Dokumentace pro evidenci rezervací návštěv veterinární ambulance

Semestrální práce 4IZ238

Hynek Zemanec

Dokumentace pro evidenci rezervací návštěv veterinární ambulance: Semestrální práce 4IZ238

Hynek Zemanec

Table of Contents

Úvod

Tento dokument obsahuje dokumentaci k semestrální práci pro předmět 4IZ238 zaměřený na XML technologie na Vysoké škole ekonomické v Praze. Projekt se věnuje zpracování informací poskytnutých v rámci rezervací návštěv klientů malé veterinární ambulance a tyto informace zpřehledňuje v provázaných HTML dokumentech a PDF souboru.

Part I. Projekt

Table of Contents

Chapter 1. Zadání

Cílem projektu je zpracování systému evidující rezervace majitelů zvířat pro nezávislou veterinární ambulanci. Konkrétně se jedná o veterinární ambulanci se sídlem v Českých Budějovicích, zabývající se preventivní péčí a léčbou malých zvířat. Zadání zachycuje výstupy proběhlé analýzy potřeb veterinářky *MVDr. Dany Zemancové*.

Požadavky

Veterinář při výkonu své práce pracuje s řadou informací. Potřebuje vědět o jaké zvíře a plemeno se jedná, jaký je jeho věk, pohlaví a zdravotní stav a a užívanou dávku případných léků. Na základě těchto informací veterinář může stanovit diagnózu případného onemocnění a doporučit léčebu. Za dodržování stanovené léčby je ovšem zodpovědný majitel. Nejen z toho důvodu veterinář musí evidovat i kontaktní informace na majitele, včetně jeho bydliště pro případ nutného výjezdu.

I přes objednání na určitý čas se běžně stává, že časový odhad návštěv neodpovídá realitě. V takových případech musí veterinář prioritizovat ošetření vážnějších případů, případně domluvit s klientem chirurgický zákrok mimo ordinační hodiny.

Pro snadnější rozlišení zvířat je v neposlední řadě nezbytné zvířatům přiřadit jednoznačný identifikátor. Na to u psů pamatuje legislativa, která nařizuje jejich povinné očipování.

Řešení

Výstupem systému jsou přehledně poskytnuté informace o počtu a předmětu rezervací v jednotlivých dnech daného období. Rezervace jsou shromážděny do izolovaných skupin po dnech a seřazeny podle času předpokládané návštěvy. Každá rezervace mimojiné zahrnuje důvod návštěvy včetně barevně odlišené hodnoty podle přiřazené priority. Dále rezervace zahrnuje informace o zdravotním stavu zvířete včetně případných léků. Jako poslední každá rezervace zahrnuje kontaktní informace na majitele. Všechny uvedené informace může veterinář prohlížet v rámci libovolného internetového prohlížeče formou HTML dokumentů a zároveň i ve vhodné variantě určené pro tisk ve formátou PDF.

Systém předpokládá příjem XML dokumentu vyhovující pravidlům definovaných v rámci navrženého XML schématu. Bližší specifikace zpracováváných informací je uvedená v kapitole rozebírající strukturů příchozího XML dokumentu. Samotné schéma bylo definováno pomocí *XML Schema*. Pro transformaci dat do HTML byl použit stylovací (šablonovací) jazyk *XSL (eXtensible Stylesheet Language).* K HTML byly navíc vytvořeny CSS styly. V kombinaci s formátovacími objekty (*XSL FO*) jsou nakonec data přetransformována do PDF dokumentu. Veškeré výstupy jsou generovány do adresáře `/src/ouput`, do podadresáře `/html` pro výstupy určené pro webový prohlížeč a do podadresáře `pdf` pro tisk.

Part II. Implementace

Table of Contents

Chapter 2. XML

XML (Extensible Markup Language) je standardizovaný značkovací jazyk sloužící pro výměnu dat mezi systémy. Popisuje strukturu vyměňovaného obsahu a přiřazuje význam jednotlivým hodnotám. Existuje řada technologií, které respektují syntaktická pravidla XML a dokáží například transformovat XML dokument do jiných formátů bez nutnosti externího programovacího jazyka.

XML Dokument

XML dokument `reservation.xml` je ukázkou vaidního XML souboru pro účely systému. Kořenovým elementem dokumentu je seznam rezervací `<reservation-list>` obsahující dílčí rezervace `<reservation>`. Rezervace obsahuje atribut `day`, určující den návštěvy. Každá rezervce pak obsahuje uzly `<animal>` s informacemi o zvířeti, `<owner>` s informacemi o majiteli a `<appointment>` popisující samotnou návštěvu.

Významy elementů

V rámci zvířete se evidují následující elementy:

- `identifier`: číslo čipu , může zůstat prázdný
- `species`: druh zvířete
- `breed`: plemeno zvířete
- `sex`: pohlaví zvířete
- `age`: věk zvířete
- `name`: jméno zvířete
- `medical`: informace o zdravotním stavu, dále se dělí na
  - `weight`: váha zvířete
  - `notes`: poznámky veterináře o zdravotním stavu
  - `medication`: užívané léky (nepovinný), dále se dělí na:
    - `active-ingredient`: účinná látka
    - `manufacturer`: výrobce léku
    - `dose`: dávka účinné látky

V rámci majitele se evidují elementy:

- `email`: emailová adresa
- `name`: jméno a příjmení majitele
- `address`: bydliště majitele, dále se dělí na
  - `street`: ulice
  - `streetNum`: číslo popisné
  - `city`: obec

  - `zip`: poštovní směrovací číslo
  - `country`: země
- `tel`: telefonní číslo

V rámci návštěvy se evidují elementy:

- *druh návštěvy*: jedno z následujících: `examination` prohlídka, `disease` nemoc, `injury` poranění, `surgery` chirurgický zákrok, `other` ostatní
- `day`: datum rezervace
- `time`: čas rezervace
- `duration`: předpokládané trvání návštěvy

Rezervace

Příkladem validního uzlu rezervace v XML dokumentu je následující ukázka.

```
<reservation day="mon">
    <animal>
        <identifier internal="pmgj01">940000123321456</identifier>
        <species>pes</species>
        <breed>mops</breed>
        <sex>samec</sex>
        <age>1</age>
        <name>Gyros</name>
        <medical condition="false">
            <weight unit="kg">10.3</weight>
            <notes>
                <p>Zdravý, váha v normě, bez problémů,
                seklina do duhovky dex. oka zhojena</p>
            </notes>
        </medical>
    </animal>
    <owner>
        <email>jarka@seznam.cz</email>
        <name>Jaroslava Kaňková</name>
        <address>
            <street>Linecká</street> <streetNum>7/2144</streetNum>
            <city>Český Krumlov</city>
            <zip>38101</zip>
            <country>Czechia</country>
        </address>
        <tel>+42072049229</tel>
    </owner>
    <appointment priority="normální">
        <examination>Preventivní prohlídka a odčervení</examination>
        <day>2019-12-16</day>
        <time>09:30:00</time>
        <duration unit="min">15</duration>
    </appointment>
</reservation>
```

Chapter 3. Schéma

XML Schéma je standardizovaný XML formát pro popis schémat. Pomocí struktur diskutovaného jazyka byly definovány pravidla pro validní XML dokument. Standard nabízí množinu běžných datových typů jako např. `string`, `integer` nebo `date` a mnoho dalších. Ty umožňuje dále zužovat podmínkami dle libosti. XML schéma rozlišuje mezi jednoduchými a komplexními typy. Jednoduchý typem může být element bez atributu a bez vnořeného elementu nebo samotný atribut. Jestliže element porušuje jednu z těchto podmínek, jedná se o komplexní typ.

Validační soubor

Definice schéma je obsažena v souboru `reservation.xsd`. Schéma definuje jmenný prostor `urn:x-zemanec:schemas:reservation:1.0`. Ten je definován v kořenovém elementu `<xs:schema>`, který obsahuje definice validních typů, jejichž významy byly popsány v předchozí kapitole.

Jednoduché typy

Schéma obsahuje celkem 44 jednoduchých typů. Pravidla se pro validní jednoduchý typ určují pomocí restrikce a sice vypsáním diskrétního počtu hodnot pomocí `enumeration`, regulárního výrazu pomocí `pattern` nebo maximální a minimální hodnoty u číselných hodnot. Příkladem jednoduchého typu je atribut `day` na elementu `<reservation>`.

```
<xs:simpleType name="dayType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="mon"/>
        <xs:enumeration value="tue"/>
        <xs:enumeration value="wed"/>
        <xs:enumeration value="thu"/>
        <xs:enumeration value="fri"/>
    </xs:restriction>
</xs:simpleType>
```

V rámci restrikce je definováno 5 explicitních validních textových řetězců, reprezentující dny v týdnu.

Komplexní typy

Dále schéma obsahuje celkem 28 komplexních typů. Následující konstrukce je příkladem komplexního typu definující typ návštěvy

```
<xs:complexType name="appointmentType">
    <xs:sequence>
        <xs:choice>
            <xs:element name="examination" type="xs:string"/>
            <xs:element name="disease" type="xs:string"/>
            <xs:element name="injury" type="xs:string"/>
            <xs:element name="surgery" type="surgeryType"/>
            <xs:element name="other" type="xs:string"/>
        </xs:choice>
```

```
        <xs:element name="day" type="xs:date"/>
        <xs:element name="time" type="xs:time"/>
        <xs:element name="duration" type="durationType"/>
    </xs:sequence>
    <xs:attribute name="priority" type="priorityType" />
</xs:complexType>
```

Definice typu pro návštěvu konstrukcí `xs:sequence` určuje povinné pořadí 4 definovaných potomků. První potomek je specifický tím, že může být jedním z výčtu elementů v rámci `xs:choice`. Na konci konstrukce element `xs:attribute` říká, že element musí obsahovat atribut `priority` ve tvaru definovaném v typu `priorityType`

Při konstrukci schématu byla využita metoda *slepého benátčana*, která je oproti metodě *salámových koleček* a *matrióška* nejpracnější, ale zároveň kombinuje výhody obou přístupů. Proto se jedná o nejlepší metodu pro definici schémat.

Chapter 4. XSLT

*XSLT (XML Transformation)*je šablonovací jazyk popisující pravidla transformace skrze XSLT procesor do požadovaného výstupního souboru (XML, HTML, XHTML apod.). Využívá se při tom mimojiné šablonovacích konstrukcí, podmínek pro výpis uzlů a zápis pravidel pomocí XML má objektový nádech. K průchodu, výpisu nebo např. seskupování uzlů XML dokumentu se v XSLT stylu používá jazyk *Xpath*.

Výstup do HTML

Šablona pro výstup do HTML je definována v souboru `reservation.xsl`. Pomocí konstrukce `xsl:result-document` je docíleno generování HTML souborů do několika navzájem provázaných stránek. Celkem jsou generovány 4 druhy stránek:

- **Přehled rezervací**: seznam jednotlivých rezervací
- **Seznam rezervací**: výpis rezervací k určitému dni
- **Zvíře**: profil zvířete s jeho informacemi
- **Majitel**: profil majitele s kontaktními informacemi

Mimo úvodní přehled obsahují všechny HTML soubory na konci stránky jednoduchou navigaci. První odkaz uživatele navrátí vždy na Přehled rezervací. Druhý odkaz využívá History API prohlížeče a pomocí JavaScriptu vrací uživatele vždy na předchozí stránku.

Zajímavé XSLT konstrukce

Všem šablonám je přiřazen mód pomocí atributu `mode`, kterým je určeno jakou šablonu použít v případě konfliktu názvů. Následující kód je příkladem šablony, konkrétně šablony pro úvodní přehled rezervací.

```
<xsl:template match="reservation-list" mode="list">
    <h1>Přehled rezervací ve dnech</h1>
    <section class="box">
        <table>
            <tr>
                <th>Den</th>
                <th>Datum</th>
                <th>Počet rezervací</th>
            </tr>
            <xsl:for-each-group select="reservation"
            group-by="appointment/day">
                <xsl:sort select="appointment/day"/>
            <tr>
                <td>
                    <a class="name"
                    href="overview_{generate-id(.)}.html">
                        <xsl:if test=".[@day = 'mon']">
                            <xsl:text>pondělí </xsl:text>
                        </xsl:if>
                        <xsl:if test=".[@day = 'tue']">
                            <xsl:text>úterý </xsl:text>
                        </xsl:if>
```

```
                        <xsl:if test=".[@day = 'wed']">
                            <xsl:text>středa </xsl:text>
                        </xsl:if>
                        <xsl:if test=".[@day = 'thu']">
                            <xsl:text>čtvrtek </xsl:text>
                        </xsl:if>
                        <xsl:if test=".[@day = 'fri']">
                            <xsl:text>pátek </xsl:text>
                        </xsl:if>
                    </a>
                </td>
                <td>
                   <xsl:value-of
                     select="format-date(current-grouping-key(),
                     '[D]. [MNn] [Y]', 'cs', 'AD', 'GE')"
                    />
                </td>
                <td>
                    <xsl:value-of select="count(current-group())"/>
                </td>
            </tr>
            </xsl:for-each-group>
        </table>
    </section>
</xsl:template>
```

Šablona prochází XML dokumentem a seskupuje všechny rezervace na základě výsledku Xpath dotazu na uzel pro datum návštěvy. Pomocí `xsl:sort` jednotlivé skupiny seřazuje podle data návštěvy od nejstaršího. Pro každou skupinu na konkrétní den je generován odkaz na rezervace pro tento den do tabulky. Do druhého sloupce se vybírá datum datum, které se prostřednictvím xpath funkce `format-date()` automaticky formátuje do českého formátu gregoriánského kalendáře. Do posledního sloupce tabulky se generuje číslo reprezentující počet rezervací na danný den. To je docíleno Xpath funkcí `current-group()` uvnitř funkce `count()`.

CSS

Kaskádové styly (Cascading Stylesheets) definují vzhled obsahu HTML stránek. Oděluje se tak vzhled od obsahu. Narozdíl od XSL nejsou primárně určeny na přidávání obsahu. Přesto v CSS 3 lze v omezené míře pseudoelementy obsah do HTML přidávat.

Příkladem přidání vizuálního obsahu je následující ukázka použitá v rámci projektu na oddělování jednotlivých rezervací.

```
.box:not(:last-child)::after{
    content: "";
    position: absolute;
    bottom: -22px;
    left: 50%;
    transform: translateX(-50%);
    width: 40px;
    height: 5px;
    border-radius: 10px;
    background: rgb(255, 255, 255);
}
```

Pravidlo vybírá všechny třídy `box`, které nejsou posledním potomkem. Pseudoelementem `::after` pravidlo přidává absolutně napozicovaný obdélník, který je kombinací funkce `translateX(-50%)` a vlastnosti `left: 50%;` vycentrován do středu relativně napozicovaného rodiče.

Chapter 5. Formátovací Objekty

XSL FO (XSL Formatting Objects) je hojně využívaný standard pro transformaci XML dokumentů do elektronických i tištěných formátů jako PDF nebo EPUB. Formátovací objekty lze chápat jako zjednodušená obdoba CSS. Narozdíl od výstupu do HTML se výstup do PDF provádí ve 2 krocích. V prvním kroku se pomocí XSL dokument přetransformuje do podoby s formátovacími objekty. V druhém kroku probíhá samotné generování souboru určenému k tisku.

Souborem který transformuje XML do FO je `reservationFO.xsl` Ačkoliv konstrukce formátovacích objektů se pojmenováním liší od elementů používaných v HTML, vypisování hodnot pomocí XSL zůstává stejné. Proto lze s lehkými modifikacemi využít konstrukcí použitých pro generování HTML souborů. Výstup do PDF je stejně jako v HTML prolinkován odkazy.