# Solution

The application is called JavaApp, is written in Java 8 following Servlet spec, and build with Spring Framework 4 to became Application Server independent. During the development I use Eclipse Oxygen IDE and githiub (https://github.com/brunoneofiti/javaapp).

One of the concerns is to use Spring (and it dependencies, e.g. Jackson) and resources that are provided by Java (JSLT and Servlet), using Maven to manage dependencies, and don’t pack the Tests API (JUnit) and Java provided APIs, this decrease the war file.

The JavaApp is a traditional Java solution, the Controller call the objects in the business layer (business rules, services, the DAO/database) and the presentation layer (JSP). The config package is common, the other packages are divided by business context containing Controller, Model, Service and DAO of the business context. The Controller is manage by Spring.

There are four business context:

* Welcome: Public page.
* Main: Restricted page.
* Login: Public page responsible for authentication.
* Create ATM: Form that pass to a REST/JSON service information of a Dutch city in JSON, this information add to a list of ING ATMs and return a well formed JSON response.
* SelectAtmByCity: A page that shows in the form a list of ING ATMs, if you select a city and submit the form it return the JSON format of ING ATM.

I use Spring to configure the application, all the configuration are in config package. There are two files, one responsible for the presentation layer and other responsible for the application security. The configuration of the presentation layer maps the static content in the resource folder and the JSP files. The security configuration restricts the access to pages and resources and manage authentication.

Spring Security deal with authentication and authorization part, the application has just the basic Spring Security configuration, but Spring Security is so a good security API that it implement CSRF token by default, increasing the application security. From the functional parte the ideia is to restrict the access to some pages or some services.

The presentation layer is done with JSP and Bootstrap, all the static content (JavaScript, CSS and Images) are in the resource folder, CSS is in the begin of the HTML and JavaScript in the end of HTML following the best practices, and they are divided in third-party (bootstrap and jquery) and custom JavaScript (app.js). The JSP do the presentation logic, for example, in Login Page it show or hide messages (error or logout) they use some Java resources, as they are provided by Java they are not in the App package.

The controller layer is done with Spring using Annotation @Controller. The ModelAndView object to send information to the presentation layer. The controller layer uses DAO objects to Access “database” and uses some POJOS (model objects).

There isn’t a database, so the DAO represent it with object that return Java object with data. The model layer was difficult to decide, but, they are based in the ING service to use them to parse (using Jackson) JSON to Java Objects, so DTO are not used in this application. In the model layer we have two main POJOs Address and City.

All the JavaApp services are exposed with REST/JSON and they were build with Spring and Jackson.

<http://localhost:8080/javaapp/ws/getCity?cityname=haia>: One of the tasks is to expose a JSON web service, the ideia of this JSON service is to return a City information (province and country) based in the name, if you didn’t pass a city name the default city is Utrecht. The web service is exposed with Spring annotation.

<http://localhost:8080/javaapp/restricted/createAtm>: Another task of the assignment, is to add a city to the ATMs passing as entrance json data. The form is to send information to the web service, so there is a hidden field containing the json with the information passed to the web service. I could (with more time) pick the form information and add it to a json format and then add it to service.

<http://localhost:8080/javaapp/restricted/selectAtmByCity>: I do a page simple form (selectAtmByCity) to select a city and pass the city name to the web service, this can proof that, the web service search in a database (in this case a list) by the city name and return it in JSON. If you choose Amsterdam it return nothing, but, depending on the scenario it could raise an Exception, most likely, as Exception of the type Business type.

<http://localhost:8080/javaapp/ws/getCityList>: The idea is to return all cities of the web service, there isn’t an entrance parameter to the service.

JUnit is in test scope of Maven, so it runs only by Maven, the idea is to test the access to data that the JavaApp application relays on it. I am not using a Application Server with a relational database that test connection by default.

Components:

* Spring framework - 4.3.11.RELEASE
* Spring security - 4.2.3.RELEASE
* Jackson - 2.6.3
* Jstl - 1.2
* Servlet - 3.0.1
* JUnit - 3.8.1