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

Факултет по математика и информатика

*Бакалавърска програма*

*„Софтуерно инженерство”*

**Предмет: XML технологии за семантичен Уеб**

***Зимен семестър, 2023/2024 год.***

**Тема №0123: „Шаблон за оформление на документация на курсов проект“ Курсов проект**

*Автори:*

*Тодор Кермеджиев, фак. номер …*

*Цветомир Гълъбов, фак. номер 62493*

януари, 2024

София

**Съдържание**

[1 Въведение 3](#_Toc5170)

[2 Анализ на решението 3](#_Toc5171)

[2.1 Работен процес 3](#_Toc5172)

[2.2 Структура на съдържанието 3](#_Toc5173)

[2.3 Тип и представяне на съдържанието 4](#_Toc5174)

[3 Дизайн 5](#_Toc5175)

[4 Тестване 7](#_Toc5176)

[5 Заключение и възможно бъдещо развитие 9](#_Toc5177)

[6 Разпределение на работата 10](#_Toc5178)

[7 Използвани литературни източници и Уеб сайтове 10](#_Toc5179)

# Въведение

Настоящият документ представлява шаблон за оформление на документация на курсов проект за предмета „XML технологии за семантичен Уеб“. Изискванията за изработване и представяне на проекта са описани на сайта на този предмет в Мудъл.

Насоки за разработка на секцията:

* Информацията в настоящия документ е напълно актуално
* Проектът ни позволява да могат бързо да се прелистват профилите на футболни клубове, мениджъри и първенства
* Проблемът се разрешава чрез XML базиран каталог, който съхранява основната информация за въпросните обекти и ги обединява логически
* Надолу в документа се съдържа информация за нашите анализ на решението, дизайн, тестване и заключение за бъдещо развитие на проекта.

# Анализ на решението

## Работен процес

Работният процес е следният:

* Данните са взети от wikipedia.org и google.com
* Те са ръчно въведени в документа, като е спазена йерархична наредба, която е описана по-долу
* Изходът е структурираната информация, която потребителят може да използва за своите цели

## Структура на съдържанието

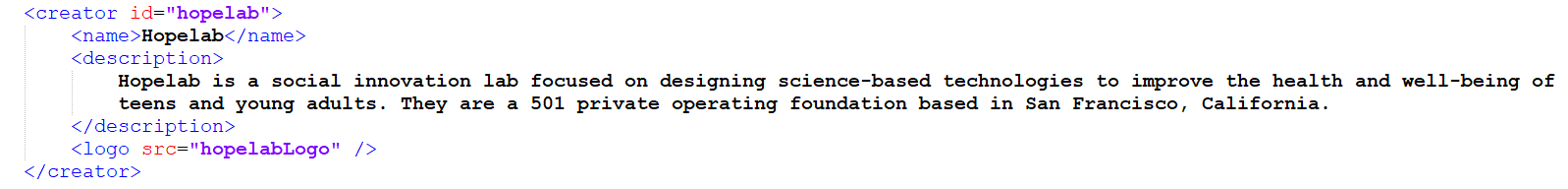
В проекта се съдържат три основни типа данни:

1.Елементи от тип game

A screenshot of a computer

Description automatically generated

2.Елементи от тип creator



3.Елементи от тип area

A close up of a text

Description automatically generated

Както може да се забележи, елементите от тип game имат уникален идентификационен номер, както и референции към приложната област, жанра от който произлизат, както и към създателя си , жанровете и създателите имат само уникален идентификационен номер.

## Тип и представяне на съдържанието

Използвали сме 4 текстови файла:

* seriousGames.xml – Size: 17502b
* seriousGames.xsl - Size: 10282b
* seriousGames.dtd - Size: 1380b
* seriousGames.pdf - Size: 12099921b

Използвали сме **28** файла с изображение:

* dragonBox\_cover.jpg - Size: 620kb
* dragonBox\_screenshot1.jpg - Size: 92kb
* dragonBox\_screenshot2.jpg - Size: 66kb
* factorion\_cover.jpg - Size: 1767kb
* factorion\_screenshot1.jpg - Size: 66kb
* factorion\_screenshot2.jpg - Size: 42kb
* foldit\_cover.jpg - Size: 36kb
* foldit\_screenshot1.jpg - Size: 222kb
* foldit\_screenshot2.jpg - Size: 734kb
* glasslabLogo.jpg - Size: 6kb
* hopelab.jpg - Size: 48kb
* Kahoot\_dragonbox\_logo.jpg - Size: 790kb
* microsoft\_flight\_Simulator\_cover.jpg - Size: 96kb
* microsoft\_flight\_Simulator\_screenshot1.jpg - Size: 36kb
* microsoft\_flight\_Simulator\_screenshot2.jpg - Size: 3701kb
* microsoft\_logo.jpg- Size: 17kb
* minecraft\_cover.jpg – Size: 282kb
* minecraft\_screenshot1.jpg – Size: 125kb
* minecraft\_screenshot2.jpg – Size: 104kb
* mojang\_logo.jpg – Size: 17kb
* re-mission2\_cover.jpg – Size: 56kb
* re-mission2\_screenshot1.jpg – Size: 28kb
* re-mission2\_screenshot2.jpg – Size: 25kb
* simcity\_cover.jpg – Size: 739kb
* simcity\_screenshot1.jpg – Size: 172kb
* simcity\_screenshot2.jpg – Size: 49kb
* uwCse\_logo.jpg – Size: 27kb
* wube\_logo.jpg – Size: 38kb

# Дизайн

1.1 Свързаност на данните

Както вече казахме в секция 2.2, връзките в проекта са следните:

* game -> genre
* game -> creator
* game -> alluvial area

За постигане на тези връзки използваме следните атрибути:

* game-> genre - genreRef=““



* game -> creator - creatorRef=""

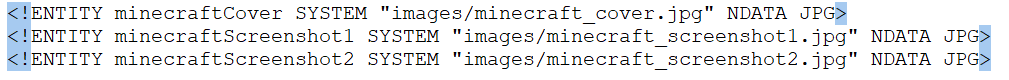
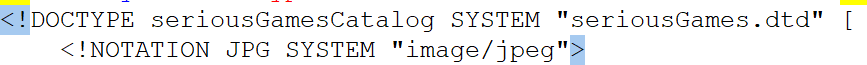


* game -> alluvial Area - alluvialAreaRef=""



1.2 Валидация

Валидацията чрез DTD се изразява в това, че всеки елемент, атрибут на елемент или поделемент е експлицитно декларирано:

* Елемент може ли да има стойност, ще се парсва ли от процесора или не?
* Елемент какви поделементи/атрибути има.
* Поделементите в каква последователност трябва да са, кои от тях са не/задължителни, кои от тях могат повече от веднъж да се повтарят.
* Атрибутите да на даден елемент какви стойности могат да приемат, дали са задължителни или не.
* Надграждане над основните типове стойности позволени от DTD o Какви стойности могат да приемат(за атрибутите) o какви под елементи или атрибути съдържат, в каква последователност трябва да бъдат o кои от тях за задължителни и кои не
* Например, за валидирането на елемента game сме използвали следното:
* **<!ELEMENT game (title, description, releaseDate, graphics)>**
* **<!ATTLIST game id ID #REQUIRED>**
* **<!ATTLIST game genreRef IDREF #REQUIRED>**
* **<!ELEMENT name (#PCDATA)>**
* **<!ELEMENT description (#PCDATA)>**
* **<!ATTLIST game alluvialAreaRef IDREF #REQUIRED>**
* **<!ATTLIST game creatorRef IDREF #REQUIRED>**
* **<!ELEMENT title (#PCDATA)>**
* **<!ELEMENT releaseDate (#PCDATA)>**
* За валидирането на нашия xml файл сме използвали външно и вътрешно DTD, като вътрешното само добавя допълнителни елементи, конкретни за стойностите на атрибутите на елементите в xml файла – например custom ENTITY, чрез които се реферира към съответните снимки :
* 
* Понеже използваме JPEG => не е сред основните типове стойности в вътрешното DTD трябва експлицитно да го добавим. Това става чрез :
* 
* Така ако решим може да преизползваме външното DTD за друг проект, без да съдържа референции локални за самия проект.

# Тестване

Тук обяснете как сте тествали предложеното решение:

* Направихме dtd валидация, за да сме сигурни, че проектът ни е валиден и добре форматиран
* Крайният резултат е представен в pdf формат, който е универсален за различните платформи за гледане на pdf файлове
* Примери от крайния резултат:

A screenshot of a video game

Description automatically generated

Screens screenshots of a video game

Description automatically generated

A paper with text on it

Description automatically generatedA document with text on it

Description automatically generatedA white background with black text

Description automatically generated

# Заключение и възможно бъдещо развитие

Резултатът от проекта е опростено портфолио в pdf формат. Заради липсата на всякакви визуални ефекти, това е лесен за използване каталог, който се фокусира върху предаването на информация. В съвременния свят съществуват множество формати за пренасяне на информация, които биха свършили същата работа. Пример за такъв формат е JSON, който до голяма степен може да прави същото, което прави XML.

Този проект е подходящ за бързи справки както от хора, които се интересуват от сериозните видео игри, така и от такива, които нищо не разбират. Той съдържа само най-важните атрибути на обектите и може да се използва както за експресно запознаване с дадена игри и разширение на общата култура за това какво са сериозните игри, какви са основните видове, както и тяхната приложна област.

# Разпределение на работата

Тодор Кермедчиев:

* Валидиране на xml файл чрез dtd
* Трансформация от xml към pdf чрез използване на xslt файл

Цветомир Гълъбов:

* Създаване на дизайна
* Събиране и внедряване на информацията
* Изготвяне на документацията

# Използвани литературни източници и Уеб сайтове

1. <https://en.wikipedia.org/wiki/Main_Page> цялата информация за обектите в проекта, включително изображенията
2. Apache FOP <https://xmlgraphics.apache.org/fop/>
3. Google
4. https://github.com/apache/xmlgraphics-fop
5. xsl:fo <https://www.ibm.com/developerworks/library/x-xslfo/>
6. Лекции и упражнение от XML технологии за семантичен web, зимен семестър 2023/2024.