DREAM

IT Document

Implementation and Testing Document

Alessandro Cecchetto (10865806)
Mattia Siriani (10571322)
Matteo Visotto (10608623)

Computer Science and Engineering
Software Engineering 2 course @ Politecnico di Miliano

v1.1 - 11/02/2022

Contents

Information

This project has been built following the indications of the stakeholders, referring to what is published in the official repository (https://github.com/UNDP-india/Data4Policy), taking into consideration, for some aspects, also the specification provided by the teachers in charge of the course.

Document version: v1.1 - 11/02/2022

1 Introduction

1.1 Purpose

This document aims to describe how the implementation and the integration testing took place.
The implementation is the last phase of the Dream application development cycle while testing is necessary in order to check if the critical parts on the application works correctly how described in the Design Document.

Source code and releases could be found in the official project repository on GitHub (https://github.com/matteovisotto/SE2-CecchettoSirianiVisotto).

1.2 Acronyms

1.2.1 Acronyms

- **API:** Application Programming Interface.
- **HTTPS:** Hyper Text Transfer Protocol over SSL.
- **DD:** Design Document.
- **TLS:** Transport Layer Security.
- **SSL:** Secure Socket Layer.
- **DBMS:** DataBase Management System.
- **IdP:** Identity Provider.
- **SP:** Service Provider.
- **RASD:** Requirements Analysis and Specification Document.
- **ACS:** Assertion Consumer Service .
- **SAML:** Security Assertion Markup Language.
- **DOM:** Document Object Model.
- **JPA:** Java Persistance API.
- **UI:** User Interface.
- **URL:** Uniform Resource Locator.

1.3 Revision history

- v.1.0 - 06/02/2022
- v.1.1 - 11/02/2022 - Update installation instruction due to a typo.

1.4 References

- Requirements Analysis Specification Document (RASD).
- Design Document(DD).
- Specification document: "R&DD Assignment A.Y. 2021-2022".
- Official Data4Policy stakeholder's project repository: https://github.com/UNDP-india/Data4Policy.
- Shibboleth documentation: https://shibboleth.atlassian.net/wiki/home.
- Meta RWI https://dataforgood.facebook.com/dfg/tools/relative-wealth-index.
- ESRI Java geometry API: documentation: https://github.com/Esri/geometry-api-java/wiki/.
- Mailjet API documentation: https://github.com/mailjet/mailjet-apiv3-java.
- Bootsrap 5 documentation: https://getbootstrap.com/docs/5.0/getting-started/introduction/.
- jQuery documentation: https://api.jquery.com/.
- Thymeleaf documentation: https://www.thymeleaf.org/documentation.html.
- HTML documentation: https://developer.mozilla.org/en-US/docs/Web/HTML
- JavaScript documentation: https://devdocs.io/javascript/.
- MySQL documentation: https://dev.mysql.com/doc/.
- Shibboleth documentation: https://shibboleth.atlassian.net/wiki/home.
- Shibboleth IdP configuration: Github link.
- Shibboleth SP configuration: Github link.
- GeoJSON specification: https://geojson.org/.

2 Development

2.1 Implemented Functionalities

2.2 Adopted Development Frameworks

As we mentioned in the Design Document, we implemented a four-tier architecture, distinguishing the database layer, the application server with the business logic, the web servers with the styling logic and finally the client.
We managed to implement the Model-View-Controller pattern dividing the code into classes that had a specific role: the "Entity" classes act as Model of the pattern, reproducing the tables in the database; the Controller was implemented by means of the "Controller" and "Service" classes: the former manage the business logic while the latter manage the data from the databases. The View was given by the "Bean" classes, which represent the data provided to the front-end, and the "Views".

2.3 Programming languages

We adopted JAVA for the back-end and JavaScript for the front-end. JAVA represent a standard in the industry and presents many features that make it the perfect language for our purpose: since it is a compiled language, the elaborations have very good performances, and the fact that it is object-oriented allows to create modular programs and use the same code in different part of the project. JAVA is also a mature language used by millions of developers during the last decades, which means that a lot of documentation was written and it is more predictable and stable. JavaScript is a text-based programming language used both on the server-side and the client-side that allows to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages engaging the user.

2.4 Java Frameworks

The Java EE stands for Java Enterprise Edition, which was earlier known as J2EE. It is a set of specifications wrapping around Java SE (Standard Edition). The Java EE provides a platform for developers with enterprise features such as distributed computing and web services. Java EE applications are usually run on reference run times such as microservers or application servers. Examples of some contexts where Java EE is used are e-commerce, accounting, banking information systems. Java EE has several specifications which are useful in making web pages, reading and writing from database in a transactional way, managing distributed queues. The Java EE contains several APIs which have the functionalities of base Java SE APIs such as Enterprise JavaBeans, connectors, Servlets, Java Server Pages and several web service technologies. It provides also user authentication unlike the Java SE. We managed to organize our architecture in the following way:

- **API** : They provide the access point to the functionalities, routing the incoming request to the correct controller;
- **Bean** : They are the object used to interact with the user and serve also as data transfer object (DTO). They are encoded in JSON format when sent to or received by the client.
- **Controller** : The controllers implement the methods used by the APIs. They manipulate the Beans and use the Service and the Mapper;
- **Service** : They act as an intermediate between the Entities and the Controllers. They contains methods to fetch data from the database and useful methods to manage data received by the Controller.
- **Filter** : They are used to filter the incoming requests. They consent to authorize the use of methods provided by the Controllers only to user that present special characteristics (ex. Policy maker use some methods that are forbidden to the user);
- **Mapper** : They are classes whose purpose is to convert the Bean objects into Entity objects and vice versa;
- **Entity** : They represent database object. They are declared with @Entity annotation, which maps the object to a database table. Inside an entity object it is possible to specify constraints and foreign references through proper annotations.

2.5 Other Frameworks

2.5.1 jQuery

jQuery is a fast, small and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation and Ajax much simpler with an easy to use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery consent to write JavaScript in a easier way.
We used jQuery to implement the network layer of the client-side web application: it is responsible of the comunication with the server and menages the requests and the responses.

2.5.2 Bootstrap

Bootstrap is one of the most popular CSS framework, considered a De Facto standard for developing Web interfaces finding application in areas such as the creation of pre-packaged HTML templates and themes for major CMS (Content Managing System), especially in a responsive perspective.

2.6 API Integration

Shibboleth is a Single-Sign-On (SSO) system that provides functionalities for identity management and federated authentication. Shibboleth consortion provides different application like Identity Provider, Service Provider and Embedded Discovery Service. In the forum application users can login by means of an identity provider. To do this, a software that is able to receive and verify authentication assertion is required.
We have decided to use Shibboleth-SP as a module of the Apache Webserver in our frontend for the following reasons:

- **Resource protection:** Shibboleth let us to protect specific set of URL paths requiring a specific session generated by the service provider module. In our case the protected path is /dream/login which is automatically handled be the module. When a user enter that resource if a SP session does not exists the user is redirected to the IdP to complete the login flow. Instead, if a valid session is present the user attribute are available in a global variable which are read and verified again by the DreamLoginServlet of the application (the connection between Apache and TomEE is performed by AJP Proxy connector).
- **Embedded Discovery Service:** Using this configuration it is possible to use also the Embedded Discovery Service module, an extension of the service provider one that let network administrator to configure multiple IdPs for the same SP. If applied, users from different identity providers can login in the forum application without any modification of the application itself.

2.6.1 MailJet

Mailjet is an email marketing tool that allows to send out both marketing emails (eg. newsletters) and transactional emails (eg. order confirmation & registration confirmation). The platform uses the external service to send notification mail when the followed discussion are updated and when the Policy maker accept or decline the publication of a post in the forum.

2.6.2 Esri Geometry

The Esri Geometry API for Java enables developers to write custom applications for analysis of spatial data. Its methods can be used to retrieve geometry objects like polygons and points from GEOJson and to test their topological relation.

3 Source Code

3.1 Backend

3.1.1 Packages

Package it.dreamplatform.forum and it.dreamplatform.data
The packages are organized as specified in section 2.4. There are also additional packages like servlet and utils.

- **API** : They provide the access point to the functionalities, routing the incoming request to the correct controller;
- **Bean** : They are the object used to interact with the user and serve also as data transfer object (DTO). They are encoded in JSON format when sent to or received by the client.
- **Controller** : The controllers implement the methods used by the APIs. They manipulate the Beans and use the Service and the Mapper;
- **Service** : They act as an intermediate between the Entities and the Controllers. They contains methods to fetch data from the database and useful methods to manage data received by the Controller.
- **Filter** : They are used to filter the incoming requests. They consent to authorize the use of methods provided by the Controllers only to user that present special characteristics (ex. Policy maker use some methods that are forbidden to the user);
- **Mapper** : They are classes whose purpose is to convert the Bean objects into Entity objects and vice versa;
- **Entity** : They represent database object. They are declared with @Entity annotation, which maps the object to a database table. Inside an entity object it is possible to specify constraints and foreign references through proper annotations.
- **Servlet** : They are Java programming language class that are used to extend the capabilities of servers that host applications accessed by means of a request-response programming model. They are used to handle the request obtained from the webserver, process the request, produce the response, then send a response back to the webserver.
- **Utils** : In this folder there are two main categories of class: the enumerations (lists of named constants) and the "util" class. Those utility classes contains methods that can be re-used. These methods are invoked inside the Controllers.

3.2 Front-end Web Application

The front-end web application is contained in webapp folder divided into *static* folder which contains all the static resources (assets, javascript and css files) while templates are located into *templates* folder.
Webpages are managed by Thymeleaf template manager that loads pages dynamically for the specific servlet (see package Servlet above). The main layout and the logged user information is loaded through the template manager while the content is loaded client side using the REST Api provided by the application.

3.3 Further implementations

In order to respect all the indications given in the RASD and DD document, the platform need to include the administrator area with all the methods related and the scheduled process to calculate the deviance asynchronously. Also the attachment are not included in the forum side.

4 Testing

In this section we will briefly describe how we tested the application following the general guideline given in the Design Document.
We decide to test only the backend because the frontend just display the data received from the backend.

4.1 Integration tests

- **DiscussionIntegrationTest:** we tested the discussion controller methods. First of all we created a new discussion with a post, then we deleted another one and tried to delete a discussion not present and finally tried to update another discussion. We tested also the methods that fetch the discussions, searching for the the discussion published by a policy maker (also for a policy maker that does not have publish any discussion yet), the most active discussions in the forum and we implemented a test to retrieve the users that follows a discussion.

- **PostIntegrationTest:** we tested the methods linked to the posts. We tried to insert a new post using a Policy maker account and a user account, then we tried to delete and update them, we tested the approve and the decline pending post methods and we repeat the same tests with non valid input or authorizations to see if the error would be catched.

- **UserIntegrationTest:** we tested the user controller methods. First of all, we created a new Policy maker account and one for a user, then we tried to search an account not present in the database and after that we updated a User account. Eventually we tested the method to retrieve the most active users.

- **DataIntegrationTest:** we tested the data related methods. We tried to fetch data for a not existing district, then we created rankings for different districts and retrieved all the data available for a single district. We fetched data from a given data set and finally we calculated the ranking list for a set of districts selecting the data sources of interest.

4.2 Testing results

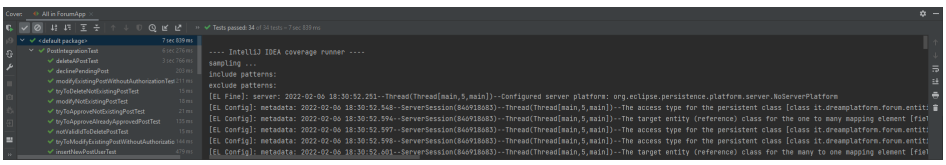

Figure 1: Forum testing results

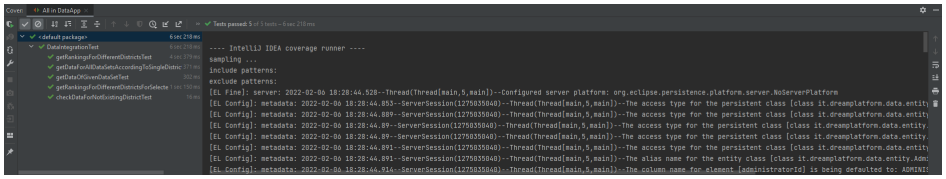

Figure 2: Data testing results

4.3 Testing coverage

Element	Class, %	Method, %	Line, %
api	0% (0/5)	0% (0/23)	0% (0/80)
bean	71% (5/7)	75% (47/62)	77% (52/67)
controller	80% (4/5)	76% (23/30)	81% (143/1...
entities	100% (4/4)	83% (41/49)	83% (45/54)
filters	0% (0/4)	0% (0/7)	0% (0/39)
mapper	75% (3/4)	51% (22/43)	53% (82/153)
services	100% (4/4)	71% (25/35)	70% (74/105)
servlet	0% (0/15)	0% (0/44)	0% (0/276)
utils	0% (0/2)	0% (0/8)	0% (0/19)

Figure 3: Forum testing coverage

Element	Class, %	Method, %	Line, %
api	0% (0/2)	0% (0/5)	0% (0/21)
bean	80% (4/5)	57% (37/64)	60% (41/68)
controller	100% (1/1)	100% (8/8)	100% (55/55)
entity	100% (5/5)	22% (9/40)	31% (14/45)
filter	0% (0/2)	0% (0/3)	0% (0/23)
mapper	100% (1/1)	50% (2/4)	52% (11/21)
service	66% (2/3)	66% (4/6)	81% (9/11)
servlet	0% (0/9)	0% (0/26)	0% (0/106)
utils	66% (2/3)	68% (11/16)	73% (135/183)

Figure 4: Data testing coverage

5 Build

5.0.1 Requirements

- Java SE JDK 17 (OracleJDK, OpenJDK)
- Maven framework version 3.0 (or newer)

5.1 Windows

5.1.1 Installing Java

Download OracleJDK or OpenJDK.
Extract content from the zip folder to your preferred location, then go to

`Start>Edit the system environment variables>Environment Variables`

In the User Section select Path variable and click on Edit; select New and type `C:\Users\<your-user>\<path-to-extracted-folder>\bin`, then save and exit. Open `Start>cmd.exe` and verify your Java version by typing the following command:

```
java -version
```

5.1.2 Installing Maven

Download Maven.
Extract to preferred location and repeat above steps.
Verify your Maven version by typing the following command:

```
mvn --version
```

5.2 Linux

If you want to download Maven and JavaJDK with your package manager be sure to fulfill the system requirements otherwise follow the next steps.

5.2.1 Installing Java

Download JavaJDK based on your distro or OpenJDK 17 from official websites.
Navigate to your preferred install location and extract the .tar.gz archive using:

```
tar zxvf jdk-17.<version-number>-x64_bin.tar.gz
```

Now let's set the PATH variable by typing the following code:

```
cd $HOME
nano .bashrc
```

Add the following line to the end of .bashrc:

```
export PATH=/<path-to-extracted-folder/bin:$PATH
```

Verify your Java version by typing the following command:

```
java -version
```

Debian-Based Linux Platforms

Type:

```
sudo apt install /path/to/package/name.deb
```

Verify your Java version by typing the following command:

```
java -version
```

RPM-Based Linux Platforms

Type the following command to install the package:

```
rpm -ivh jdk-17.<version-number>-x64_bin.rpm
```

Upgrade the package using the following command:

```
rpm -Uvh jdk-17.<version-number>-x64_bin.rpm
```

You can now delete the .rpm file and verify your installation by typing:

```
java -version
```

5.2.2 Installing Maven

Download Maven, extract it with:

```
tar zxvf maven.<version-number>-x64_bin.tar.gz
```

Set PATH variable by appending this command to .bashrc as previous steps:

```
cd $HOME
nano .bashrc
export PATH=/<path-to-extracted-folder/bin:$PATH
```

5.3 MacOS

5.3.1 Installing Java

Download JavaJDK 17, double-click on .dmg file and Install it. Verify your Java version by typing the following command:

```
java -version
```

5.3.2 Installing Maven

Follow Linux installation steps.

5.4 Compiling the source files

Open the terminal and navigate to the root application folder (containing pom.xml). Type

```
mvn package -Dmaven.test.skip=true
```

and hit Enter.

5.5 Run tests

In order to run tests move to the home directory of the project (ForumApp ot DataApp) and run

```
mvn test
```

6 Installation and Configuration

Follow this part of the guide after you have compiled war files or you are using provided one. If you do not want to configure Shibboleth skip all the related sections and also Apache Web Server.

6.1 TomEE Application Server

Dream applications work inside a Java Application Server which can be downloaded directly from the official website (Download TomEE Plume 8, version 9 is not supported). It also requires mySQL or MariaDB for the data layer (standard official installation is sufficient)
Once downloaded it is required to configure connectors to the database.
Open <TomEE installation folder>/conf/tomee.xml and add the following lines before the tomee closing tag:

```
<Resource id="DreamForumDB" type="DataSource">
    JdbcDriver com.mysql.cj.jdbc.Driver
    JdbcUrl jdbc:mysql://xxx.xxx.xxx.xxx:3306/dream_forum
    UserName yourusername
    Password yourpassword
    JtaManaged true
</Resource>

<Resource id="DreamDataDB" type="DataSource">
    JdbcDriver com.mysql.cj.jdbc.Driver
    JdbcUrl jdbc:mysql://xxx.xxx.xxx.xxx:3306/dream_data
    UserName yourusername
    Password yourpassword
</Resource>
```

where xxx.xxx.xxx.xxx represents your DBMS IP address while yourusername and yourpassword represent your Database username and password.
Finally download mysql-connector-java-8.0.27.jar from the official website and copy it inside <TomEE installation folder>/lib/

Prepare TomEE for Shibboleth SP

Open <TomEE installation folder>/conf/server.xml and navigate until <Service name="Catalina"> and comment all the available connectors, then add the following connector to enable AJP:

```
<Connector protocol="AJP/1.3"
    port="8009"
    redirectPort="8080"
    enableLookups="false"
```

```
    URIEncoding="UTF-8"
    secretRequired="false"
    allowedRequestAttributesPattern=".*" />
```

6.2 Shibboleth SP

To install Shibboleth SP download the latest version from the official website http://shibboleth.net and follow the official instruction that you can find here.
When the installation is completed move to the installation folder.
First of all we need to configure the certificates, we can use a builtin command to do this:

```
shib-keygen -u _shibd -g _shibd -h example.com
    -y 30 -e https://example.com/shibboleth -n sp-signing -f
shib-keygen -u _shibd -g _shibd -h example.com
    -y 30 -e https://example.com/shibboleth -n sp-encrypt -f
/usr/sbin/shibd -t
systemctl restart shibd.service
```

Then we need the attributes that will be received from the IdP, so open `attribute-map.xml` and replace the content with the follow:

```
<Attributes xmlns="urn:mace:shibboleth:2.0:attribute-map"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <!-- DREAM Attributest -->
  <Attribute name="urn:oid:2.5.4.3" id="cn"/>
  <Attribute name="urn:mace:dir:attribute-def:cn" id="cn"/>

  <Attribute name="urn:oid:2.5.4.4" id="sn"/>
  <Attribute name="urn:mace:dir:attribute-def:sn" id="sn"/>

  <Attribute name="urn:oid:2.5.4.42" id="givenName"/>
  <Attribute name="urn:mace:dir:attribute-def:givenName" id="givenName"/>

  <Attribute name="urn:oid:2.16.840.1.113730.3.1.241" id="displayName"/>
  <Attribute name="urn:mace:dir:attribute-def:displayName" id="displayName"/>

  <Attribute name="urn:oid:0.9.2342.19200300.100.1.1" id="uid"/>
  <Attribute name="urn:mace:dir:attribute-def:uid" id="uid"/>

  <Attribute name="urn:oid:0.9.2342.19200300.100.1.3" id="mail"/>
  <Attribute name="urn:mace:dir:attribute-def:mail" id="mail"/>

  <Attribute name="urn:oid:2.16.840.1.113730.3.1.200.2" id="dateOfBirth"/>
```

```
  <Attribute name="urn:mace:dir:attribute-def:dateOfBirth" id="dateOfBirth"/>

  <Attribute name="urn:oid:2.16.840.1.113730.3.1.200.3" id="policyMakerID"/>
  <Attribute name="urn:mace:dir:attribute-def:policyMakerID" id="policyMakerID"/>

  <Attribute name="urn:oid:2.16.840.1.113730.3.1.200.4" id="areaOfResidence"/>
  <Attribute name="urn:mace:dir:attribute-def:areaOfResidence"
                                                        id="areaOfResidence"/>

  <Attribute name="urn:oid:2.16.840.1.113730.3.1.200.5" id="memberOf"/>
  <Attribute name="urn:mace:dir:attribute-def:memberOf" id="memberOf"/>

</Attributes>
```

Then we need to modify some configuration in `shibboleth2.xml` to simplify we provide the whole configuration file highlighting the parts that have to be changed according to your configuration.

```
<SPConfig xmlns="urn:mace:shibboleth:3.0:native:sp:config"
    xmlns:conf="urn:mace:shibboleth:3.0:native:sp:config"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    clockSkew="180">

    <OutOfProcess logger="shibd.logger"
        tranLogFormat="%u|%s|%IDP|%i|%ac|%t|%attr|%n|%b|%E|%S|%SS|%L|%UA|%a">
    </OutOfProcess>

  <StorageService type="Memory" id="mem" cleanupInterval="900"/>
  <SessionCache type="StorageService" StorageService="mem" cacheAssertions="false"
               cacheAllowance="900" inprocTimeout="900" cleanupInterval="900"/>
  <ReplayCache StorageService="mem"/>
  <ArtifactMap artifactTTL="180"/>

  <!-- CHANGE ENTITY ID HERE -->
  <ApplicationDefaults entityID="https://yourservice/shibboleth"
      REMOTE_USER="eppn subject-id pairwise-id persistent-id undefined"
      metadataAttributePrefix="Meta-"
      attributePrefix="AJP_"
      sessionHook="/Shibboleth.sso/AttrChecker"
      cipherSuites="DEFAULT:!EXP:!LOW:!aNULL:!eNULL:!DES:!IDEA
            :!SEED:!RC4:!3DES:!kRSA:!SSLv2:!SSLv3:!TLSv1:!TLSv1.1">

    <Sessions lifetime="28800" timeout="3600" checkAddress="false"
        handlerURL="/Shibboleth.sso" handlerSSL="false" cookieProps="http"
        relayState="ss:mem"
        exportLocation="http://localhost/Shibboleth.sso/GetAssertion"
        exportACL="127.0.0.1"
        idpHistory="false" idpHistoryDays="7">
```

```
<Logout>Local</Logout>
<!-- IDP CONFIGURATION HERE -->
<SessionInitiator type="Chaining" Location="/Login" isDefault="true"
    id="Login" entityID="https://example.com/idp/shibboleth">

    <SessionInitiator type="SAML2" template="bindingTemplate.html"/>
    </SessionInitiator>

    <md:AssertionConsumerService Location="/SAML2/POST" index="1"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
    <md:AssertionConsumerService Location="/SAML2/POST-SimpleSign"
        index="2"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"/>
    <md:AssertionConsumerService Location="/SAML2/Artifact" index="3"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>
    <md:AssertionConsumerService Location="/SAML2/ECP" index="4"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS"/>
    <md:AssertionConsumerService Location="/SAML/POST" index="5"
        Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post"/>
    <md:AssertionConsumerService Location="/SAML/Artifact" index="6"
        Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01"/>


    <LogoutInitiator type="Chaining" Location="/Logout">
        <LogoutInitiator type="SAML2" template="bindingTemplate.html"/>
        <LogoutInitiator type="Local"/>
    </LogoutInitiator>


    <LogoutInitiator type="Admin" Location="/Logout/Admin" acl="127.0.0.1 ::1" />

    <md:SingleLogoutService Location="/SLO/SOAP"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>
    <md:SingleLogoutService Location="/SLO/Redirect"
        conf:template="bindingTemplate.html"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"/>
    <md:SingleLogoutService Location="/SLO/POST"
        conf:template="bindingTemplate.html"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
    <md:SingleLogoutService Location="/SLO/Artifact"
        conf:template="bindingTemplate.html"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>
    <md:ArtifactResolutionService Location="/Artifact/SOAP" index="1"
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>


    <Handler type="MetadataGenerator" Location="/Metadata" signing="false"/>
    <Handler type="Status" Location="/Status" acl="127.0.0.1 ::1"/>
    <Handler type="Session" Location="/Session" showAttributeValues="true"/>
```

```
            <Handler type="DiscoveryFeed" Location="/DiscoFeed"/>
            <Handler type="AttributeChecker" Location="/AttrChecker"
                template="attrChecker.html"
                attributes="mail" flushSession="true"/>
        </Sessions>

        <!--
        Allows overriding of error template information/filenames. You can
        also add your own attributes with values that can be plugged into the
        templates, e.g., helpLocation below.
        -->
        <Errors supportContact="root@localhost"
            helpLocation="/about.html"
            styleSheet="/shibboleth-sp/main.css"/>

        <!-- SET HERE THE URL OF IDP METADATA -->
        <MetadataProvider type="XML" validate="false"
            url="https:/example.com/idp/shibboleth"
            backingFilePath="dreamplatformidp-metadata.xml"
            maxRefreshDelay="7200">
        </MetadataProvider>

        <AttributeExtractor type="XML" validate="true" reloadChanges="false"
            path="attribute-map.xml"/>


        <AttributeFilter type="XML" validate="true" path="attribute-policy.xml"/>


        <CredentialResolver type="File" use="signing"
            key="sp-signing-key.pem" certificate="sp-signing-cert.pem"/>
        <CredentialResolver type="File" use="encryption"
            key="sp-encrypt-key.pem" certificate="sp-encrypt-cert.pem"/>

    </ApplicationDefaults>


    <SecurityPolicyProvider type="XML" validate="true" path="security-policy.xml"/>


    <ProtocolProvider type="XML" validate="true" reloadChanges="false"
        path="protocols.xml"/>

</SPConfig>
```

Finally, restart shibboleth service again:

```
systemctl restart shibd.service
```

6.3 Apache Web Server

Follow this part only if you want to use Shibboleth SP, otherwise go to the next step.
Install the Apache Web server throwing the following command:

```
apt-get install vim default-jdk ca-certificates
    openssl apache2 ntp expat --no-install-recommends
```

Modify now the default website configuration
`<apache installation directory>/sites-enabled/000-default.conf`
replacing the content with the follow:

```
<VirtualHost *>

    ServerName https://forum.dreamplatform.it
    ServerAdmin webmaster@localhost

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    ProxyPass /Shibboleth.sso/ !
    ProxyPass / ajp://localhost:8009/
    ProxyPassReverse / ajp://localhost:8009/

    <Location /dream/login>
        Authtype shibboleth
        ShibRequireSession On
        require valid-user
    </Location>

</VirtualHost>
```

Finally restart Apache

```
systemctl restart apache2
```

6.4 Dream Application

To get a better result install the application as ROOT in the application server. To do this, rename the generated war file into `ROOT.war` and open `<TomEE installation folder>/webapps` folder, rename on delete the actual ROOT folder and copy the `ROOT.war`.
Now you can start the application server using:

```
cd <TomEE installation folder>/bin
sudo ./startup.sh
```

Now you can see the homepage at `http://localhost` (or `http://localhost:8080` if you've not configured Apache).
If you're on Windows use startup.bat instead of .sh one

Local login for testing

For testing purpose you can enable a user to perform local authentication. To do that modify `web.xml` inside the application folder in TomEE webapps (`<TomEE installation folder>/webapps/ROOT/WEB-INF/web.xml`). For the forum application both user and policy maker account could be configured while for the data app only a policy maker.
You can access to the login form as follow:

- **Forum:** connect to `/login` and click on the Admin tab.
- **Data:** connect to `/login/local`

6.5 Identity Provider

The Identity Provider is out of scope of this project but some guideline are given in order to configure it.
Due to the fact that SAML2.0 is a standard you can use the Identity Provider you prefer. In our testing configuration we used OpenLDAP with Shibboleth IdP installed with docker.
OpenLDAP is a open source directory server which let you to maintain data in a gerarchical way. In order to provide the best fit with our needs we also created a custom object that extends the standard `inetOrgPerson` calling it `dreamPerson` with the following definition:

```
dn: cn=dreamperson,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: dreamperson
olcAttributeTypes: ( 2.16.840.1.113730.3.1.200.2
       NAME 'dateOfBirth'
       DESC 'Date of birth of the user'
       EQUALITY caseIgnoreMatch
       SUBSTR caseIgnoreSubstringsMatch
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
olcAttributeTypes: ( 2.16.840.1.113730.3.1.200.3
       NAME 'policyMakerID'
       DESC 'Unique code to enable policy maker role'
       EQUALITY caseIgnoreMatch
       SUBSTR caseIgnoreSubstringsMatch
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
olcAttributeTypes: ( 2.16.840.1.113730.3.1.200.4
```

```
        NAME 'areaOfResidence'
        DESC 'Area of residence of the user'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
olcObjectClasses: ( 2.16.840.1.113730.3.1.200.1
   NAME 'dreamPerson'
   DESC 'dreamPerson'
   SUP inetOrgPerson
   STRUCTURAL
   MAY  (dateOfBirth $ policyMakerID $ areaOfResidence))
```

All the attribute we have configured (using LDAP standard) are the follows:

Table 1 LDAP attributes

Name	OID	Usage
uid	0.9.2342.19200300.100.1.1	User identifier
mail	0.9.2342.19200300.100.1.3	User mail
sn	2.5.4.4	Surname
cn	2.5.4.3	Complete Name
givenName	2.5.4.42	Name
displayName	2.16.840.1.113730.3.1.241	User name to show
dateOfBirth	2.16.840.1.113730.3.1.200.2	Dream date of birth
policyMakerID	2.16.840.1.113730.3.1.200.3	Dream policy maker ID
areaOfResidence	2.16.840.1.113730.3.1.200.4	Dream area of residence
userPassword	2.5.4.35	LDAP password

This attributes are then used to perform user authentication with Shibboleth IdP in which the most important configuration files are the follows:

attribute-resolver.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<AttributeResolver
        xmlns="urn:mace:shibboleth:2.0:resolver"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:mace:shibboleth:2.0:resolver
        http://shibboleth.net/schema/idp/shibboleth-attribute-resolver.xsd">


    <AttributeDefinition xsi:type="Simple" id="uid">
        <InputDataConnector ref="myLDAP" attributeNames="uid"/>
        <AttributeEncoder xsi:type="SAML1String"
            name="urn:mace:dir:attribute-def:uid" encodeType="false" />
```

```
    <AttributeEncoder xsi:type="SAML2String"
        name="urn:oid:0.9.2342.19200300.100.1.1" friendlyName="uid"
        encodeType="false" />
</AttributeDefinition>

<AttributeDefinition xsi:type="Simple" id="mail">
    <InputDataConnector ref="myLDAP" attributeNames="mail"/>
    <AttributeEncoder xsi:type="SAML1String"
        name="urn:mace:dir:attribute-def:mail" encodeType="false" />
    <AttributeEncoder xsi:type="SAML2String"
        name="urn:oid:0.9.2342.19200300.100.1.3" friendlyName="mail"
        encodeType="false" />
</AttributeDefinition>

<AttributeDefinition xsi:type="Simple" id="sn">
    <InputDataConnector ref="myLDAP" attributeNames="sn"/>
    <AttributeEncoder xsi:type="SAML1String"
        name="urn:mace:dir:attribute-def:sn" encodeType="false" />
    <AttributeEncoder xsi:type="SAML2String"
        name="urn:oid:2.5.4.4" friendlyName="sn"
        encodeType="false" />
</AttributeDefinition>

<AttributeDefinition xsi:type="Simple" id="cn">
    <InputDataConnector ref="myLDAP" attributeNames="cn"/>
    <AttributeEncoder xsi:type="SAML1String"
        name="urn:mace:dir:attribute-def:cn" encodeType="false" />
    <AttributeEncoder xsi:type="SAML2String"
        name="urn:oid:2.5.4.3" friendlyName="cn"
        encodeType="false" />
</AttributeDefinition>

<AttributeDefinition xsi:type="Simple" id="givenName">
    <InputDataConnector ref="myLDAP" attributeNames="givenName"/>
    <AttributeEncoder xsi:type="SAML1String"
        name="urn:mace:dir:attribute-def:givenName" encodeType="false" />
    <AttributeEncoder xsi:type="SAML2String"
    name="urn:oid:2.5.4.42" friendlyName="givenName" encodeType="false" />
</AttributeDefinition>

<!-- Schema: inetOrgPerson attributes-->

<AttributeDefinition xsi:type="Simple" id="displayName">
    <InputDataConnector ref="myLDAP" attributeNames="displayName"/>
    <AttributeEncoder xsi:type="SAML1String"
        name="urn:mace:dir:attribute-def:displayName" encodeType="false" />
    <AttributeEncoder xsi:type="SAML2String"
        name="urn:oid:2.16.840.1.113730.3.1.241" friendlyName="displayName"
        encodeType="false" />
```

```
</AttributeDefinition>

    <!-- Schema dreamPerson attributes -->

    <AttributeDefinition xsi:type="Simple" id="dateOfBirth">
    <InputDataConnector ref="myLDAP" attributeNames="dateOfBirth"/>
    <AttributeEncoder xsi:type="SAML1String"
        name="urn:mace:dir:attribute-def:dateOfBirth" encodeType="false" />
    <AttributeEncoder xsi:type="SAML2String"
        name="urn:oid:2.16.840.1.113730.3.1.200.2" friendlyName="dateOfBirth"
        encodeType="false" />
</AttributeDefinition>

    <AttributeDefinition xsi:type="Simple" id="policyMakerID">
    <InputDataConnector ref="myLDAP" attributeNames="policyMakerID"/>
    <AttributeEncoder xsi:type="SAML1String"
        name="urn:mace:dir:attribute-def:policyMakerID" encodeType="false" />
    <AttributeEncoder xsi:type="SAML2String"
        name="urn:oid:2.16.840.1.113730.3.1.200.3" friendlyName="policyMakerID"
        encodeType="false" />
</AttributeDefinition>

    <AttributeDefinition xsi:type="Simple" id="areaOfResidence">
    <InputDataConnector ref="myLDAP" attributeNames="areaOfResidence"/>
    <AttributeEncoder xsi:type="SAML1String"
        name="urn:mace:dir:attribute-def:areaOfResidence" encodeType="false" />
    <AttributeEncoder xsi:type="SAML2String"
        name="urn:oid:2.16.840.1.113730.3.1.200.4" friendlyName="areaOfResidence"
        encodeType="false" />
</AttributeDefinition>


    <AttributeDefinition id="memberOf" xsi:type="ScriptedAttribute">
        <InputDataConnector ref="myLDAP" attributeNames="policyMakerID memberOf" />
        <AttributeEncoder xsi:type="SAML1String"
            name="urn:mace:dir:attribute-def:memberOf" encodeType="false" />
    <AttributeEncoder xsi:type="SAML2String"
        name="urn:oid:2.16.840.1.113730.3.1.200.5" friendlyName="memberOf"
            encodeType="false" />
        <Script><![CDATA[
    valueType = Java.type("net.shibboleth.idp.attribute.StringAttributeValue");

    if (typeof policyMakerID === 'undefined' || policyMakerID.getValues().size() < 1){
        memberOfV = "user";
    } else {
        memberOfV = "policy_maker";
    }
            if (typeof memberOf === 'undefined' || memberOf.getValues().size() < 1){
                    memberOf.addValue(new valueType(memberOfV));
```

```
            }

    ]]></Script>


        </AttributeDefinition>

    <!-- ========================================== -->
    <!--      Data Connectors                       -->
    <!-- ========================================== -->


        <DataConnector id="myLDAP" xsi:type="LDAPDirectory"
        ldapURL="%{idp.attribute.resolver.LDAP.ldapURL}"
        baseDN="%{idp.attribute.resolver.LDAP.baseDN}"
        principal="%{idp.attribute.resolver.LDAP.bindDN}"
        principalCredential="%{idp.attribute.resolver.LDAP.bindDNCredential}"
                >
        <FilterTemplate>
            <![CDATA[
        %{idp.attribute.resolver.LDAP.searchFilter}
            ]]>
        </FilterTemplate>
    </DataConnector>

</AttributeResolver>
```

---

attribute-filter.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<AttributeFilterPolicyGroup id="ShibbolethFilterPolicy"
        xmlns="urn:mace:shibboleth:2.0:afp"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:mace:shibboleth:2.0:afp
        http://shibboleth.net/schema/idp/shibboleth-afp.xsd">

<AttributeFilterPolicy id="Default">
        <PolicyRequirementRule xsi:type="ANY" />
                        <AttributeRule attributeID="uid">
                                <PermitValueRule xsi:type="ANY" />
                        </AttributeRule>

                        <AttributeRule attributeID="givenName">
                                <PermitValueRule xsi:type="ANY" />
                        </AttributeRule>
```

```
                        <AttributeRule attributeID="cn">
                                <PermitValueRule xsi:type="ANY" />
                        </AttributeRule>
                        <AttributeRule attributeID="sn">
                                <PermitValueRule xsi:type="ANY" />
                        </AttributeRule>
                        <AttributeRule attributeID="mail">
                                <PermitValueRule xsi:type="ANY" />
                        </AttributeRule>
                        <AttributeRule attributeID="displayName">
                                <PermitValueRule xsi:type="ANY" />
                        </AttributeRule>

                        <AttributeRule attributeID="dateOfBirth">
                                <PermitValueRule xsi:type="ANY" />
                        </AttributeRule>

                        <AttributeRule attributeID="areaOfResidence">
                                <PermitValueRule xsi:type="ANY" />
                        </AttributeRule>

                        <AttributeRule attributeID="policyMakerID">
                                <PermitValueRule xsi:type="ANY" />
                        </AttributeRule>

                        <AttributeRule attributeID="memberOf">
                                <PermitValueRule xsi:type="ANY" />
                        </AttributeRule>
                </AttributeFilterPolicy>

</AttributeFilterPolicyGroup>
```

---

idp.properties

```
idp.additionalProperties=/conf/ldap.properties,
    /conf/saml-nameid.properties,
    /conf/services.properties,
    /conf/authn/duo.properties


idp.entityID=https://example.com/idp/shibboleth

idp.scope=example.it
```

```
idp.sealer.storeResource=%{idp.home}/credentials/sealer.jks
idp.sealer.versionResource=%{idp.home}/credentials/sealer.kver
idp.sealer.storePassword=YourSecurePassword
idp.sealer.keyPassword=YourSecurePassword


idp.signing.key=%{idp.home}/credentials/idp-signing.key
idp.signing.cert=%{idp.home}/credentials/idp-signing.crt
idp.encryption.key=%{idp.home}/credentials/idp-encryption.key
idp.encryption.cert=%{idp.home}/credentials/idp-encryption.crt
idp.encryption.optional = true

idp.session.StorageService = shibboleth.ClientSessionStorageService

idp.session.trackSPSessions = true
idp.session.secondaryServiceIndex = true
idp.session.defaultSPlifetime = PT2H

idp.authn.flows=Password

idp.ui.fallbackLanguages=en,fr,de
```

---

ldap.properties

```
idp.authn.LDAP.authenticator = bindSearchAuthenticator

idp.authn.LDAP.ldapURL = ldap://localhost:389
idp.authn.LDAP.useStartTLS = false
idp.authn.LDAP.useSSL = false

idp.authn.LDAP.returnAttributes = *

idp.authn.LDAP.baseDN = ou=Users,dc=example,dc=com
idp.authn.LDAP.subtreeSearch = true
idp.authn.LDAP.userFilter = (uid={user})

idp.authn.LDAP.bindDN = cn=admin,dc=example,dc=com
idp.authn.LDAP.bindDNCredential = SecureAdminPassword

idp.authn.LDAP.dnFormat = uid=%s,ou=Users,dc=example,dc=com

idp.attribute.resolver.LDAP.ldapURL = %{idp.authn.LDAP.ldapURL}
idp.attribute.resolver.LDAP.connectTimeout =
```

```
        %{idp.authn.LDAP.connectTimeout:PT3S}
idp.attribute.resolver.LDAP.responseTimeout =
        %{idp.authn.LDAP.responseTimeout:PT3S}
idp.attribute.resolver.LDAP.baseDN = %{idp.authn.LDAP.baseDN:undefined}
idp.attribute.resolver.LDAP.bindDN = %{idp.authn.LDAP.bindDN:undefined}
idp.attribute.resolver.LDAP.bindDNCredential =
        %{idp.authn.LDAP.bindDNCredential:undefined}
idp.attribute.resolver.LDAP.useStartTLS = %{idp.authn.LDAP.useStartTLS:true}
idp.attribute.resolver.LDAP.trustCertificates =
        %{idp.authn.LDAP.trustCertificates:undefined}
idp.attribute.resolver.LDAP.searchFilter = (uid=$resolutionContext.principal)
```

7 Effort Spent

	Alessandro Cecchetto	Mattia Siriani	Matteo Visotto
Time for implementation & testing	60h	60h	60h