

# TP Assurance–2018

---

**Objectif :**

Réaliser une application simple mais complète pour une société d'assurance.

Le futur client (prospect) s'inscrit via le site web, en fournissant ses coordonnées, les caractéristiques de son véhicule, il choisit les différentes options via un système de panier, puis se voit retourner un devis au format PDF.

Par ailleurs, l'administrateur du site doit gérer les différents prospects : lister, supprimer, modifier, ...

Il est prévu d'ajouter une partie mobile avec un client natif Android et une application Web Responsive.

Cette assurance se déroulera en plusieurs phases en fonction des technologies abordées.

**Contraintes de développement :**

- IDE NetBeans, GlassFish, ...
- HTML5, jQuery, ...
- WebServices en Java

# Partie 1 : Côté client : HTML, Ajax, WS RestFul, Websockets

## Modélisation

- Maquettage avec Balsamiq et/ou AxurePRO
- Diagramme de navigation (UML diagramme d'états)
- Diagramme Use Case
- Diagramme de Séquence, pour la partie Ajax et WebSocket

## Phase 1 :

- Etude HTML5 et CSS;
- Construire les pages Web de l'application, avec une charte graphique cohérente
- Choix d'un thème Bootstrap

## Phase 2

- Étude **Javascript** : le langage
- Mettre en place les outils de validation du formulaire via JS et les outils de HTML5
- Remarque : HTML5 prend en charge une bonne partie de la validation, permettant de se passer de JS

## Phase 3

- Étude d'AJAX, interaction avec une Servlet standard.
- Conception d'un « protocole » de communication applicatif, pour récupérer, versions: marques, modèles.
  - Exemple :
    - GET / marques, renvoi de la liste des marques
    - GET /marque=Panhard, renvoi de la liste de la marque,
    - ...
- Injection dans les listes déroulantes : innerHTML ou objets DOM (new Option() ...)
- Réalisation minimale. Dans un premier temps, la liste peut être « en dur » coté serveur, mais sera au final en base de données avec l'ORM JPA.
- Liste déroulante : Marque/Modèle/Version !

## Phase 4

- Réécriture en jQuery de la partie cliente
- Utilisation de JSON pour la base des véhicules, par exemple.
- Intégration d'effets spéciaux : date Picker, Accordeon, etc...
- Recherche et intégration de plugins : carousel-3d, jquery/zepto (line charts) ...

## Phase 5

- Persistance (BDD MySQL, PostgreSQL, SQL Server, ...)
- Conception par un diagramme de classes via **JPA Modeler**
- Accès à la base via une façade, Ejb Stateless, EJB Entity ()
- Étude de l'administration GF

#### **Phase 6**

- WS RestFul pour une autocomplétion d'un champ (par exemple Ville)
- WS RestFul sur GF
- Client JS RestFul

#### **Phase 7**

- WS Soap pour le calcul de la prime d'assurance, par exemple sur la base de la puissance de la voiture, de l'âge du conducteur, ...
- Réalisation en Java côté serveur

#### **Phase 8**

- Réalisation du bandeau publicitaire par WebSockets
- Réservation d'une DIV
- API Jersey (JSR 356)
- 

## **Partie 2 : Côté serveur MVC, JPA (!)**

#### **Phase 1**

- Contrôleur en Servlet, AEF, ...
- Modélisation par un UML : diagramme Etat/transition
- Réalisation par le pattern « Commande »
- View : pages JSP avec EL (pas de Java dans JSP !)
- Communication inter Servlet et JSP (setAttribute, objet session)

#### **Phase 2**

- Persistance avec JPA : approfondissement (1:1; 1:n, mode lazy et eager, ...)
- Déploiement et diagramme UML
- Test des interfaces graphiques en mode automatique avec Selenium/Jubula

#### **Phase 3**

- Mise en place de la partie métier JavaEE

- EJB Stateless, statefull, remote, ...
- 

#### **Phase 4**

- Mise en route du Framework JSF 2
- Composants Primefaces

### **Partie 3 : Mobilité**

#### **ANDROID natif**

- à l'usage de commerciaux
- pour déterminer la prime d'assurance pour un véhicule donné

#### **Client Mobile :**

- Application Web de type Web Responsive

#### **Développement Hybride : Cordova/Phonegap**

## Partie 4 : Evolution de l'architecture initiale

### Côté client

- Evolution conception côté client : Angular JS (FW de consommation WS Rest)
- Interconnexion Java Serveur
- 

### Côté serveur NodeJS

- Reprendre l'architecture côté serveur,
- Avec JS côté serveur
- [Framework](#) Web avec SocketIO, ou [ExpressJS](#)
- ORM pour NodeJS
- <http://www.toam.fr/06-04-2014-bien-demarrer-avec-nodejs/>
- 

### Côté serveur et client

- Symfony, LARAVEL - PHP

## RESSOURCES

### MVC et servlet

- Support Afpa : JSP et Servlets - Concepts avancés (MVC et pattern Commande) - page 38

### WebSocket

- <https://netbeans.org/kb/docs/javaee/maven-websocketapi.html>

### Angular

- <http://thecodersbreakfast.net/index.php?post/2013/08/12/Restangular-integration-de-Restlet-et-AngularJS>
- <https://www.tutorialspoint.com/angularjs/index.htm>

### Closure :

- <https://developer.mozilla.org/fr/docs/Web/JavaScript/Closures>
- [http://www.w3schools.com/js/js\\_function\\_closures.asp](http://www.w3schools.com/js/js_function_closures.asp)
- <http://www.w3schools.com>

## JPA

- <http://jpamodeler.github.io/>
- [Vidéo](#) JPA Modeler NB
- [Slides](#)
- [JPA et Primefaces](#) avec NB (2014)

## Etude de tuto

- <https://netbeans.org/kb/docs/javaee/ecommerce/intro.html>
- [Diagramme de navigation en UML](#)

## ANNEXES

### Expression des Besoins

#### Pour le client

- page d'accueil statique
- page d'inscription: nom, prénom, date de naissance, adresse complète avec validation sur la page client en JS/jQuery
- page de description du véhicule : (marque, modèle, version, puissance, millésime, et avec rappel du nom et prénom saisis sur la page précédente

- pages avec le choix
- page affichage du calcul de la prime d'assurance (fonction de l'âge par exemple) avec rappel du nom/prénom et du véhicule choisi. Le calcul de la prime d'assurance sera réalisé par un Web Service
- navigation entre toutes les pages (avant et arrière) avec conservation des données.
- mémorisation de la connexion pour un client donné et restauration des données
- Chaque page aura un emplacement réservé pour la publicité (ou chat) ! Celle-ci sera modifiée automatiquement toutes les 5 secondes. Le délai sera réglable par l'administrateur du site. Les messages publicitaires seront textuels ! (voir pour la transmission d'images)

### **Options :**

- Chat client/assureur : le client peut échanger des messages avec son assureur en temps réel.
- Evolution Web Workers : ajax, transfert des images, animation bannière, compression/décompression d'images, polling d'URLs, indiquer l'arrivée du devis, ...
- (évolution indexedDB) A la relance de la page HTML, sur le site, dans la page "définition du véhicule" les informations importantes seront restituées à partir de "indexdDB" de HTML5
- (évolution) i18n : mode international
- Page de paiement en ligne avec PayPal (SandBox)
- Approche noSQL, pour remplacer la base ou une partie !

### **Pour l'administrateur de l'application**

- page avec la liste des inscrits, permettant d'accéder à la page des coordonnées du client sur un clic
- page du client : pour modifier les coordonnées d'un client
- réglages publicitaires : les différents textes sont stockés en base

### **Éléments d'Architectures**

- Servlet, JSP, Javabeans, MVC, pattern Commande
- Sécurisation des couches : convertors et validators
- EJB - Stateless, ...
- Pool de connexion du container
- Calcul de la prime d'assurance, via un WS
- Application multi-lingues/internationalisation
- (évolution): Cryptographie asymétrique et certificat numérique
- JS, jQuery, Bootstrap, Ajax, WebSocket
- Muti-threading : Java
- Pattern : MVC, Commande, Façade, Singleton, ...

## WebSocket

- est supporté nativement par le serveur JavaEE7
- <http://www.imdoudoux.fr/java/dej/chap-websockets.htm>
- voir w3schools coté client
- côté serveur : <https://netbeans.org/kb/docs/javaee/maven-websocketapi.html>

## Pour IndexedDB et/ou WebSQL (pérennité ???)

- <https://developer.mozilla.org/fr/demos/tag/tech:indexeddb>
- w3schools
- <http://www.debray-jerome.fr/articles/L-api-websql-en-html5.html>
- websql vs indexedDB :  
<http://www.html5rocks.com/en/tutorials/webdatabase/websql-indexeddb/>

## Diagramme de navigation :

- <https://www.lucidchart.com/invitations/accept/18ade8b6-285b-4d67-98f0-df124ad7be7d>
- Bases du hacking : <http://www.bases-hacking.org/hacking.html>

## Diagramme de déploiement :

- [http://fr.wikipedia.org/wiki/Diagramme\\_de\\_d%C3%A9ploiement](http://fr.wikipedia.org/wiki/Diagramme_de_d%C3%A9ploiement)



## LIVRABLES - DSPP

### Première partie

- Use Case
- Maquettes
- Diagramme de navigation

### Deuxième partie

- Diagramme de séquence : Ajax et WebSocket
- Diagramme de classe - JPA
- Codes sources commentés
- plan de test : Code junit et Code Test Selenium

### Troisième partie

- documentation technique
- documentation utilisateur
- diagramme UML de déploiement

---

Vous recevez ce message, car vous êtes un membre du groupe [Pompey-cdi11](#). Si vous ne souhaitez pas recevoir de messages ou d'événements de ce groupe, [arrêter de le suivre dans votre boîte de réception](#).

[Afficher les conversations du groupe](#) | [Afficher les fichiers du groupe](#)