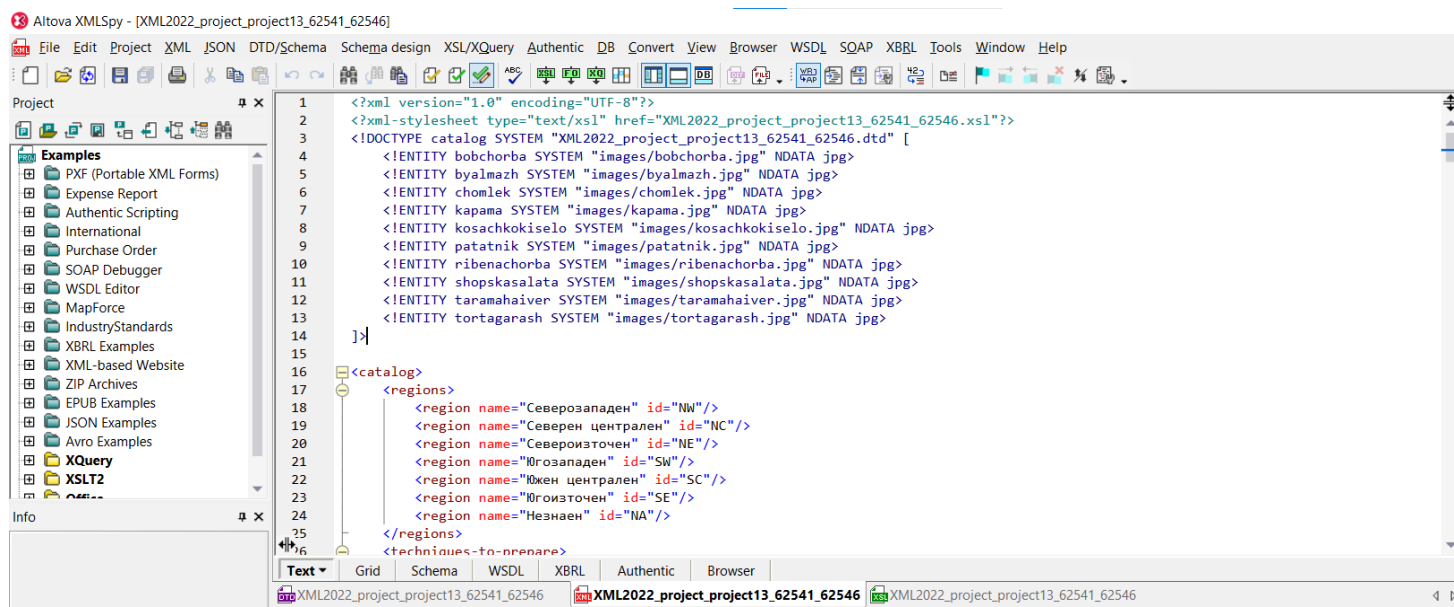
[XML document:](#)

[XML2022_project_project13_62541_62546.dtd](#)

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

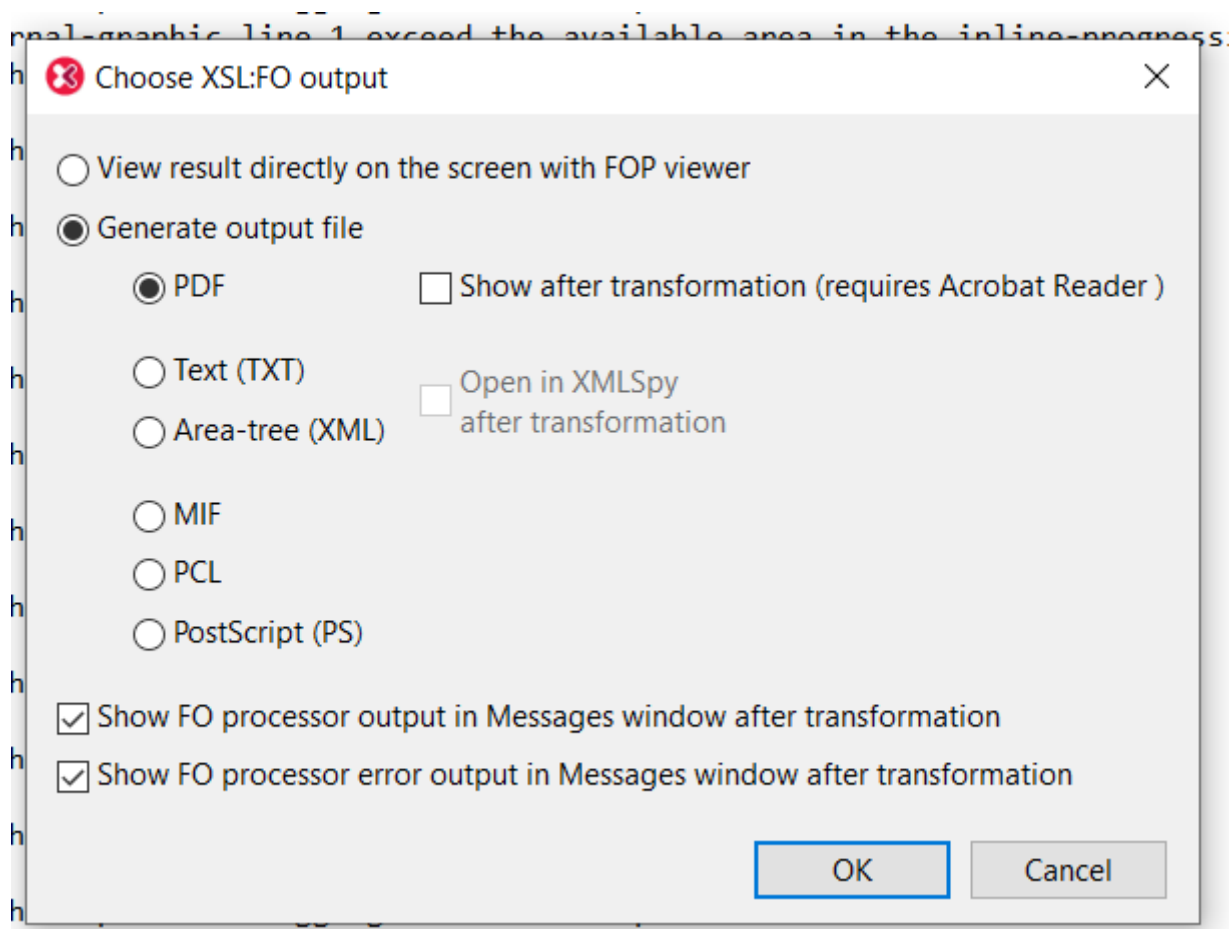
Фиг 3. Резултат от валидация.

Чрез Altova XMLSpy 2023 Editor генерирахме крайния PDF документ, съдържащ рецептите. Сега ще опишем стъпките на тази генерация. Програмата изисква необходимите файлове, а именно – XML, DTD и XSL документите, да се отворят.

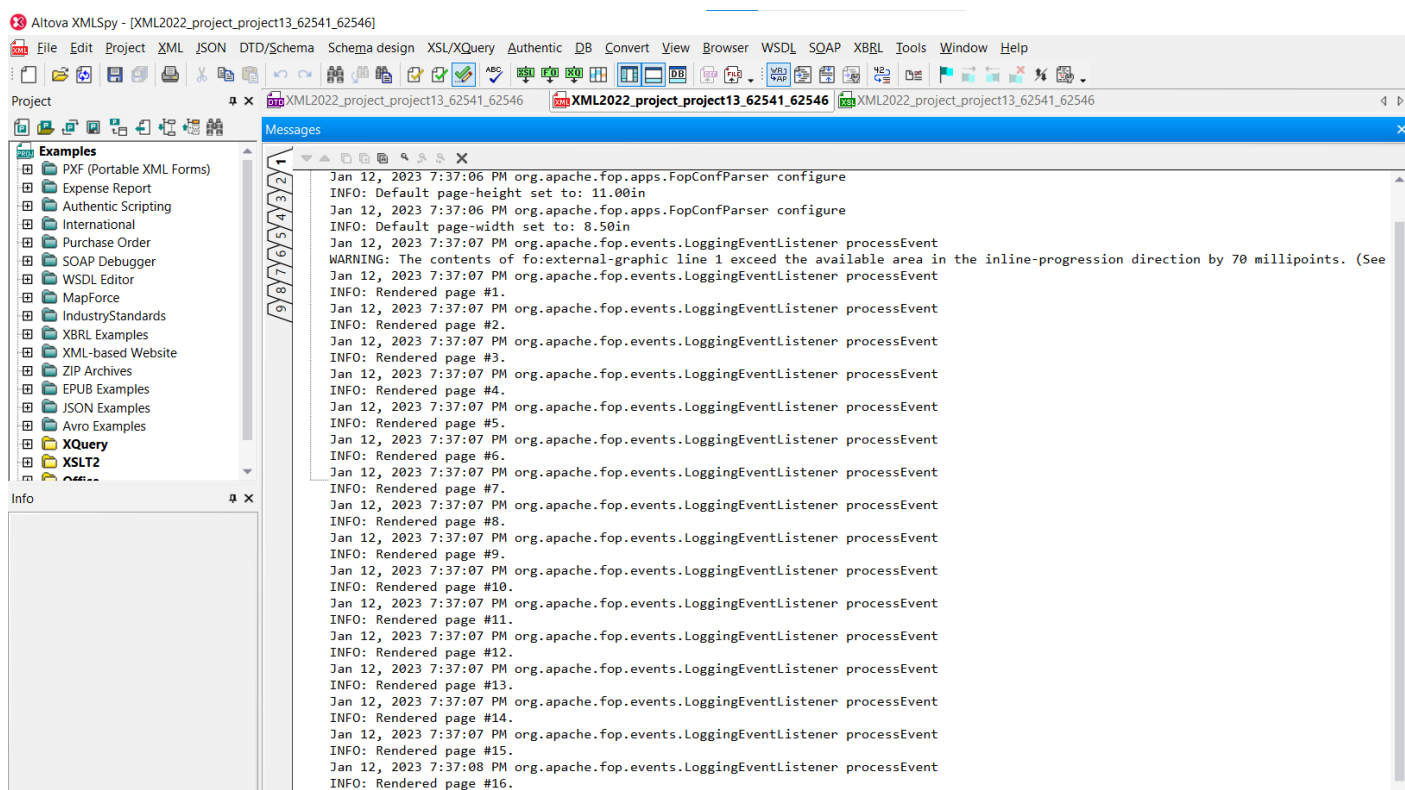


Избира се бутон “XSL-FO Transformation”, при което се изисква да се изберат изходните файлове и да се избере формата на желания генериран файл.

Фиг 4. Зареждане на изходните файлове в Altova XMLSpy 2023.



Фиг 5. Избор на функционалността за генериране на PDF файл



Фиг 6. Успешна генерация на шестнадесетте страници на PDF файла RecipeCatalog.

След като всичко е избрано, програмата генерира страниците на PDF файла спрямо зададените му документи. Изгледът на генерираните страници може да се види в точка 3 Дизайн – [Начална страница](#).

За да работи коректно Altova XMLSpy Editor 2023 изисква работеща Java, както и версия на Apache FOP.

5 Заключение и възможно бъдещо развитие

Проектът е добре структуриран благодарение на използването на езика XML и връзките между елементите в него са ясни и разбираеми. Чрез XSL-fo сравнително лесно се създава шаблон за генериране на PDF файл за желаните в проекта сборник от рецепти. При необходимост от промяна на съдържанието сравнително лесно може да се въведе такава, преправяйки нужните за целта на промяната документи. Резултатът от проекта е нагледен и добре подреден сборник от рецепти, удобен за употреба от всеки читател.

Сред възможностите за бъдещо развитие е добавяне на енергийни стойности към рецептите, както съдържание на белтъци, мазнини и въглехидрати. Възможно е и добавянето на още рецепти и тяхното филтриране в категории „Предястия“, „Десерти“, „Гарнитюри“ и тн. Тъй като готварската индустрия е богата откъм идеи, проектът винаги би могъл да се развива и разширява повече и повече.

6 Разпределение на работата

1. Събиране на необходими ресурси за проекта – Илиян Йорданов
2. Разработване на XML, DTD и XSL файлове – съвместна работа между Илиян Йорданов и Александра Радева
3. Документация – съвместна работа между Илиян Йорданов и Александра Радева
 1. Въведение - Илиян Йорданов и Александра Радева
 2. Анализ на решението
 - Работен процес – Илиян Йорданов
 - Структура на съдържанието – Александра Радева
 - Тип и представяне на съдържанието – Александра Радева
 3. Дизайн – Илиян Йорданов
 4. Тестване – Александра Радева
 5. Заключение и възможно бъдещо развитие – Александра Радева

6. Разпределение на работата - Илиян Йорданов и Александра Радева
7. Използвани литературни източници и Уеб сайтове – Илиян Йорданов и Александра Радева

7 Използвани литературни източници и Уеб сайтове

1. Рецепта Боб чорба - <https://www.supichka.com/%D1%80%D0%B5%D1%86%D0%B5%D0%BF%D1%82%D0%B0/424/%D0%B1%D0%BE%D0%B1-%D1%87%D0%BE%D1%80%D0%B1%D0%B0-%D0%BF%D0%BE-%D1%82%D1%80%D0%B0%D0%B4%D0%B8%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0-%D1%80%D0%B5%D1%86%D0%B5%D0%BF%D1%82%D0%B0>
2. Рецепта Бял мъж - <https://www.supichka.com/%D1%80%D0%B5%D1%86%D0%B5%D0%BF%D1%82%D0%B0/80/%D0%B1%D1%8F%D0%BB-%D0%BC%D1%8A%D0%B6-%D0%BA%D1%83%D1%82%D0%BC%D0%B0%D1%87-%D0%B1%D0%B5%D0%BB%D0%BC%D1%8A%D1%88>
3. Рецепта Бански Чомлек - https://recepti.gotvach.bg/r-5190-%D0%91%D0%B0%D0%BD%D1%81%D0%BA%D0%B8_%D1%87%D0%BE%D0%BC%D0%BB%D0%B5%D0%BA
4. Рецепта Капама - https://recepti.gotvach.bg/r-54165-%D0%9C%D0%BE%D1%8F%D1%82%D0%B0_%D0%BA%D0%B0%D0%BF%D0%B0%D0%BC%D0%B0
5. Рецепта Косачко кисело - <https://pendara.bg/recepti/kosachko-kiselo/>
6. Рецепта Пататник - <https://pendara.bg/recepti/kosachko-kiselo/>
7. Рецепта Бургаска рибена чорба - https://recepti.gotvach.bg/r-99620-%D0%91%D1%83%D1%80%D0%B3%D0%B0%D1%81%D0%BA%D0%B0_%D1%80%D0%B8%D0%B1%D0%B5%D0%BD%D0%B0_%D1%87%D0%BE%D1%80%D0%B1%D0%B0
8. Рецепта Шопска салата - https://recepti.gotvach.bg/r-3670-%D0%A8%D0%BE%D0%BF%D1%81%D0%BA%D0%B0_%D1%81%D0%B0%D0%BB%D0%B0%D1%82%D0%B0
9. Рецепта Тарама хайвер - https://recepti.gotvach.bg/r-62291-%D0%A2%D0%B0%D1%80%D0%B0%D0%BC%D0%B0_%D1%85%D0%B0%D0%B9%D0%B2%D0%B5%D1%80
10. Рецепта Торто „Гараш“ - https://recepti.gotvach.bg/r-82571-%D0%9E%D1%80%D0%B8%D0%B3%D0%B8%D0%BD%D0%B0%D0%BB%D0%BD%D0%B0_%D1%82%D0%BE%D1%80%D1%82%D0%B0_%D0%93%D0%B0%D1%80%D0%B0%D1%88_%D0%BE%D1%82_1885_%D0%B3
11. Снимка Боб чорба - <https://m3.netinfo.bg/media/images/15871/15871185/960-540-bon-apeti.jpg>
12. Снимка Бял мъж - <https://apriltsi.net/wp-content/uploads/2021/12/kutmach-e1618417679638.jpg>

13. Снимка Бански чомлек - https://www.culinarytourism.bg/cache/com_zoo/images/chomlek_e310005a93dc05c7e16ce62ec4a49597.jpg
14. Снимка Капама - <https://www.supichka.com/files/images/596/kapama.jpg>
15. Снимка Косачко кисело - <https://www.bgnes.bg/site/assets/files/0/04/41/055/festival-naturlashkata-kuhnya-v-chuprenskia-balkan-20200808-145241.840x500-pim2-0ea0d670.jpg>
16. Снимка Пататник - [https://m.1001recepti.com/images/photos/recipes/size_5/patatnik-ot-surovi-kartofi-iaica-sirene-i-luk-na-tigan-8a9657ae80db509466799eb2cc4d73e7-\[110927\].jpg](https://m.1001recepti.com/images/photos/recipes/size_5/patatnik-ot-surovi-kartofi-iaica-sirene-i-luk-na-tigan-8a9657ae80db509466799eb2cc4d73e7-[110927].jpg)
17. Снимка Бургаска рибена чорба - <https://i.ytimg.com/vi/UjDNRyXIL64/maxresdefault.jpg>
18. Снимка Шопска салата - [https://m.1001recepti.com/images/photos/recipes/size_5/klasicheska-shopska-salata-s-domati-krastavici-chushki-luk-i-sirene-a44e003a9f69e08c30bdee30f83f570e-\[103344\].jpg](https://m.1001recepti.com/images/photos/recipes/size_5/klasicheska-shopska-salata-s-domati-krastavici-chushki-luk-i-sirene-a44e003a9f69e08c30bdee30f83f570e-[103344].jpg)
19. Снимка Тарама хайвер - https://www.culinarytourism.bg/cache/com_zoo/images/tarama_haiver_7b463c2f4bcccc37eb20ec3e983570ac.jpg
20. Снимка Торта „Гараш“ - <https://i.ytimg.com/vi/b9TMtoaZwBc/maxresdefault.jpg>
21. XML валидатор - <http://www.xmlvalidation.com>