|  |
| --- |
| **Travaux Pratiques**  **FI BDCC S4 : Architecture distribuée JEE** TP : Gestion des produits (Spring Boot MVC) |

# Introduction :

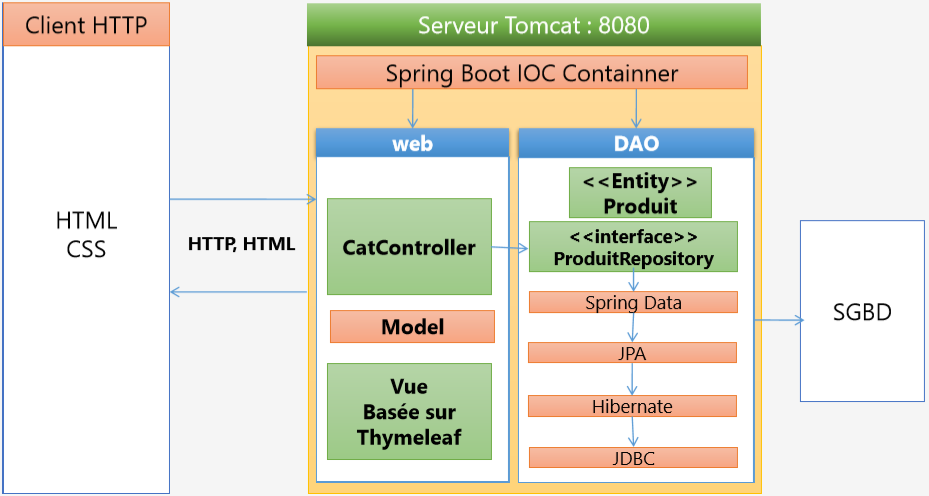
Thymeleaf est un Java XML/XHTML/HTML5 Template Engine qui peut travailler à la fois dans des environnements Web (Servlet) et celui de non Web. Il est mieux adapté pour diffuser XHTML/HTML5 sur View (View Layer) des applications Web basées sur MVC. Mais il peut traiter n'importe quel fichier XML même dans des environnements hors ligne (offline). Il fournit une intégration complète de Spring Framework​​​​​​​.

# Enoncé :

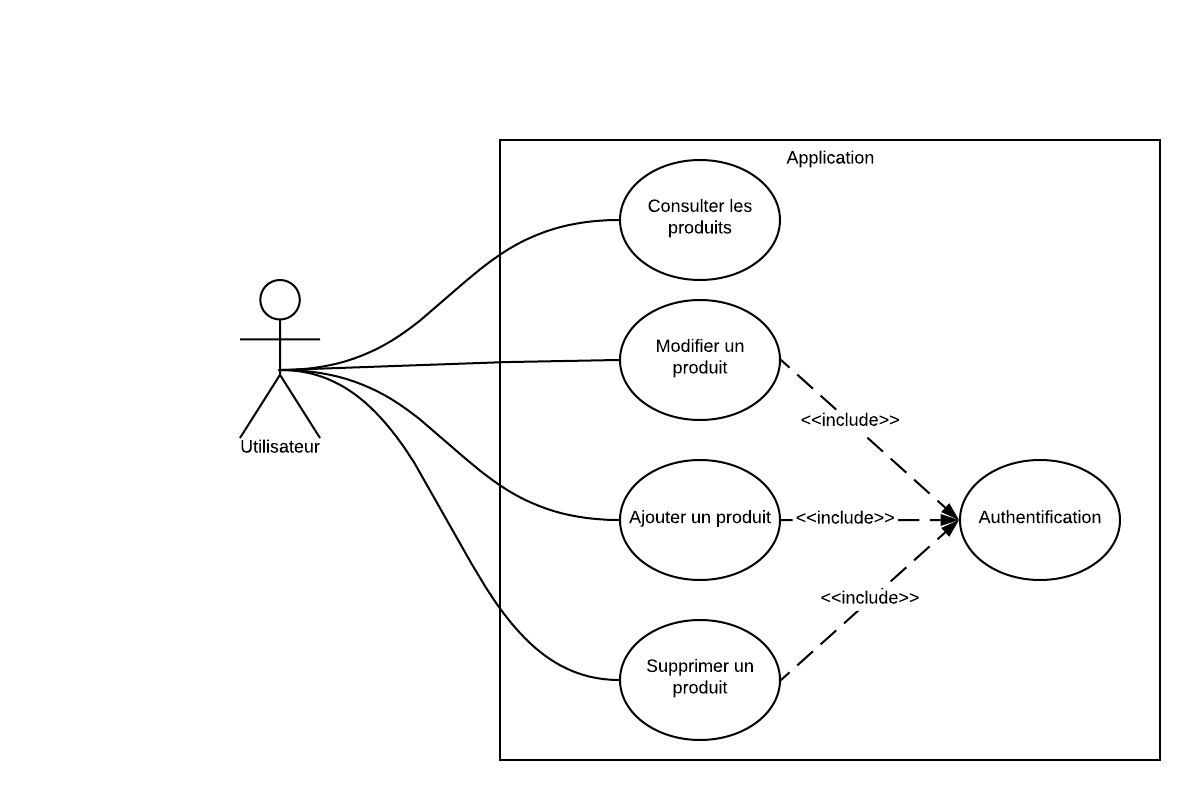
On souhaite créer une application qui permet de gérer les produits en respectant les exigences fonctionnelles : Gérer les produits (ajouter, modifier, supprimer, consulter) et les exigences techniques : les données sont stockées dans une base de données MySQL, l’application se compose de trois couches (la couche DAO, la couche Métier, la couche Web basée sur MVC coté serveur en utilisant Thymeleaf, la sécurité est basée sur Spring Security).

# Architecture et conception :

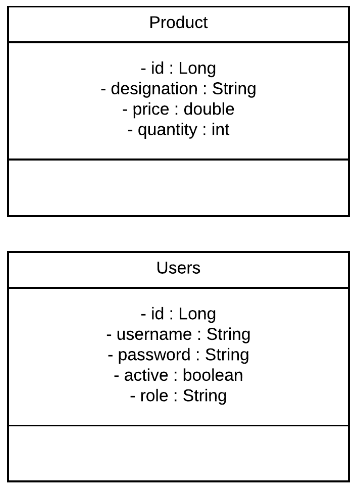
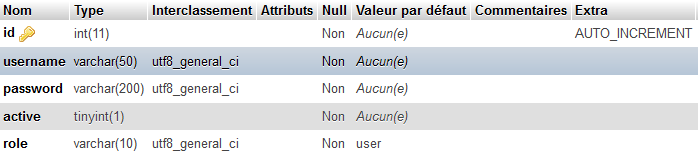
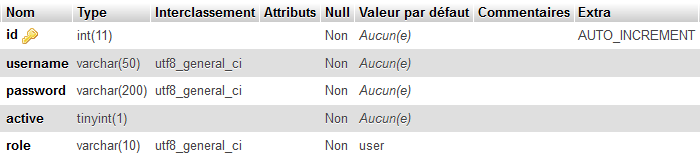
* + Architecture technique :



* + Diagramme de cas d’utilisation :



* + Tableaux utilisés :

# Code source :

* + Structure du TP :



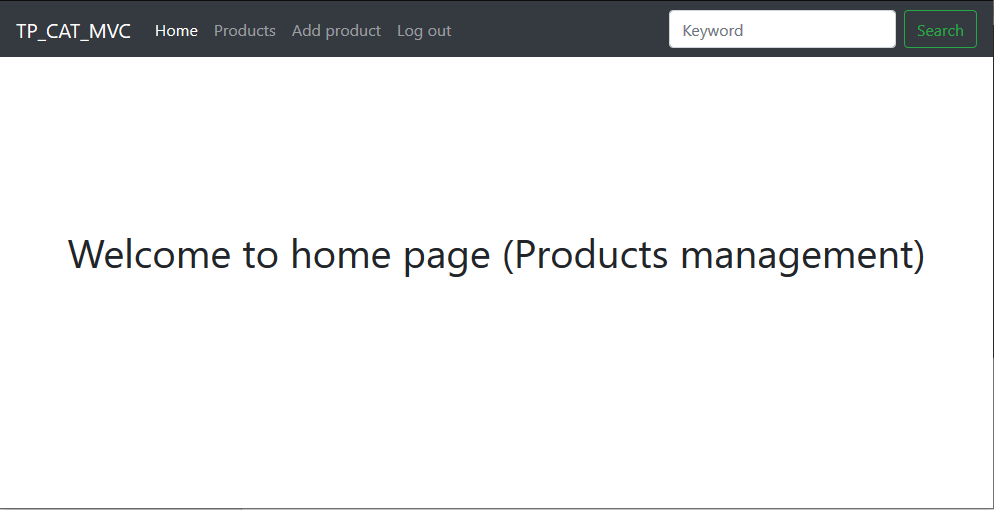
* + ProductController :
* @RestController  
  public class ProductController {  
   private final ProductRepository productRepository;  
   public ProductController(ProductRepository productRepository) {  
   this.productRepository = productRepository;  
   }  
   @RequestMapping(value="/products",method= RequestMethod.*GET*)  
   public ModelAndView products(Model model,  
   @RequestParam(name="page",defaultValue="0")int page,  
   @RequestParam(name="size",defaultValue="4")int size,  
   @RequestParam(name="mc",defaultValue="")String keyword){  
   Page<Product> pageProducts = productRepository.search("%"+keyword+"%", PageRequest.*of*(page, size));  
   model.addAttribute("products", pageProducts);  
   model.addAttribute("currentPage", page);  
   model.addAttribute("mc",keyword);  
   int[] pages=new int[pageProducts.getTotalPages()];  
   for(int i=0;i<pages.length;i++) pages[i]=i;  
   model.addAttribute("pages", pages);  
   return new ModelAndView("products");  
   }  
    
   @RequestMapping(value = "/admin/products/delete",method= RequestMethod.*GET*)  
   public RedirectView delete(Model model, @RequestParam(name="id",required = true)Long id){  
   productRepository.deleteById(id);  
   return new RedirectView("/products");  
   }  
    
   @RequestMapping(value="/admin/form",method= RequestMethod.*GET*)  
   public ModelAndView form(Model model, @RequestParam(name="id",defaultValue = "0") Long id){  
   Product p = (productRepository.existsById(id))? productRepository.getOne(id):new Product();  
   model.addAttribute("product", p);  
   return new ModelAndView("form");  
   }  
   @RequestMapping(value="/admin/save",method=RequestMethod.*POST*)  
   public ModelAndView save(Model model, @Valid Product p, BindingResult bindingResult){  
   if(bindingResult.hasErrors()) return new ModelAndView("form");  
   productRepository.save(p);  
   model.addAttribute("product", p);  
   return new ModelAndView("confirmation");  
   }  
   @RequestMapping(value="/403",method= RequestMethod.*GET*)  
   public ModelAndView error(){  
   return new ModelAndView("error/403");  
   }  
  }
  + Product :
* @Entity  
  @Data  
  public class Product {  
   @Id  
   @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
   private Long id ;  
   @NotNull  
   @Size(min=5,max=12)  
   private String designation ;  
   @NotNull  
   @DecimalMin(value="100")  
   private double price ;  
   @NotNull  
   @DecimalMin(value="0")  
   private double quantity ;  
  }
  + ProductRepository :
* public interface ProductRepository extends JpaRepository<Product,Long> {  
   List<Product> findByDesignation(String designation);  
   @Query("select p from Product p where p.designation like :x")  
   Page<Product> search(@Param("x")String mc, Pageable pageable);  
  }
  + SecurityConfig :
* @Configuration  
  @EnableWebSecurity  
  public class SecurityConfig extends WebSecurityConfigurerAdapter {  
   private final DataSource dataSource;  
   public SecurityConfig(DataSource dataSource) {  
   this.dataSource = dataSource;  
   }  
   @Override  
   protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
   auth.jdbcAuthentication().dataSource(dataSource).  
   usersByUsernameQuery("select username as principal, password as credentials, active from users where username = ?").  
   authoritiesByUsernameQuery("select username as principal, role, active from users where username = ?").rolePrefix("ROLE\_").  
   passwordEncoder(new BCryptPasswordEncoder());  
   }  
   protected void configure(HttpSecurity http) throws Exception {  
   http.formLogin();  
   http.authorizeRequests().antMatchers("/admin/\*").hasRole("ADMIN");  
   http.exceptionHandling().accessDeniedPage("/403");  
   }  
  }
  + 403.html :
* <!DOCTYPE html>  
  <html xmlns="http://www.w3.org/1999/xhtml"  
   xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"  
   layout:decorator="layout" lang="en">  
   <body>  
   <div layout:fragment="content">  
   <table class="position-absolute w-100 h-100">  
   <tr>  
   <td style="vertical-align:middle;" class="text-center">  
   <h1>You don't have the authorization to access to this page</h1>  
   </td>  
   </tr>  
   </table>  
   </div>  
   </body>  
  </html>
  + confirmation.html :
* <!DOCTYPE html>  
  <html xmlns="http://www.w3.org/1999/xhtml"  
   xmlns:th="http://www.thymeleaf.org"  
   xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"  
   layout:decorator="layout" lang="en">  
   <body>  
   <div layout:fragment="content">  
   <table class="position-absolute w-100 h-100">  
   <tr>  
   <td style="vertical-align:middle;">  
   <div class="w-25 m-auto">  
   <div class="card border-dark mb-3">  
   <div class="card-header">Confirmation</div>  
   <div class="card-body">  
   <h5 class="card-title" th:text="${'ID : '+ product.id}"></h5>  
   <h5 class="card-title" th:text="${'Designation : '+ product.designation}"></h5>  
   <h5 class="card-title" th:text="${'Price : '+ product.price}"></h5>  
   </div>  
   </div>  
   </div>  
   </td>  
   </tr>  
   </table>  
   <script>  
   $(function(){  
   $('ul.navbar-nav li').eq(2).addClass('active');  
   });  
   </script>  
   </div>  
   </body>  
  </html>
  + form.html :

J’ai utilisé l’input **name="\_csrf"** contre les attaques CSRF dans toute les formulaires.

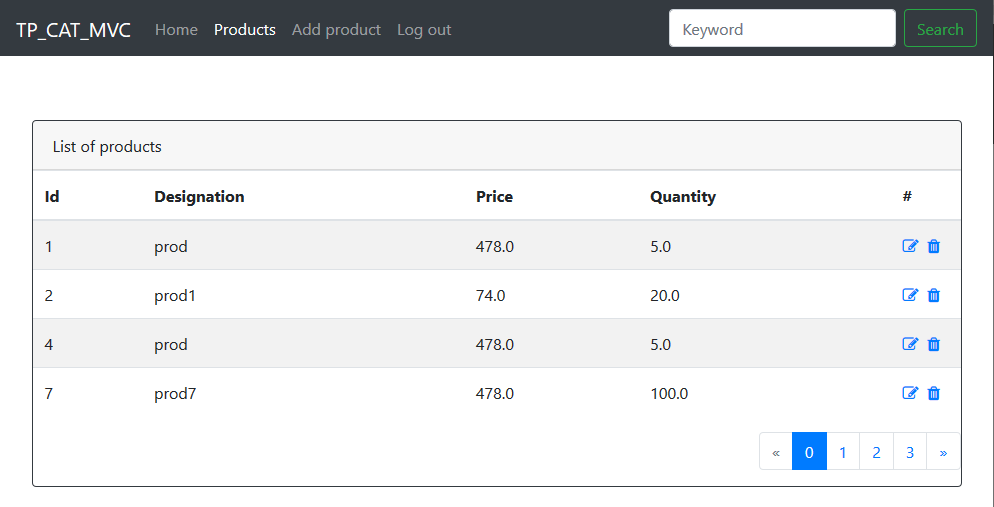
* <!DOCTYPE html>  
  <html xmlns="http://www.w3.org/1999/xhtml"  
   xmlns:th="http://www.thymeleaf.org"  
   xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"  
   layout:decorator="layout" lang="en">  
   <body>  
   <div layout:fragment="content">  
   <table class="position-absolute w-100 h-100">  
   <tr>  
   <td style="vertical-align:middle;">  
   <div class="w-25 m-auto">  
   <form th:action="${'/admin/save'}" method="post">  
   <div th:object="${product}">  
   <div class="form-group">  
   **<input type="hidden" name="\_csrf" value="61390c32-08f8-4c85-b55afe20bc7f0913" />**  
   <input class="form-control" type="hidden" name="id" th:value="\*{id}"/>  
   </div>  
   <div class="form-group">  
   <label class="control-label">Designation : </label>  
   <input class="form-control" type="text" name="designation" th:value="\*{designation}"/>  
   <span class="valid-feedback d-block" th:errors="\*{designation}"></span>  
   </div>  
   <div class="form-group">  
   <label class="control-label">Price : </label>  
   <input class="form-control" type="number" min="100" name="price" th:value="\*{price}"/>  
   <span class="valid-feedback d-block" th:errors="\*{price}"></span>  
   </div>  
   <div class="form-group">  
   <label class="control-label">Quantity : </label>  
   <input class="form-control" type="number" min="100" name="quantity" th:value="\*{quantity}"/>  
   <span class="valid-feedback d-block" th:errors="\*{quantity}"></span>  
   </div>  
   <div>  
   <input class="btn btn-primary" type="submit" value="Save"/>  
   </div>  
   </div>  
   </form>  
   </div>  
   </td>  
   </tr>  
   </table>  
   <script>  
   $(function(){  
   $('ul.navbar-nav li').eq(2).addClass('active');  
   });  
   </script>  
   </div>  
   </body>  
  </html>
  + layout.html :
* <!DOCTYPE html>  
  <html xmlns:th="http://www.thymeleaf.org"  
   xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout" lang="en">  
   <head>  
   <meta charset="UTF-8">  
   <title>Title</title>  
   <link href="/css/bootstrap.css" rel="stylesheet" type="text/css">  
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">  
    
   <script src="/js/jquery.js" type="application/javascript"></script>  
   <script src="/js/popper.js" type="application/javascript"></script>  
   <script src="/js/bootstrap.js" type="application/javascript"></script>  
   </head>  
   <body>  
   <nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">  
   <a class="navbar-brand" href="/">TP\_CAT\_MVC</a>  
   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">  
   <span class="navbar-toggler-icon"></span>  
   </button>  
   <div class="collapse navbar-collapse" id="navbarSupportedContent">  
   <ul class="navbar-nav mr-auto">  
   <li class="nav-item">  
   <a class="nav-link" href="/">Home</a>  
   </li>  
   <li class="nav-item">  
   <a class="nav-link" href="/products">Products</a>  
   </li>  
   <li class="nav-item">  
   <a class="nav-link" href="/admin/form">Add product</a>  
   </li>  
   <li class="nav-item">  
   <a class="nav-link " href="/login?logout">Log out</a>  
   </li>  
   </ul>  
   <form class="form-inline my-2 my-lg-0" action="/products?page=0" method="GET">  
   <input name="mc" class="form-control mr-sm-2" type="search" placeholder="Keyword" aria-label="Search">  
   <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>  
   </form>  
   </div>  
   </nav>  
   <section layout:fragment="content">  
    
   </section>  
   </body>  
  </html>
  + products.html :
* <!DOCTYPE html>  
  <html xmlns="http://www.w3.org/1999/xhtml"  
   xmlns:th="http://www.thymeleaf.org"  
   xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"  
   layout:decorator="layout" lang="en">  
   <body>  
   <div layout:fragment="content">  
   <br/><br/><br/><br/><br/>  
   <div class="container">  
   <div class="card border-dark mb-3">  
   <div class="card-header">List of products</div>  
   <div>  
   <table class="table table-striped">  
   <thead>  
   <tr>  
   <th>Id</th>  
   <th>Designation</th>  
   <th>Price</th>  
   <th>Quantity</th>  
   <th style="width: 70px;">#</th>  
   </tr>  
   </thead>  
   <tbody>  
   <tr th:each="p:${products.content}">  
   <td th:text="${p.id}"></td>  
   <td th:text="${p.designation}"></td>  
   <td th:text="${p.price}"></td>  
   <td th:text="${p.quantity}"></td>  
   <td>  
   <a th:href='${"/admin/form?id="+p.id}' class="fa fa-edit"></a>&nbsp;  
   <a th:href="${'/admin/products/delete?id='+p.id}" class="fa fa-trash"></a>  
   </td>  
   </tr>  
   </tbody>  
   </table>  
   <nav aria-label="Page navigation example">  
   <ul class="pagination justify-content-end">  
   <li th:class="${currentPage == 0}?'page-item disabled':'page-item'">  
   <a class="page-link" th:href="@{products(page=${currentPage - 1},mc=${mc})}" aria-label="Previous">  
   <span aria-hidden="true">&laquo;</span>  
   </a>  
   </li>  
    
   <li th:each="p:${pages}" th:class="(${currentPage==p}? 'page-item active': 'page-item')">  
   <a th:href="@{products(page=${p},mc=${mc})}" th:text="${p}" class="page-link"></a>  
   </li>  
    
   <li th:class="${currentPage == products.getTotalPages()-1}?'page-item disabled':'page-item'">  
   <a class="page-link" th:href="@{products(page=${currentPage + 1},mc=${mc})}" aria-label="Next">  
   <span aria-hidden="true">&raquo;</span>  
   </a>  
   </li>  
   </ul>  
   </nav>  
   </div>  
   </div>  
   </div>  
   <script>  
   $(function(){  
   $('ul.navbar-nav li').eq(1).addClass('active');  
   });  
   </script>  
   </div>  
   </body>  
  </html>
  + application.yml :
* server:  
   port: 8082  
  spring:  
   datasource:  
   url: jdbc:mysql://localhost:3306/db\_cat\_mvc\_1  
   username: root  
   password:  
   jpa:  
   hibernate:  
   ddl-auto: update
  + pom.xml :
* <?xml version="1.0" encoding="UTF-8"?>  
  <project xmlns="http://maven.apache.org/POM/4.0.0"  
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
   <modelVersion>4.0.0</modelVersion>  
   <groupId>org.enset</groupId>  
   <artifactId>TP\_CAT</artifactId>  
   <version>1.0-SNAPSHOT</version>  
    
   <description>Project to manipulate products</description>  
    
   <parent>  
   <groupId>org.springframework.boot</groupId>  
   <artifactId>spring-boot-starter-parent</artifactId>  
   <version>2.2.4.RELEASE</version>  
   <relativePath/>  
   </parent>  
    
   <properties>  
   <java.version>1.8</java.version>  
   </properties>  
    
   <dependencies>  
   <dependency>  
   <groupId>org.springframework.boot</groupId>  
   <artifactId>spring-boot-starter-data-jpa</artifactId>  
   </dependency>  
   <dependency>  
   <groupId>org.springframework.boot</groupId>  
   <artifactId>spring-boot-starter-web</artifactId>  
   </dependency>  
   <dependency>  
   <groupId>mysql</groupId>  
   <artifactId>mysql-connector-java</artifactId>  
   <scope>runtime</scope>  
   </dependency>  
   <dependency>  
   <groupId>org.projectlombok</groupId>  
   <artifactId>lombok</artifactId>  
   <version>1.18.8</version>  
   <scope>provided</scope>  
   </dependency>  
   <dependency>  
   <groupId>org.springframework.boot</groupId>  
   <artifactId>spring-boot-devtools</artifactId>  
   <scope>runtime</scope>  
   <optional>true</optional>  
   </dependency>  
   <dependency>  
   <groupId>org.springframework.boot</groupId>  
   <artifactId>spring-boot-starter-thymeleaf</artifactId>  
   </dependency>  
   <dependency>  
   <groupId>nz.net.ultraq.thymeleaf</groupId>  
   <artifactId>thymeleaf-layout-dialect</artifactId>  
   </dependency>  
   <dependency>  
   <groupId>org.springframework.boot</groupId>  
   <artifactId>spring-boot-starter-security</artifactId>  
   </dependency>  
   </dependencies>  
    
   <build>  
   <plugins>  
   <plugin>  
   <groupId>org.springframework.boot</groupId>  
   <artifactId>spring-boot-maven-plugin</artifactId>  
   </plugin>  
   </plugins>  
   </build>  
  </project>

# Exécution :

* + Page d’accueil :

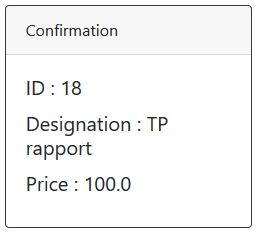


* + Page de liste de produits :



* + Ajouter un produit :

Pour ajouter un produit il faut s’authentifier autant qu’administrateur :

* + Modifier ou supprimer un produit :

Il faut aussi s’authentifier pour exécuter cette action.

# Conclusion :

En utilisant Spring Boot MVC (Spring Security, Thymeleaf …) la realization des applications Web devient très facile et aussi très performante car on sait que coté sécurité et coté optimisation il y a un framework derrière qui fait ça d’une excellente manière.

Code source GitHub : <https://github.com/Medomane/JEE> ainsi une simulation mp4.