

Webový repozitár pre vývojové artefakty

Autori: Peter Kováč, Peter Lipčák, Jakub Kadaši

Vedúci tímu: Peter Lipčák

Vedúci: Filip Nguyen

PB138

2015

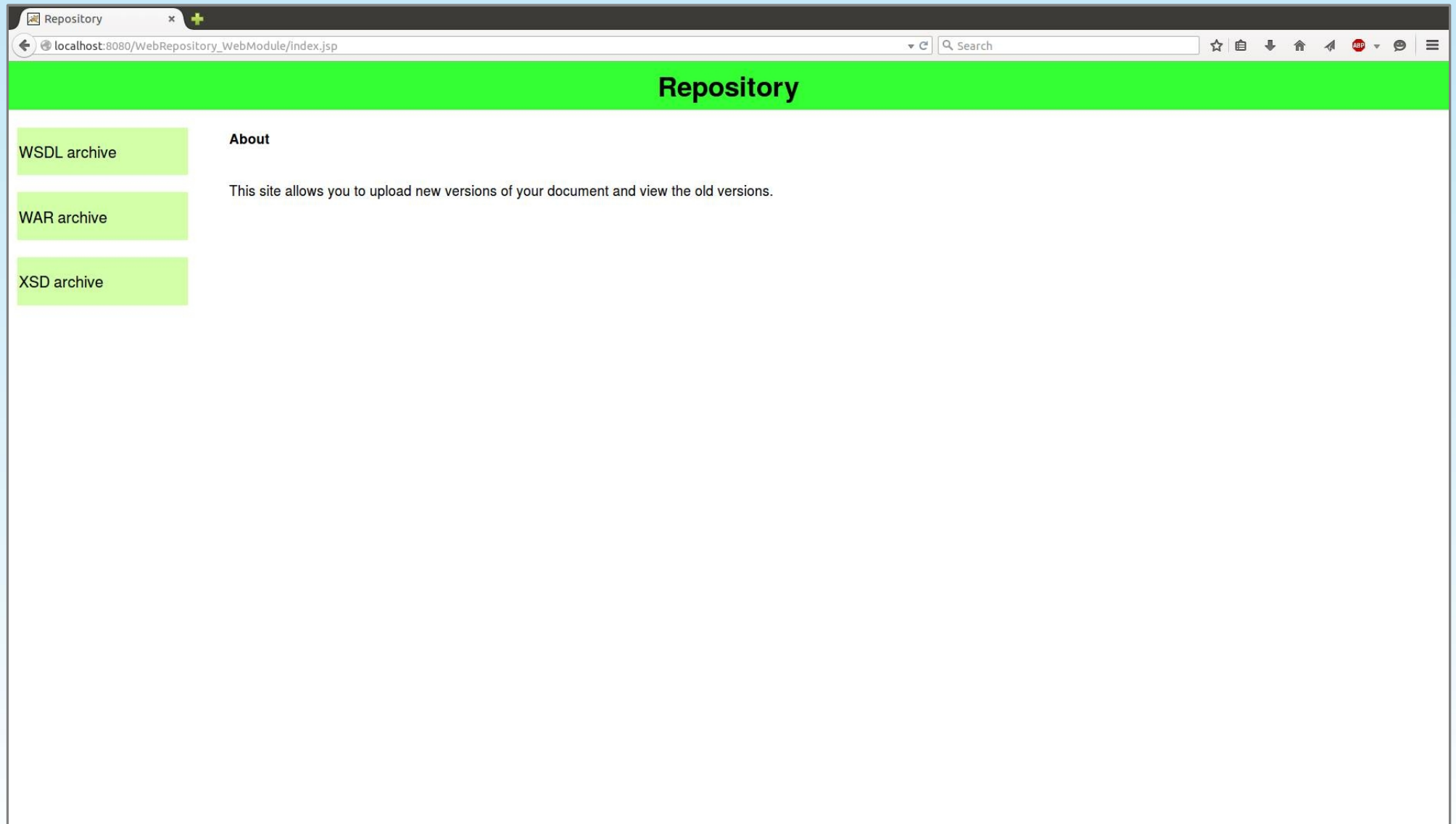
Organizácia projektu

- rozdelenie úloh:
 - Jakub Kadaši - parsovanie metadát z dokumentov
 - Peter Kováč - managers, práca s databázou
 - Peter Lipčák - ActionBeans, jsp pages
 - Roman Koníček – vypadol z tímu
- využitie GitHub
- dokumentácia pomocou JavaDoc, Wiki na GH

Repozitár

- webová aplikácia
- uchovávanie vývojárskych artefaktov a ich nahliadanie
- po nahraní sú extrahované a uložené metadáta
- vyhľadávanie pomocou metadát

Repozitár



Repozitár

WAR archives

localhost:8080/WebRepository_WebModule/showXSD.jsp

Search

XSD archive

WSDL archive

WAR archive

XSD archive

Search:

Search

☐ All ☐ Simple types ☐ Complex types

Browse... No file selected.

Submit

Name	Version	Insight
contacts.xsd	2015/06/08 14:11:52	View
compact.xsd	2015/06/08 14:11:56	View
test.xsd	2015/06/08 14:11:59	View
anyElementDemo.xsd	2015/06/08 14:12:03	View
contacts.xsd	2015/06/08 14:12:08	View
formAttribute.xsd	2015/06/08 14:12:12	View
contacts.xsd	2015/06/08 14:12:17	View
company.xsd	2015/06/08 14:21:59	View

Repozitár

WAR archives

localhost:8080/WebRepository_WebModule/showXSD.jsp

Search

XSD archive

WSDL archive

WAR archive

XSD archive

Search:

Search

☐ All ☐ Simple types ☐ Complex types

Browse...

No file selected.

Submit

Name	Version	Insight
contacts.xsd	2015/06/08 14:11:52	<div>View</div>
compact.xsd	2015/06/08 14:11:56	<div>View</div>
test.xsd	2015/06/08 14:11:59	<div>View</div>
anyElementDemo.xsd	2015/06/08 14:12:03	<div>View</div>
contacts.xsd	2015/06/08 14:12:08	<div>View</div>
formAttribute.xsd	2015/06/08 14:12:12	<div>View</div>
contacts.xsd	2015/06/08 14:12:17	<div>View</div>
company.xsd	2015/06/08 14:21:59	<div>View</div>

Repozitár

Show single XSD sch... x

localhost:8080/WebRepository_WebModule/xsd

Q Search

☆

📁

⬇

🏠

🔍

🔴

🗨

☰

XSD archive

WSDL archive

WAR archive

XSD archive

File: company.xsd
Date: 2015/06/08 14:21:59

XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="company" type="companyType">
    <xs:key name="didKey">
      <xs:selector xpath="division"/>
      <xs:field xpath="@did"/>
    </xs:key>
    <xs:key name="pidKey">
      <xs:selector xpath="./division/head/person"/>
      <xs:field xpath="@pid"/>
    </xs:key>
    <xs:key name="employeeIdUnique">
      <xs:selector xpath="./division/employees/employee"/>
      <xs:field xpath="@eid"/>
    </xs:key>
  </xs:element>
  <xs:complexType name="companyType">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="division" type="divisionType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="divisionType">
    <xs:all>
      <xs:element maxOccurs="1" minOccurs="0" name="name" type="xs:string"/>
      <xs:element maxOccurs="1" minOccurs="0" name="head" type="headType"/>
      <xs:element maxOccurs="1" minOccurs="0" name="employees" type="employeesType"/>
    </xs:all>
    <xs:attribute name="did" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```

Complex and simple types

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata date="2015/06/08 14:21:59" fileName="company.xsd" xsd_id="7">
  <simple_complex_types>
    <xs:complexType name="companyType">
      <xs:element maxOccurs="unbounded" name="division" type="divisionType"/>
    </xs:complexType>
    <xs:complexType name="divisionType">
      <xs:element maxOccurs="1" minOccurs="0" name="name" type="xs:string"/>
      <xs:element maxOccurs="1" minOccurs="0" name="head" type="headType"/>
    </xs:complexType>
  </simple_complex_types>
```

Frontend

- vyhľadávanie v artefaktoch

```
public Resolution searchData() {  
    String searchData = context.getRequest().getParameter("dataInput");  
    String radioSelected = context.getRequest().getParameter("radioButton");  
    if(radioSelected == null) return new ForwardResolution("/wrongSearch.jsp");  
    resultList = manager.findXSDByData(searchData, radioSelected);  
    return new ForwardResolution("/showSearchXSD.jsp");  
}
```

```
<d:form beanclass="org.cz.muni.fi.pb138.Bean.XSDActionBean" method="GET">  
    <tr>  
        <th>Search: </th>  
        <th><d:text id="data" name="dataInput"/></th>  
        <th><d:submit name="searchData">Search</d:submit></th>  
        <th><d:radio name="radioButton" id="radioAll" value="xs:simpleType | $meta//xs:complexType" checked="checked"/>All</th>  
        <th><d:radio name="radioButton" id="radioSimpleType" value="xs:simpleType"/>Simple types</th>  
        <th><d:radio name="radioButton" id="radioComplexType" value="xs:complexType"/>Complex types</th>  
    </tr>  
</d:form>
```


Frontend

- nahrávanie artefaktov

```
<d:form action="xsdUpload" method="POST" beanclass="org.cz.muni.fi.pb138.Bean.XSDActionBean" >
  <d:file name="xsdInput"/>
  </br>
  <d:submit name="xsdUpload">Submit</d:submit>
</d:form>
```

```
public Resolution xsdUpload() {
    try {
        File toFile = new File(System.getProperty("user.home")+File.separator+xsdInput.getFileName());
        String extension = xsdInput.getFileName().substring(xsdInput.getFileName().lastIndexOf(".") + 1, xsdInput.getFileName().length());
        if(!extension.equals("xsd")) {
            toFile.delete();
            xsdInput.delete();
            return new ForwardResolution("/wrongFile.jsp");
        }
        xsdInput.save(toFile);
        String content = Util.readFile(toFile);

        XSD xsd = new XSD();
        xsd.setId(manager.getNewId());
        xsd.setTimestamp(Util.getTimeStamp());
        xsd.setFileName(xsdInput.getFileName());
        xsd.setDocument(Util.stripXMLHeader(content));
        xsd.setExtract(Util.docToString(xsdParser.xsdExtract(Util.stringToDoc(content))));
        manager.createXSD(xsd);

        toFile.delete();
        xsdInput.delete();
    } catch (IOException ex) {
        Logger.getLogger(XSDActionBean.class.getName()).log(Level.SEVERE, null, ex);
    } catch (NullPointerException ex) {
        return new ForwardResolution("/wrongFile.jsp");
    }
    return new ForwardResolution("/showXSD.jsp");
}
```

Typy artefaktov

- umožňuje nahrávať artefakty: XSD, WSDL, WAR archívy
- ak sa jedná o rovnaký artefakt, tak je nahraná jeho nová verzia

WAR archív

- vytiahne sa web.xml
- vyextrahuje sa zoznam listenerov, filtrov a servletov

```
<?xml version="1.0" encoding="UTF-8"?>
<war id="0">
  <web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http
  <filter>
    <description>dynamicicky mapuje URL na ActionBeans</description>
    <display-name>Stripes Dynamic Mapping Filter</display-name>
    <filter-name>DynamicMappingFilter</filter-name>
    <filter-class>net.sourceforge.stripes.controller.DynamicMappingFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>DynamicMappingFilter</filter-name>
    <url-pattern>*/</url-pattern>
    <dispatcher>REQUEST</dispatcher>
    <dispatcher>FORWARD</dispatcher>
    <dispatcher>INCLUDE</dispatcher>
  </filter-mapping>
  <servlet>
    <servlet-name>StripesDispatcher</servlet-name>
    <servlet-class>net.sourceforge.stripes.controller.DispatcherServlet</servl
    <load-on-startup>1</load-on-startup>
  </servlet>
  <session-config>
    <session-timeout>30</session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
</war>
```

```
NodeList filterList = doc.getElementsByTagName("filter");

for (int i = 0; i < filterList.getLength(); i++) {
    if (filterList.item(i) instanceof Element) {
        Element operationElement = (Element) filterList.item(i);
        Node nodeToMove = output.importNode(operationElement, true);
        root.appendChild(nodeToMove);
    }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata date="2015/06/08 13:43:58"
  fileName="WebRepository-1.0-SNAPSHOT.war" war_id="0">
  <filters_servlets_and_listeners>
    <filter>
      <description>dynamicicky mapuje URL na ActionBeans</description>
      <display-name>Stripes Dynamic Mapping Filter</display-name>
      <filter-name>DynamicMappingFilter</filter-name>
      <filter-class>net.sourceforge.stripes.controller.DynamicMappingFilter</filter-class>
    </filter>
    <servlet>
      <servlet-name>StripesDispatcher</servlet-name>
      <servlet-class>net.sourceforge.stripes.controller.DispatcherServlet</servlet-class>
      <load-on-startup>1</load-on-startup>
    </servlet>
  </filters_servlets_and_listeners>
</metadata>
```

XSD schéma

- vyextrahuje sa zoznam typov (simple a complex) a zoznam elementov a atribútov

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="company" type="companyType">
    <xs:key name="didKey">
      <xs:selector xpath="division"/>
      <xs:field xpath="@did"/>
    </xs:key>
    <xs:key name="pidKey">
      <xs:selector xpath="./division/head/person"/>
      <xs:field xpath="@pid"/>
    </xs:key>
    <xs:key name="employeeIdUnique">
      <xs:selector xpath="./division/employees/employee"/>
      <xs:field xpath="@eid"/>
    </xs:key>
  </xs:element>
  <xs:complexType name="companyType">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="division" type="divisionType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="divisionType">
    <xs:all>
      <xs:element maxOccurs="1" minOccurs="0" name="name" type="xs:string"/>
      <xs:element maxOccurs="1" minOccurs="0" name="head" type="headType"/>
      <xs:element maxOccurs="1" minOccurs="0" name="employees" type="employeesType"/>
    </xs:all>
    <xs:attribute name="did" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```

```
NodeList simpleTypeList = doc.getElementsByTagName(namespace+"simpleType");

for (int i = 0; i < simpleTypeList.getLength(); i++) {
    if (simpleTypeList.item(i) instanceof Element) {
        Element simpleElement = (Element) simpleTypeList.item(i);
        Node nodeToMove = output.importNode(simpleElement, false);
        root.appendChild(nodeToMove);
    }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata date="2015/06/08 13:59:23" fileName="company.xsd" xsd_id="2">
  <simple_complex_types>
    <xs:complexType name="companyType">
      <xs:element maxOccurs="unbounded" name="division" type="divisionType"/>
    </xs:complexType>
    <xs:complexType name="divisionType">
      <xs:element maxOccurs="1" minOccurs="0" name="name" type="xs:string"/>
      <xs:element maxOccurs="1" minOccurs="0" name="head" type="headType"/>
      <xs:element maxOccurs="1" minOccurs="0" name="employees" type="employeesType"/>
      <xs:attribute name="did" type="xs:string" use="required"/>
    </xs:complexType>
  </simple_complex_types>
</metadata>
```


WSDL dokumenty

- vyextrahuje sa zoznam operácií spolu s informáciou o request a response správách

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl id="3">
  <definitions name="EndorsementSearch"
    targetNamespace="http://namespaces.snowboard-info.com"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:es="http://www.snowboard-info.com/EndorsementSearch.wsdl"
    xmlns:esxsd="http://schemas.snowboard-info.com/EndorsementSearch.xsd" xmlns:s
  <message name="GetEndorsingBoarderRequest">
    <part element="esxsd:GetEndorsingBoarder" name="body"/>
  </message>
  <message name="GetEndorsingBoarderResponse">
    <part element="esxsd:GetEndorsingBoarderResponse" name="body"/>
  </message>
  <portType name="GetEndorsingBoarderPortType">
    <operation name="GetEndorsingBoarder">
      <input message="es:GetEndorsingBoarderRequest"/>
      <output message="es:GetEndorsingBoarderResponse"/>
      <fault message="es:GetEndorsingBoarderFault"/>
    </operation>
  </portType>
  <binding name="EndorsementSearchSoapBinding" type="es:GetEndorsingBoarderF
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap
  <operation name="GetEndorsingBoarder">
    <input>
      <soap:body
        namespace="http://schemas.snowboard-info.com/EndorsementSearch.xsd
      </input>
    <output>
      <soap:body
        namespace="http://schemas.snowboard-info.com/EndorsementSearch.xsd
      </output>
    </operation>
  </binding>
</definitions>
</wsdl>
```

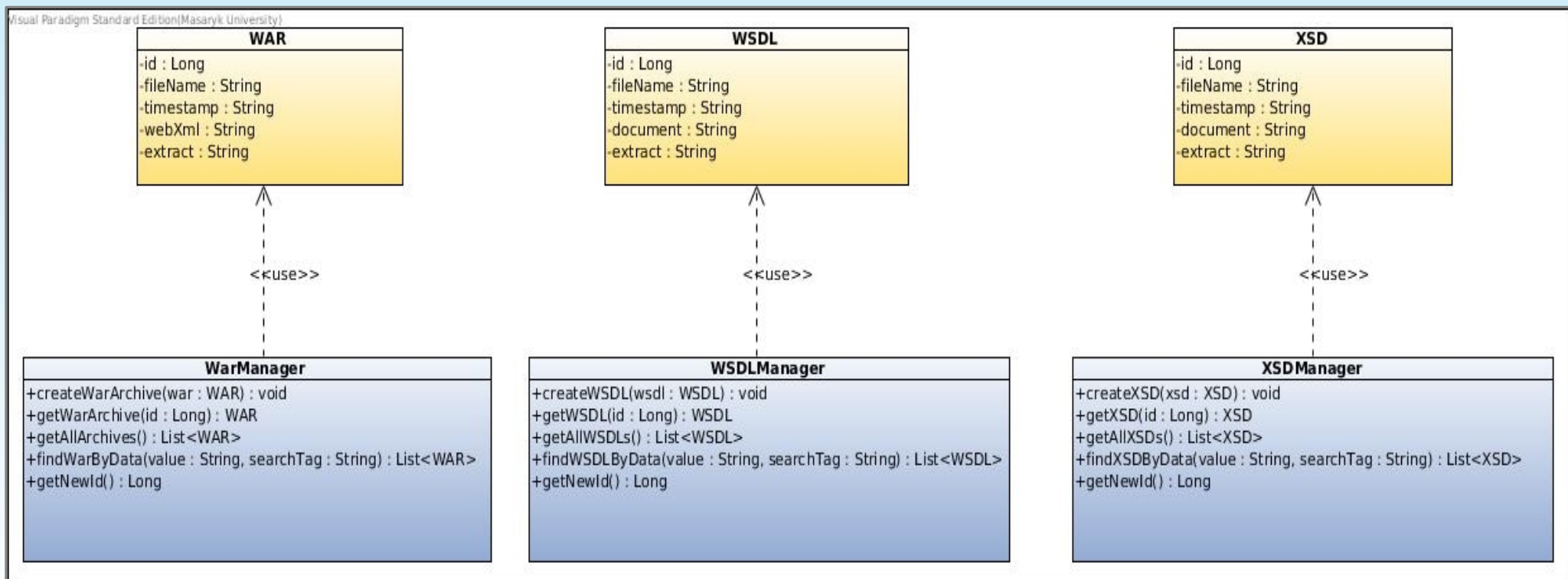
```
NodeList operationList = doc.getElementsByTagName("operation");
NodeList messageList = doc.getElementsByTagName("message");

for (int i = 0; i < operationList.getLength(); i++) {
  if (operationList.item(i) instanceof Element) {
    Element operationElement = (Element) operationList.item(i);
    Node nodeToMove = output.importNode(operationElement, true);
    root.appendChild(nodeToMove);
  }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata date="2015/06/08 14:04:24" fileName="skuska.wsdl" wsdl_id="3">
  <operations_and_messages>
    <operation name="GetEndorsingBoarder">
      <input message="es:GetEndorsingBoarderRequest"/>
      <output message="es:GetEndorsingBoarderResponse"/>
      <fault message="es:GetEndorsingBoarderFault"/>
    </operation>
    <operation name="GetEndorsingBoarder">
      <input>
        <soap:body
          namespace="http://schemas.snowboard-info.com/EndorsementSearch.xsd" use="literal"/>
      </input>
      <output>
        <soap:body
          namespace="http://schemas.snowboard-info.com/EndorsementSearch.xsd" use="literal"/>
      </output>
    </operation>
  </operations_and_messages>
</metadata>
```

Backend

- ukladanie dát pomocou BaseX databázy



Backend

```
public void createXSD(XSD xsd) {
    //add whole XML to database
    String wholeXml = "<xsd id='"+xsd.getId().toString()+"'>" + xsd.getDocument() + "</xsd>";
    this.dm.addXML("xsd", xsd.getId().toString(), wholeXml);

    //add METADATA for xml to database
    String metadataXml = "<metadata xsd_id='"+xsd.getId().toString()+"' date='"+xsd.getTimestamp()
        + "' fileName='"+xsd.getFileName()+"'>" + xsd.getExtract() + "</metadata>";
    this.dm.addXML("xsd-metadata", xsd.getId().toString(), metadataXml);
}
```

```
public List<XSD> findXSDByData(String s, String searchTag) {
    List<XSD> output = new ArrayList<XSD>();
    String query = this.dm.queryCollection("xsd-metadata", "declare namespace xs='http://www.w3.org/2001/XMLSchema';"
        + " distinct-values( for $meta in collection('xsd-metadata')/metadata "
        + " for $node in $meta//'" + searchTag
        + " where fn:contains($node/@name,'" + s + "') "
        + " return distinct-values($meta/@xsd_id))");

    if(query.equals("")) {
        return output;
    }
    String strarray[] = query.split(" ");
    int intarray[] = new int[strarray.length];
    for (int i=0; i < intarray.length; i++) {
        intarray[i] = Integer.parseInt(strarray[i]);
    }
    for (int x : intarray) {
        output.add(this.getXSD(new Long(x)));
    }
    return output;
}
```

```
public void addXML(String collection, String name, String xml) {
    Context context = new Context();
    try {
        new Set("dbpath", this.DBPath).execute(context);
        new Open(collection).execute(context);
        new Add(name, xml).execute(context);
    } catch (BaseXException ex) {
        logger.log(Level.SEVERE, "DBError when adding XML file");
    }
    context.close();
}
```

Ďakujeme za pozornost'