

DOSSIER DE VALIDATION

CONCEPTEUR DEVELOPPEUR D'APPLICATIONS NUMERIQUES

Nom Prénom	BENQUET Emeline
Nom(s) Prénom(s) du ou des tuteurs	Carole LAUGA – Anthony DARDILHAC
Acronyme de la certification IPI visée	CDAN
Niveau visé	RNCP niveau 6 (ancien niveau II)
Date de la soutenance	Lundi 25 novembre 2019
Lieu de la soutenance	EPSI/WIS BORDEAUX – 114 rue Lucien Faure – 33000 BORDEAUX

Remerciements

Je tiens à remercier toutes les personnes qui ont contribuées au succès de mon alternance et qui m'ont aidées lors de la rédaction de ce dossier.

Je remercie CGI pour avoir mis en place l'école « U-DEV », d'avoir pris en charge ma formation et de m'avoir accueilli dans ses locaux lors de cette année d'alternance. Également de m'avoir permis de monter en compétences et de continuer à apprendre encore auprès de toutes ses équipes.

Je souhaite aussi remercier l'EPSI qui, en partenariat avec l'école « U-DEV », nous a mis à disposition leurs locaux ainsi que leurs intervenants afin de nous mettre en place des cours sur des thèmes divers et variés mais toujours en lien avec le monde de l'informatique.

Pour finir, je souhaite remercier tout particulièrement mes collègues, **Alexandre SAVARY** et **Sébastien CARNELOS**, pour toute l'aide qu'ils m'ont apportée, d'avoir pris le temps de me former tout au long de cette année et pour leur patience.

Table des matières

Remerciements.....	3
Présentation Personnelle	5
1 - Présentation du groupe CGI	6
Présentation générale	6
1.2 - Centre de production du site de Bordeaux	8
1.3 - Business Unit : France Global Delivery Center (FGDC).....	8
1.4 – Un rêve à grande échelle.....	9
2 - Présentation du poste	11
3 - Le projet : Spirit un Framework propriétaire	12
3.1 - Spirit qu'est-ce que c'est ?	12
3.2 - Principes fondamentaux.....	12
3.3 - Cinématique inter applications / transactions.....	13
3.4 – IHM SPIRIT	13
3.3.1 - Règles IHM.....	13
3.3.2 - Connecteurs Spirit	14
3.3.3 - Architecture	15
3.3.4 – Squelette.....	16
3.3.5 – Environnement et livraison	17
4 - Les outils de développement pour SPIRIT	18
5 - Création d'une application Spirit	20
5.1 - Initialisation du projet.....	20
5.2 - Build du projet avec Grunt	20
5.3 - Organisation des sources de la transaction	21
Missions.....	33
Montée de version GOAL : Vendre sur Bourse France	33
1.1 - Présentation de Bourse France	33
1.2 - La mission	34
Les Châtaignettes	38
Conférence sur l'hypnose médicale.....	40
EAT THE WORD.....	41
ANNEXES	48
FICHE TUTEUR	49
CV	50
Tableau de compétences	51
Cahier des charges	58
Exemple Bouchon.....	64
DCIG	65
CONCLUSION.....	68

Présentation Personnelle

Avant de me diriger vers l'univers du développement web j'ai tout d'abord passé un **bac professionnel en Communication Visuelle et Plurimedias**. Durant ces trois années, j'ai pu effectuer plusieurs stages en entreprise, qui m'ont permis de me former et de me perfectionner au métier de graphiste. Par la suite, j'ai continué mes études au sein de l'IUT d'Angoulême, pour passer un **DUT Métiers du Multimédia et de l'Internet**. J'ai pu apprendre les bases du Web, de la communication et de l'audiovisuel. Après ces cinq années formatrices, j'ai eu l'opportunité de travailler en tant que webdesigner. J'ai donc saisi cette chance. J'avais enfin un pied dans le milieu du travail. Ce métier m'a beaucoup plu, mais au bout de deux ans j'avais besoin d'un nouveau challenge dans ma vie professionnelle, de progresser techniquement parlant. J'ai donc eu l'opportunité de faire la formation U-DEV avec l'entreprise CGI. Ce nouveau projet m'a plu j'ai donc embarqué dans cette nouvelle aventure pour devenir développeuse.

Dans ce dossier de validation, je vais principalement parler de mon année d'alternance chez CGI et par la suite de certaines expériences ou projets qui m'ont permis de valider certaines compétences pour l'obtention de ce titre :

Concepteur Développeur d'Application Numérique

1 - Présentation du groupe CGI

Présentation générale



Logo CGI de 1976 à 1982 Logo CGI de 1982 à 1993 Logo CGI de 1993 à 1998 Logo CGI de 1998 à 2012 Logo de la filiale "CGI Federal" à partir de janvier 2006 Logo CGI à partir de janvier 2013³⁴

Pour commencer un petit peu d'information sur CGI, une entreprise Canadienne, qui a été fondée en 1976, par deux jeunes entrepreneurs de 26 ans, Serge Godin et André Imbeau. Pour l'anecdote, ils ont démarré l'entreprise dans le sous-sol de la résidence de Serge Godin dans la ville de Québec avec un seul client, un téléphone et beaucoup d'ambition.

L'acronyme CGI signifie : « Conseil en Gestion et Informatique », plusieurs traductions sont disponibles en anglais, la plus courante étant « Consultants in management and information technology ». Mais, l'entreprise étant Canadienne, le nom en français a été retenu.

Durant ces 10 premières années, l'entreprise a développé une stratégie, un modèle et un ensemble de principes de gestion, appelé « la Constitution de CGI », qui continuent de guider l'entreprise encore aujourd'hui. A partir des années 80, l'entreprise commence à faire l'acquisition d'autres entreprises de services d'intégration de systèmes permettant ainsi à l'entreprise CGI de devenir un groupe et d'élargir son secteur d'activité. À partir des années 2000, la clientèle de CGI comprend des leaders mondiaux permettant au groupe de continuer de s'étendre et d'être reconnu mondialement.

CGI continue sa croissance et essaye continuellement d'agrandir son marché grâce à ses rachats d'entreprise. Après l'acquisition de plusieurs grands groupes tel que Stanley Inc. et de ses filiales Oberon, Techrizon, CGI vient de réaliser en 2010, sa plus grande acquisition avec le rachat de Logica, une entreprise anglo-néerlandaise de services technologiques et d'entreprise. Cette acquisition a plus que doublé la taille de l'équipe de CGI en passant de 30 000 professionnels, appelés « membre », à plus de 65 000.

Aujourd'hui, le groupe compte plus de 74 000 membres, travaillant à travers le monde entier et permet donc de servir et satisfaire ses clients dans les Amériques, en Europe et en Asie. Le groupe se classe actuellement au cinquième rang parmi les plus importantes entreprises mondiales dans le secteur du « service-conseils complets en technologie de l'information et en management-gestion des processus d'affaires ». CGI est reconnue pour l'excellence de ses services et affiche une croissance rentable durable.

Pour terminer, le groupe CGI reste fidèle à sa Constitution tout en continuant de s'adapter au changement du marché des technologies de l'information afin de répondre aux exigences commerciales de ses clients à l'échelle locale et mondiale et de satisfaire les attentes de ses professionnels et de ses actionnaires.

CGI est présent partout dans le monde comme le montrent les cartes ci-dessous :



CGI travaille pour cinq secteurs cibles :

- 34 % gouvernement,
- 23 % secteur manufacturier, distribution et détail,
- 21 % services financiers,
- 15 % télécommunications et services publics,
- 7% santé.

Croissance interne		Croissance par acquisition	
Contrats d'intégration de systèmes et projets	Contrats d'impartition	Acquisitions de firmes spécialisées	Acquisitions transformationnelles
<ul style="list-style-type: none"> ■ Offre plus vaste de services aux clients établis ■ Nouveaux contrats et renouvellements ■ Partenariats avec de nouveaux clients 	<ul style="list-style-type: none"> ■ Croissance du nombre d'opportunités ■ Nouveaux contrats d'impartition avec des clients établis et nouveaux 	<ul style="list-style-type: none"> ■ Amélioration de l'offre dans nos secteurs cibles ■ Élargissement de notre présence géographique ■ Extension de notre portefeuille de solutions 	<ul style="list-style-type: none"> ■ Accroissement de la masse critique afin de se positionner pour remporter plus de grands contrats ■ Élargissement de notre présence géographique ■ Convergence stratégique et contribution au bénéfice net

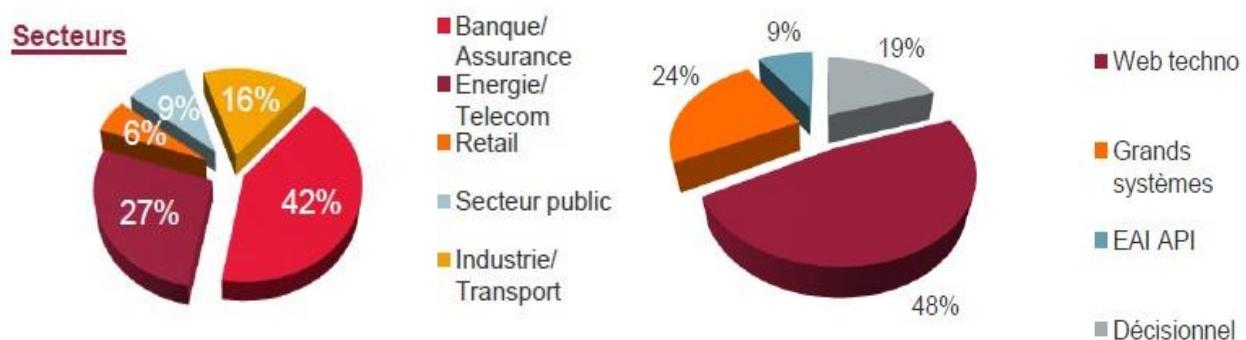
1.2 - Centre de production du site de Bordeaux

Le centre de Bordeaux compte environ 700 membres et propose plusieurs types de services :

- AMS : Evolutif, maintenance corrective, assistance
- Projets de développement
- Architecture logicielle / Chantiers de performance
- Pôle de compétence Qualimétrie

Etude, déploiement et supervision d'infrastructures de développement.

CGI Bordeaux travaille pour différents secteurs tels que les banques et assurances, l'énergie et les télécoms, le commerce, le secteur public, l'industrie et le transport.



1.3 - Business Unit : France Global Delivery Center (FGDC)

FGDC, qui compte plus de 1 600 professionnels, a démarré ses activités en 2003, à plus de 350 clients internes et externes de tous les secteurs de métiers. FGDC est présent sur 10 sites industrialisés en France et au Maroc et présente une large palette de compétences : SAP, Test, JAVA, Cobol, BI, Support, Oracle, AS/400, .Net, SharePoint, infrastructure Management...

Les services proposés par FGDC sont :

- Conseil et stratégie d'utilisation
- Conseil intégration au SI
- Formation et support
- Tierce Recette Applicative
- Intégration Déploiement

- Tierce Maintenance Applicative
- Migrations Applicatives
- Data Management
- Supervision et Administration
- Réseau et Téléphonie
- Hébergement et Cloud
- Sécurité.

Les projets de l'entité FGDC utilisent un certain nombre d'outils leurs permettant d'être performants :

- Datamart : analyse de données, tableaux de bord, agrégation de données
- Digital Kanban : intègre les systèmes de ticketing, permet un retour visuel sur les échéances prévues
- GrecO'nligne : gestion des compétences et des ressources
- Une gestion des compétences intégrée aux processus de gestion d'activités
- SONARQube : solution complète d'audit de code de l'éditeur SonarSource, permet une vision globale et détaillée de la qualité et permet de suivre son évolution
- Cockpit : un outil d'affichage d'informations en temps réel, open source fortement configurable.

1.4 – Un rêve à grande échelle

La vision et la mission diffusées aux salariés sont les suivantes : « Être un leader de classe mondiale en technologies de l'information et en gestion des processus d'affaires qui contribue au succès de ses clients ». CGI dit également s'inspirer d'un rêve : « Crée un environnement où nous avons du plaisir à travailler ensemble et où, en tant que propriétaires, nous participons au développement d'une entreprise dont nous sommes fiers ».

Pour cela elle s'appuie sur des valeurs :

- Partenariat et qualité
- Objectivité et intégrité
- Intrapreneurship et partage
- Respect
- Solidité financière
- Responsabilité sociale

Aujourd’hui, il est difficile pour les équipes de s'approprier ces valeurs car elles sont trop abstraites. Ces valeurs, cette vision et ce rêve sont ceux de la direction, c'est un objectif à grande échelle qui s'applique à CGI dans son ensemble. De ce fait, il est plus difficile à visualiser au niveau d'un seul projet, aux niveaux des salariés. L'entité FGDC à Bordeaux en a conscience et étudie en ce moment une autre façon de les diffuser, de les faire connaître et de donner aux différents collaborateurs l'envie d'y adhérer.

2 - Présentation du poste

Le pôle CGI Bordeaux est divisé en trois grandes parties, nommées Business Units. J'ai intégré l'équipe **IHM** (Interface Homme Machine), au sein de la Business Unit France Global Delivery Center (FGDC), présenté ci-dessus. Plus précisément, l'équipe IHM qui travaille pour un client bancaire. Cette équipe a pour but de réaliser des développements d'applications IHM.

Après la formation de la **Cellule d'expertise IHM de BDDF-IT**, qui porte la responsabilité du maintien et de l'évolution des Interfaces Homme Machine des transactions bancaires utilisées par le réseau France ainsi que les socles techniques (frameworks) sur lesquels reposent ces développements, l'équipe peut intervenir sur la réalisation de développement et de correction d'anomalies.

BDDF-IT Direction informatique de la Banque de Détail en France (BDDF), définit et met en œuvre les solutions et outils informatiques adaptés à ses clients. Elle accompagne les différents métiers dans le développement de leurs ambitions commerciales. BDDF-IT s'inscrit dans un cadre industriel de conception et de réalisations de solutions informatiques de qualité répondant aux besoins de ses utilisateurs.

Dans le cadre des projets métiers gérés au sein de BDDF-IT, l'atelier d'expertise IHM contribue à la phase de conception des interfaces des transactions bancaires. Elle est garante de la mise en place d'une démarche de conception centrée sur les utilisateurs et prend en charge la réalisation des maquettes.

J'ai donc eu la chance de pouvoir évoluer au sein de cette toute jeune équipe qui vient d'être créée depuis quelques mois et de pouvoir participer tout au long de cette année à cette nouvelle aventure au sein de CGI.

Le client : une banque internationale

Pour des raisons de confidentialité, je n'ai pas le droit de citer le nom du client pour lequel je travaille.

Le client est un groupe bancaire international présent dans 75 pays, et comptant 189 000 employés. Il fait partie du CAC 40. Le résultat net du groupe s'élève, au 31 décembre 2016, à 7 702 millions d'euros, c'est la première banque de la zone euro en 2016 et la cinquième banque internationale. Elle compte 32 millions de clients particuliers et 1 million de professionnels.

3 - Le projet : Spirit un Framework propriétaire

3.1 - Spirit qu'est-ce que c'est ?

SPIRIT est un outil de développement des Interfaces Homme – Machine, pour la DSI BDDF, plus précisément un Framework. Il contribue à industrialiser la fabrication logicielle des applications. Il a été créé spécifiquement pour le client.

SPIRIT permet :

- l'harmonisation de la présentation et de la navigation, ainsi que la standardisation dans le transactionnel,
- d'améliorer la productivité en fournissant des normes et des composants préfabriqués,
- la portabilité (pas d'adhérence au poste, compatibilité universelle) et l'évolutivité (internationalisation, indépendance de l'IHM et de la couche métier) du transactionnel.

3.2 - Principes fondamentaux

SPIRIT est basé sur une architecture client léger avec une navigation Web. Les utilisateurs accèdent aux applications et transactions SPIRIT par un navigateur web standard non customisé (IE9 sur les postes NEOS). Par exigence, SPIRIT est portable et n'a pas besoin de la SDO pour fonctionner.

Toutes les applications SPIRIT sont optimisées pour un affichage dans la résolution native des écrans standards, soit 1280 X 1024 pixels minimum pour des écrans 17 pouces. Toutefois, selon la configuration du poste de travail des utilisateurs, il est possible d'optimiser l'affichage de l'application pour des écrans 24 pouces.

Une application peut être vue en 3 parties distinctes :

- la couche donnée,
- la couche métier qui met en œuvre toute l'intelligence métier de l'application,
- et la couche de présentation de l'application (IHM), la partie visible par l'utilisateur.

Le socle technique SPIRIT permet uniquement de développer la couche de présentation d'une application. SPIRIT est une solution technique qui permet :

- de mettre en forme les informations que l'intelligence métier lui transmet,
- de recueillir des données saisies par l'utilisateur pour les transmettre à l'intelligence métier,
- de gérer les actions et les évènements de l'utilisateur.

Cela signifie que les contrôles réalisés par la partie IHM ne concernent que les contrôles de surface. Aucun contrôle métier ou calcul n'est donc réalisé par SPIRIT. L'IHM n'est pas capable de lire le contenu d'un fichier soumis par l'utilisateur (par exemple un fichier .CSV).

Ce traitement doit être réalisé par la couche métier.

Contrairement à VIOTO, il n'y a pas de session sur le serveur mutualisé SPIRIT. Cela veut dire que tant que l'utilisateur ne ferme pas la page (affichée dans un onglet du navigateur) sur laquelle il travaille, les données saisies non enregistrées ne sont pas perdues. Il peut reprendre et terminer une saisie qu'il aurait initiée la veille par exemple.

La mise en cache de données temporaires peut être réalisée en utilisant les possibilités offertes par le navigateur avec HTML 5. Par exemple, cela permet de mettre en mémoire des données spécifiques devant subsister après une actualisation de la page.

3.3 - Cinématique inter applications / transactions

Une transaction peut être lancée à partir :

- du Dossier Client,
- du Bureau,
- d'une autre transaction ou application V4, V6, VIOTO ou SPIRIT,
- d'un favori ou d'un lien.

Il est possible qu'une transaction ne soit lançable qu'à partir d'une autre transaction et non depuis le Bureau ou le Dossier Client. Dans ce cas, à des fins de test, la transaction est lançable unitairement via son URL. Les paramètres sont alors visibles dans l'URL.

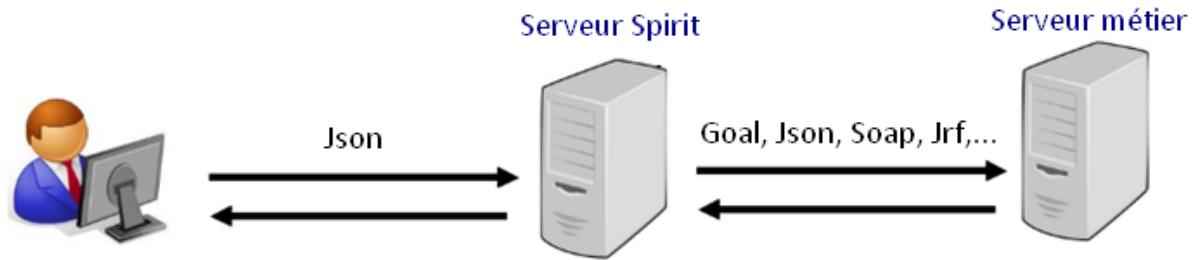
3.4 – IHM SPIRIT

3.3.1 - Règles IHM

L'IHM SPIRIT fonctionne sur les principes suivants :

- 1 appel par action utilisateur.
- Rebonds transactionnels :
 - 3 paramètres maximum
 - Pas de rafraîchissement automatique de l'application appelante
- Volumétrie appel GOAL :
 - Tenter de limiter au maximum la volumétrie des échanges entre IHM et services

3.3.2 - Connecteurs Spirit



C'est une interface entre l'IHM statique et le service métier.

Le type de l'appel métier est transparent. Les données en entrées sont toutes en JSON et tous les appels se font via la fonction callService.

Les informations passent toutes du poste client au serveur Spirit. L'ouverture de flux se fait donc entre serveurs et non pas avec chaque poste client.

Il renseigne les données de contexte. Cela évite ainsi que l'utilisateur puisse usurper une identité dans la console. Chaque service est associé à un connecteur dans le spirit-config.xml. C'est en spécifiant le type de connecteur que l'on indiquera à la partie dynamique du serveur Spirit comment faire l'appel.

Par exemple : Connecteur Goal

Il est déclaré de manière suivante :

```

<!-- 013 -->
<service id="dss_i.1.0.cree" type="goal">
    <param name="sdo-transaction-code" value="COAAC" />
    <param name="ims-transaction-code" value="YOA21" />
</service>
  
```

Comme on peut le voir il y'a plusieurs paramètres :

- sdo-transaction-code: liste des codes transaction IHM. Par défaut, c'est le code transaction principal qui est utilisé. Il faut préciser dans la requête (en Get : /sdo=) le code à utiliser si on en déclare plusieurs
- ims-transaction-code: liste des codes transaction IMS. Il faut préciser dans la requête (en Get: /ims=) le code IMS à utiliser si on en déclare plusieurs

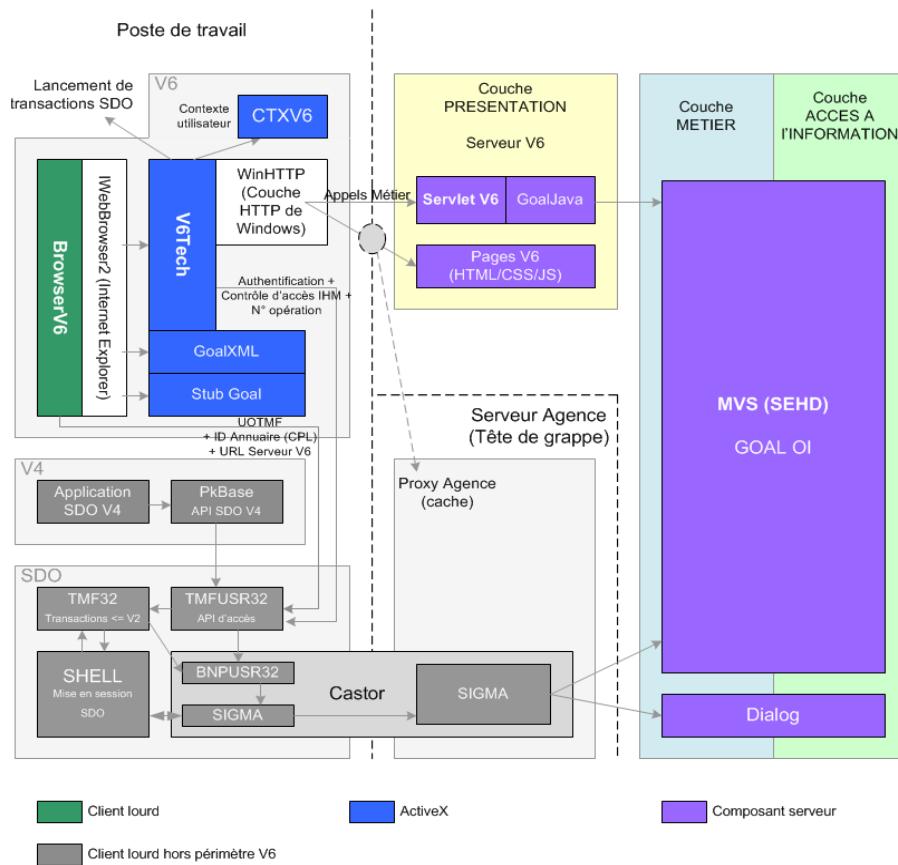
Les paramètres en GET ne doivent pas être vide :

```

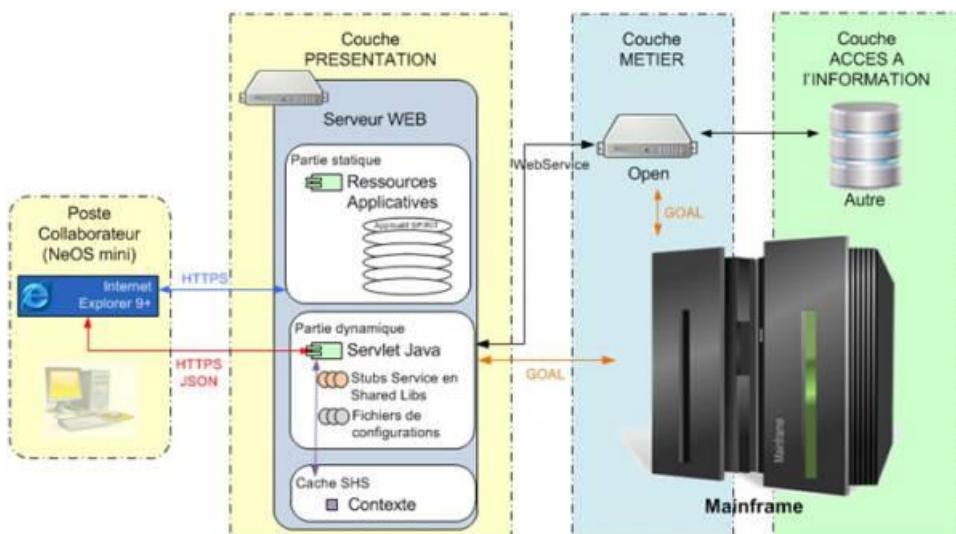
this.callService("defipro_1.1.0.gere_comtr/jms=" + App.contexte.IMSCode + "/sdo=" + App.contexte.TransactionCode +
  (App.operationNumber ? "/operation=" + App.operationNumber : ""), sentObject, next);
  
```

3.3.3 - Architecture

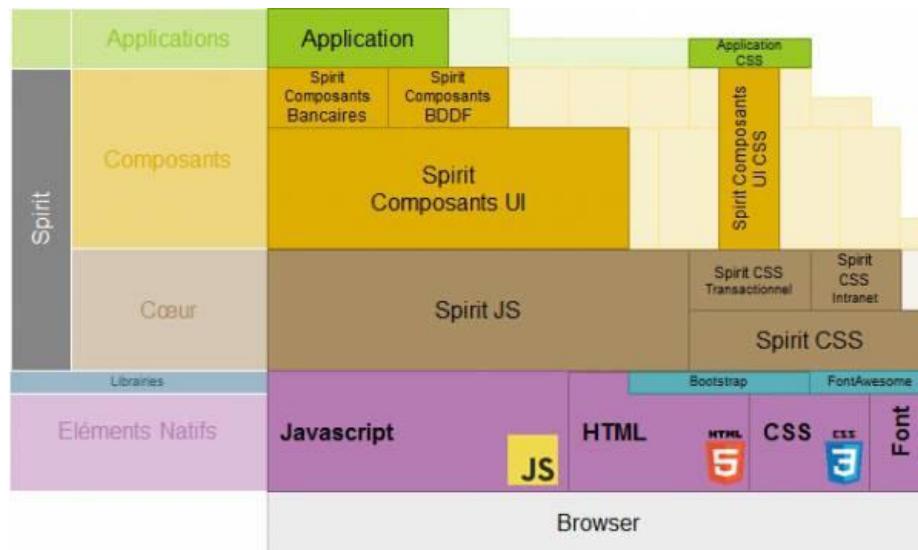
L'architecture SPIRIT sera présentée en 2 parties : une partie serveur et une partie client.



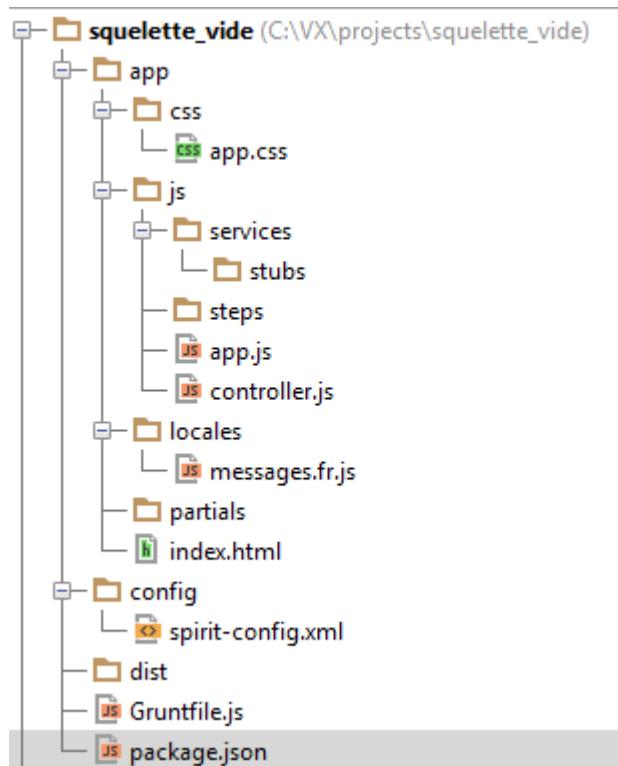
- L'architecture serveur se compose d'une partie commune JAVA à toutes les applications qui est appelée en AJAX et qui échange du JSON avec la partie client.



- Quant à l'architecture client, Spirit JS fonctionne à partir des standards JS/HTML5/CSS3 pour délivrer des applications IHM puissantes.



3.3.4 – Squelette



- Gruntfile
 - Fichier de configuration du « build »
- Package.json
 - Fichier de configuration du projet
- app/
 - Contient les sources applicatives
- dist/
 - Dossier contenant les fichiers buildés.
- config/
 - Contient le fichier spirit-config.xml – Fichier de paramétrage pour les appels métiers. (Possibilité de spécifier par environnement en plaçant un fichier spirit-config.xml dans des répertoires pour chaque environnement: dev, cpl, val et prod).

J'explique plus en détails l'architecture d'un projet quelques pages après.

3.3.5 – Environnement et livraison

Une fois que la tâche demandée est effectuée, il faut s'occuper de faire la livraison pour le client. Sur ce projet, je n'ai accès qu'à deux environnements pour la livraison DEV et CPL (expliquer ci-dessous).

C'est le client qui se charge de faire la livraison sur les autres environnements une fois que les anomalies ou éventuelles améliorations sont effectuées.



Développement : Environnement utilisé pendant la phase de développement.

- **CPL** : Premier environnement où sont livrées les applications IHM. Les MOE valident la livraison et réalisent des tests d'intégration avec leurs composants métiers.
- **VALREC** : Environnement presque dimensionné comme celui de production, avec les mêmes infrastructures. Il est utilisé par la MOA pour réaliser la recette fonctionnelle.
- **Prégénéralisation** : Environnement de production identique à celui de généralisation. Il est utilisé par une population restreinte pour tester des sites d'expérimentation.
- **Généralisation** : Environnement de production accessible à tous les utilisateurs.

4 - Les outils de développement pour SPIRIT

Les outils qui vont être présentés ci-dessous sont ceux que j'utilise principalement pour travailler sur des transactions SPIRIT.

4.1 – Tortoise SVN



Tortoise SVN est un outil permettant de gérer le versioning de fichier de façon graphique. Les fonctionnalités de SVN sont directement proposées via le menu lors d'un clic-droit dans un explorateur. Il est installé dans l'ordinateur des membres de l'équipe, il aide facilement à partager des données sur SVN Server et à récupérer des données de SVN Server. C'est un modèle de travail entre Tortoise SVN et SVN Repository.

4.2 – Server Apache



Apache est un serveur http qui permet de gérer les requêtes navigateurs pour récupérer le code applicatif. La configuration nécessaire pour le fonctionnement des applications Spirit est simple.

Node JS



Node.js est une plateforme logicielle libre et événementielle en JavaScript. Dans le contexte de développement Spirit, NodeJS va simplement permettre d'avoir la possibilité de récupérer des modules nécessaires au build d'un projet.

WebStorm



WebStorm est un IDE pour les langages Web, développé par l'entreprise JetBrains et basé sur la plateforme IntelliJ IDEA.

BootStrap 3



Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

Depuis la version 2, le framework adopte la conception de sites web adaptatifs, permettant aux projets utilisant Bootstrap de s'adapter dynamiquement au format des supports depuis lesquels ils sont accédés (PC, tablette, smartphone).

5 - Création d'une application Spirit

5.1 - Initialisation du projet

- Pour tout nouveau projet, la MOE doit créer une ASAP avec :
 - Information sur la TMA concernée
 - Création de l'espace svn + droit svn
 - Droit de livraisons
- Actuellement, lorsqu'un correspondant IHM crée l'espace svn, il met un squelette applicatif dans le répertoire.

5.2 - Build du projet avec Grunt

- Paramétré par le Gruntfile.js, cet outil permet de construire l'application étape par étape.
 - Templating
 - Concaténation des fichiers js
 - Minimification...

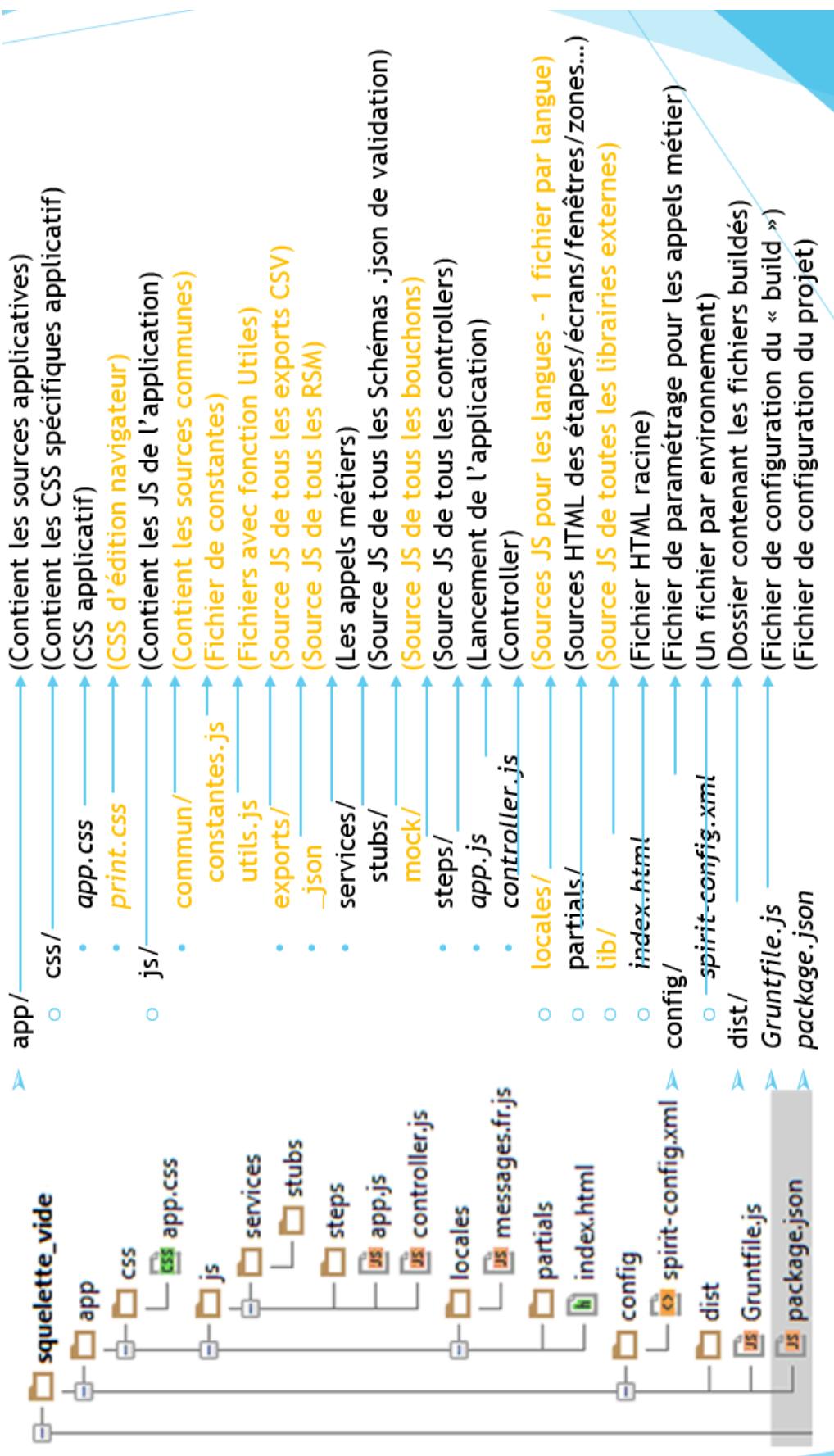
```
grunt.registerTask("prod", ["clean:main", "dom tpl", "rewrap", "uglify",
concat", "cssmin", "clean:min", "clean:tmp", "copy", "targethtml:dist"]);
```

- JSHint : Contrôle du code, calcul de complexité.

La configuration : .jshintrc (ne pas y toucher, l'outil de livraison a son propre fichier et bloquera en cas d'anomalie)

- Utilisation :
 - Directement sur Webstorm
 - En ligne de commande (grunt + target)

5.3 - Organisation des sources de la transaction



6.3.1 - Package.json – lien static / dynamique

```
{  
  "name": "ARQ00",  
  "version": "0.0.1",  
  "socle": "1.0.6-dev",  
  "code": "ARQ00",  
  "theme": "v6",  
  "services": [  
    {  
      "type": "goal",  
      "stubs": [  
        "7wa-1.0",  
        "6zt-1.0",  
        "6zq-1.0",  
        "6zr-1.0"  
      ]  
    },  
    {  
      "type": "papyrus-pdf",  
      "stubs": [  
        "com.bnpp.spirit:papyrus.backend:1.4"  
      ]  
    }  
  ],  
}
```

- **Name / Code** : le code transaction de l'application. Des contrôles sont réalisés sur cette valeur.
- **Version** : donnée mise à jour lors de la livraison.
- **Socle** : permet de définir lors du grunt le package Spirit à inclure.
- **Thème** : spécifique au programme de transformation.
- **Services** : la déclaration des services est nécessaire pour certains connecteurs et plus particulièrement avec goal qui doit utiliser des stubs côté java.

6.3.2 - Initialisation de l'application – App.js

C'est un objet avec une structure basique fournie dans le squelette.

```
this.App = {  
    code: "AVPM1",  
    name: "AVPM1",  
    Routes: {},  
    Tpl: {},  
    Messages: {fr: {}},  
    Params: {},  
    Services: {},  
    Steps: {},  
    Links: [  
        {code: "CVPM2"}  
    ]  
};
```

- Code et name servent comme base de nommage et de control lors du grunt.
- Lorsque l'utilisateur crée les routes, il doit initialiser cette propriété pour que le socle puisse y accéder.
- Les fonctions d'initialisation de ces propriétés seront générées par le grunt lors du build:
 - Tpl : les templates,
 - Messages : les libellés internationalisés
 - Services : la liste des services déclarés dans la configuration
 - Steps: la liste des étapes dans le répertoire step
- La liste des codes transactions de l'application, mais aussi des liens transactionnels lancables doit être déclarée ici pour que les données du descstran soient obtenues.

6.3.3 - Initialisation de l'application – Controller.js

```
App.Steps.e404 = Spirit.ui404("container");
Spirit.Bddf.init(App, {
  agl: true,
  container:"container",
  context : true,
  callback: function (ctx) {
    if(Spirit.BddfLink.canLaunch("CVPM2")) {
      App.Routes = new Spirit.Router({
        "/": initializeApplication,
        "/contracts" : App.Steps.searchAndDisplayContract,
        "/requests" : App.Steps.searchAndDisplayRequest,
        "/*": App.Steps.e404
      });
    } else {
      App.Routes = new Spirit.Router({
        "/": App.Steps.searchAndDisplayContract,
        "/contracts" : App.Steps.searchAndDisplayContract,
        "/*": App.Steps.e404
      });
    }
    App.Routes.run();
  }
});
```

On trouve principalement l'appel à Spirit.Bddf.init.

- Agl : pour obtenir les données de l'AGL à partir des App.links et afficher le nom de la transaction tel que défini dans l'AGL
- Container : cadre principal de l'application
- Context : option permettant d'obtenir les informations de contexte suite à l'initialisation.
- Callback : Suite à l'initialisation on teste la juridiction et on crée les routes.
- Récupération des paramètres.
-

6.3.4 - Système de routes

- La navigation est réalisée via les URL sous la forme de Hashbangs
 - URL : #!/step1
 - Déclaration : « /step1 » : App.Steps.step1
- Le code déclarant les routes est normalement dans le « controller.js »
- Objectif: Lier une fonction à une URL

- Une route peut être paramétrable
 - Déclaration : « /step1/:id » : App.Steps.step1
 - La fonction associée aura un paramètre récupérable qui sera la chaîne dans l'url
- URL : #!/step1/6 → Fonction déclarée : App.Steps.step1(param) → A l'exécution : param vaudra « 6 »
- L'exécution d'une nouvelle route permet de :
 - Changer la mise en page
 - Changer les données affichées
 - Mettre des pages d'erreurs

6.3.4 - Construire une Step standard

- Définir la fonction (constructeur de l'object Step) lancée par la route :

```
App.Steps.searchAndDisplayContract = function searchAndDisplayContract() {
```

- Récolter les données d'initialisation de l'écran
 - En les collectant à partir de variables diverses.
 - En réalisant un appel métier (qui doit être unique). L'enchaînement d'appels métiers est interdit.
- Générer le DOM de la page à partir d'un template :

```
Spirit.Dom.populateDom(App.Tpl.tplFormSearchContracts({
  messages: App.Messages.contracts.formSearchContracts,
  userAuthorized: !Spirit.BddfLink.canLaunch("CVPM2")
}), "form");
```

- Instancier les composants Spirit (création de tableaux, tree, etc...)
- Gérer les événements : Le code exécuté lors des événements doit être associé à la Step comme méthode.
 - Ajouter les appels de services dans les événements.
 - Il faut obligatoirement une action utilisateur pour faire un appel.
Un seul appel au central par action utilisateur autorisé (sauf GED).
- Ajouter les redirections : normalement pas d'appel à l'ouverture de la prochaine Step.

Exemple de Schéma attendu :

```
(function (Spirit, App) {
    "use strict";

    // Listeners
    var validerClick = function (parameters) {
        [...]
        App.Routes.to("/suite");
    }
    var imprimerClick = function (parameters) {
        [...]
    }

    // Step
    App.Steps.foStart = function () {
        //animation chargement de la page
        Spirit.UiLoading("container");

        //appel central pour récupérer les données
        App.Services.getInformations( params , function(dataOut) {

            //ajout d'infos
            dataOut.property = value;

            //construction du template
            Spirit.Dom.populateDom(App.Tpl.tplIdentification(dataOut), "container");

            //initialisation des objets Spirits
            this.tableFdcList = Spirit.UITableView("tableFdcList", {
                selMode: "single",
                currentPage: App.Variables.formFdc.page,
                sortLocal: false,           fixedHeader: true,          altRows: true,           showSelCheckbox: true,      pageShowTotal: true,
                headers: [
                    {name: App.Messages.fdcList.labelHeaderIdentifiant, sortable: false, className: "text-center"}, 
                    {name: App.Messages.fdcList.labelHeaderPresentationATraiter, className: "text-center", headers: [
                        {name: App.Messages.fdcList.labelHeaderPresentationATraiterSAGED, sortable: false, className: "text-right"}, [...]
                    ]},
                    ...
                ]
            });

            //ajout des evenements
            Spirit.Events.addEvent("buttonAnnuler", "click", function () {[...]});
            Spirit.Events.addEvent("buttonValider", "click", validerClick);
            [...]
        });
    };
})
```

6.3.5 - Manipulation DOM

```
["ul",
  ["li",
    {
      "style" : "color:red" },
     "First Item"
   ],
  ["li",
    {
      "title" : "some hover text.",
      "style" : "color:green"
    },
     "Second Item"
   ],
  ["li",
    ["span",
      {
        "class" : "code-example-third" },
       "Third"
     ],
     "Item"
   ]
]
```

- Qu'est-ce que le DOM (Document Object Model) ?
- L'API Spirit contient des fonctions pour manipuler le DOM
 - Les plus utilisées :
 - Spirit.Dom.populateDom()
 - Spirit.Dom.cleanNode()
 - Compatibilité navigateur
 - Simplification de tâches redondantes
 - Respect des normes de sécurité / architecture
- JsonML* (JSON Markup Language) :
 - Représentation JSON du DOM

*JsonML, le langage de balisage JSON est un langage de balisage léger utilisé pour mapper entre XML et JSON.

6.3.6 – Les templates

- Le chargement d'un écran est réalisé « zones / sous-zones » de manière dynamique. Un template = une zone ou sous-zone.
- Utilisation du moteur de templating **dom_tpl v1.0.5**
- Par convention, les templates ont comme extension .tpl. Ils sont stockés dans app/partials/
- L'écriture d'un fichier tpl se fait en HTML.
- Il est possible :
 - D'exploiter des **données** passées en paramètre via l'objet json it.
 - D'insérer des macros fonctions pour rendre dynamique la construction des écrans avec des instructions :
 - Conditionnelles
 - Itératives
 - ...

Les étapes :

- **Découpage en cadres / sous-cadres**
- Découpage en lignes et **colonnes**
- Remplissage des zones

Référence Contrat	Offre	Relation commerciale	Etat	Debut	Fin	Avenant en cours	Suspens client	Feu Rouge Distributeur	Action
30004 00813 RAN00170528	NETCASH AMPLITUDE	0138252525252	En création	01/01/2017					
30004 01328 RAN00636853	MA BANQUE ENTREPRISE EVOLUTION	0138252525252	Validé	15/06/2017					

- Le code dynamique se construit ainsi:
 - **{{=it.VARIABLE}}** pour afficher une valeur / un libellé
 - **{{~it.VARIABLE:elem:i}}** pour réaliser une boucle foreach (function(elem,i)) {...}
 - **{{~}}** pour terminer la boucle
 - **{{?it.VARIABLE === X}}** pour réaliser un if(var === x) {
 - **{{??it.VARIABLE === Y}}** pour démarrer un elseif (var === Y) {
 - **{{??}}** pour démarrer le else
 - **{{?}}** pour terminer le if

6.3.7 – Les évènements en Spirit

- Spirit.Events
 - addEvent/remove : `Spirit.Events.addEvent("buttonAddContract", "click", addContract);`
Fonction apportant
une compatibilité navigateur.
 - fireEvent : déclenche un événement dom comme s'il s'agissait d'un utilisateur.
 - on/off: Démarre/Stop l'écoute événements sur un élément.
- Certains composants Spirit déclenchent des évènements non pas sur action utilisateur, mais sur un changement d'état. Cette fonction prend en paramètre une fonction de callback exécutée lors du déclenchement.
 - trigger : déclenche un événement pour signifier un changement aux éléments en écoute. (Généralement réservé au socle)
- Chain : Utilisé pour « capter » les retours des appels asynchrones.

6.3.8 – Les appels aux services

En Spirit, il existe différents types des services métiers :

- Mainframe ou Goal
- Interaction
- Webservices RestFull
- Webservices Soap (legacy)
- Ged
- Papyrus

Je vais principalement parler ici de la mise en place d'un service GOAL car c'est celui que j'ai le plus utilisé au cours de cette année de formation.

6.3.9 - Appels aux services métier – les stubs GOAL

- Mise en place :
 - 1 - Récupération des stubs : liens fournis par le client, génération de stub à l'aide d'un mnémonique.
 - 2 - Le fichier .js doit être placé dans le dossier app/js/services/stubs. Le nom du fichier est le nom du service.
 - 3 – Déclaration dans le spirit-config et si besoin dans le package.json (cf chapitre sur les connecteurs)
- Utilité :
 - Donne la signature du service
 - Permet de contrôler la requête envoyée
- Précautions
 - Il ne faut pas modifier ce fichier généré car il est associé à un jar sur le serveur.

6.3.9 - Appels aux services métier – callService

- 1 fichier par service dans le répertoire app/js/services. Il est préconisé de nommer le fichier srv_+ nom du service .js
- Plusieurs zones importantes que l'on doit toujours retrouver :

```
App.Services.param_pst_typ = function param_pst_typ(callback, dataIn, errorContainer, warningContainer) {
    Spirit.Service.Application(App, Spirit.Events.chain(function (next) {
        var req = {
            "no_version_1": 1,
            "fond_in1": {
                "fond_in101": {
                    "c_type_traitement": dataIn.cTypeTrait
                }
            },
            "fond_in_out1": {
                "fond_in_out101": [
                    "g_donnees_trt_t1234": {
                        "c_poste_type_t1234": dataIn.cTypePost,
                        "l_abrg_poste_type_t4": dataIn.LPostRype,
                        "c_expln_fmrce_t1234": dataIn.CodFam,
                        "l_expln_fmrce_t1234": dataIn.LibFam,
                        "c_categ_vndr_t1234": "",
                        "c_cit_prtcl_t1234": "",
                        "c_cit_entrp_t1234": "",
                        "c_comss_acte_t1234": "",
                        "c_cagnot_t1234": "",
                        "c_prime_accueil_t1234": "",
                        "c_fond_commun_t1234": "",
                        "d_deb_vldt_t1234": dataIn.dDebVldt,
                        "d_fin_vldt_t1234": dataIn.dFinVldt,
                        "d_maj_t1234": dataIn.dMajVldt
                    }
                ]
            }
        };
        this.callService("mpc_pr_pot.1.0.get_postes_types", req, next);
    }).chain(function (next, res) {
        //chain(function (res, ASR) {
        var returnService;
        var returnServiceInOut;
        if (res instanceof Spirit.Service.TechnicalException) {
            // Erreur lors de l'appel du service métier
            Spirit.Error(errorContainer, { title: "Erreur technique", text: "Erreur lors de l'appel au service : " + res.message, "closeable": true});
            return callback(null);
        }
        if (res && res.result && res.result.tech_out1.code_retour === 0) {
            if (4 === res.result.tech_out1.code_retour) {
                Spirit.Error(warningContainer, {
                    title: "Erreur applicative métier",
                    type: "warning",
                    text: res.result.fond_out1.fond_out101.l_ret + " (" + res.result.fond_out1.fond_out101.c_ret + ")",
                    "closeable": true
                });
            } else {
                Spirit.Error(errorContainer, {
                    title: "Erreur applicative métier",
                    type: "error",
                    text: res.result.fond_out1.fond_out101.l_ret + " (" + res.result.fond_out1.fond_out101.c_ret + ")",
                    "closeable": true
                });
            }
        }
        if ((8 === res.result.tech_out1.code_retour)) {
            return callback(null);
        }
        App.Utils.traverse(res.result, App.Utils.processRTrim);
        returnService = App.Utils.buildObjectForHandle(res.result.fond_out1, modelResult, null);
        returnServiceInOut = App.Utils.buildObjectForHandle(res.result.fond_in_out1, modelResultInOut, null);
        callback(returnService, returnServiceInOut, res.result.tech_out1.code_retour);
        //, res.result.tech_out1.code_retour
    })
    })
});
```

Avant l'appel :

```
App.Services.param_pst_typ = function param_pst_typ(callback, dataIn, errorContainer, warningContainer) {
    Spirit.Service.Application(App, Spirit.Events.chain(function (next) {
        var req = {
            "no_version_1": 1,
            "fond_in1": {
                "fond_in101": {
                    "c_type_traitement": dataIn.cTypeTrait
                }
            },
            "fond_in_out1": {
                "fond_in_out101": [
                    "g_donnees_trt_t1234": {
                        "c_poste_type_t1234": dataIn.cTypePost,
                        "l_abrg_poste_type_t4": dataIn.LPostRype,
                        "c_expln_fmrce_t1234": dataIn.CodFam,
                        "l_expln_fmrce_t1234": dataIn.LibFam,
                        "c_categ_vndr_t1234": "",
                        "c_cit_prtcl_t1234": "",
                        "c_cit_entrp_t1234": "",
                        "c_comss_acte_t1234": "",
                        "c_cagnot_t1234": "",
                        "c_prime_accueil_t1234": "",
                        "c_fond_commun_t1234": "",
                        "d_deb_vldt_t1234": dataIn.dDebVldt,
                        "d_fin_vldt_t1234": dataIn.dFinVldt,
                        "d_maj_t1234": dataIn.dMajVldt
                    }
                ]
            }
        };
        this.callService("mpc_pr_pot.1.0.get_postes_types", req, next);
    }).chain(function (next, res) {
        //chain(function (res, ASR) {
```

Initialisation de la structure à passer en entrée de service avec les données passées en paramètres (dataIn).
En Goal : structure récupérable dans les stubs

Appel au service. (avec éventuellement des paramètres GET)

Un événement sera déclenché au retour du service. On utilise le mécanisme « chain » pour capter ce retour et le traiter. Le retour se trouve dans la paramètre res

Après l'appel :

```

}).chain(function (next, res) {
  var returnService;
  var returnServiceInOut;
  if (res instanceof Spirit.Service.TechnicalException) {
    // Erreur lors de l'appel du service métier
    Spirit.Error(errorContainer, { title: "Erreur technique",
      text: "Erreur lors de l'appel au service : " + res.message, "closeable": true});
    return callback(null);
  }

  if (res && res.result && res.result.tech_out1.code_retour !== 0) {
    if (4 === res.result.tech_out1.code_retour) {

      Spirit.Error(warningContainer, {
        title: "Erreur applicative métier",
        type: "warning",
        text: res.result.fonc_out1.fonc_out101.l_ret + " (" + res.result.fonc_out1.fonc_out101.c_ret + ")",
        "closeable": true
      });
    }
    else{

      Spirit.Error(errorContainer, {
        title: "Erreur applicative métier",
        type: "error",
        text: res.result.fonc_out1.fonc_out101.l_ret + " (" + res.result.fonc_out1.fonc_out101.c_ret + ")",
        "closeable": true
      });
    }
    if ((8 === res.result.tech_out1.code_retour)) {
      return callback(null);
    }
  }

  App.Utils.traverse(res.result, App.Utils.processRTrim);
  returnService = App.Utils.buildObjectForHandle(res.result.fonc_out1, modelResult, null);
  returnServiceInOut = App.Utils.buildObjectForHandle(res.result.fonc_in_out1, modelResultInOut, null);
  callback(returnService, returnServiceInOut, res.result.tech_out1.code_retour);
  // res.result.tech_out1.code_retour
}).end();

```

Gestion des erreurs techniques retournées par le serveur Spirit.

Gestion des code retours applicatifs.
Attention aux règles « message bloquant/non bloquant/ message d'info »

Pré-traitement éventuel de données en sortie.
Est-il opportun de le faire ici? : Ce traitement doit être nécessaire à chaque appel et ne modifie pas un IHM particulier.

Passage des données de sortie à la fonction de callback. C'est dans la fonction de callback que sont construit les IHM où sont fait les traitements spécifiques à une fonctionnalité/étape.

6.3.10 - Création et utilisation d'un bouchon « IHM »

Pourquoi créer des bouchons ? Lorsque l'on reçoit un DCIG* et qu'on n'a pas encore le service développé, on doit au moins avoir la signature. Le bouchon va nous permettre de simuler l'appel au central.

* DCIG : Document de Conception d'Interface Graphique

Missions

Montée de version GOAL : Vendre sur Bourse France

1.1 - Présentation de Bourse France

L'application Bourse France concerne la négociation en bourse des valeurs qui cotent sur le marché français.

Un client doit nécessairement posséder un compte titre (CIF, PEA ou PEA PME) pour négocier des ordres Bourse France.

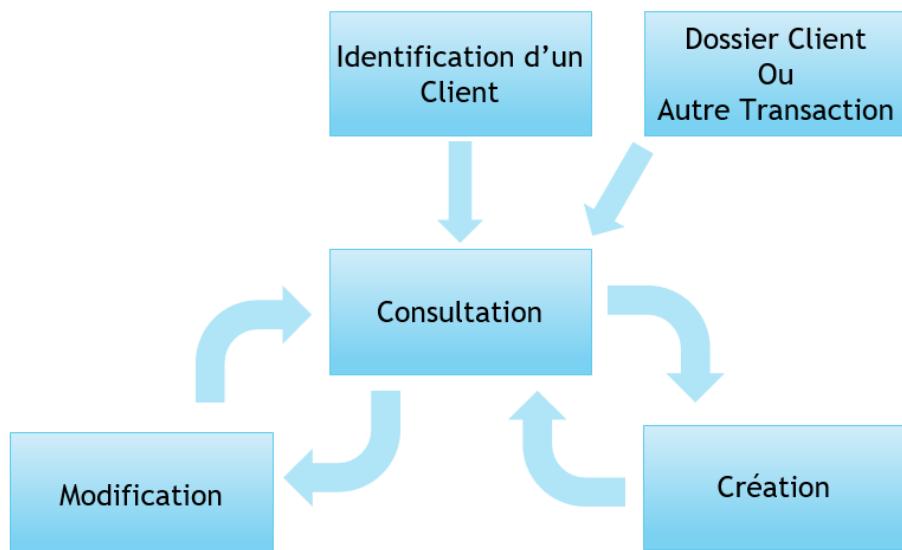
Un client a diverses possibilités pour passer un ordre Bourse France :

- En agence (SDO)
- En ligne (Net, smartphone)
- Par téléphone (Back Office ou GODE)

Type d'opération :

- Achat
- Vente
- Vendu / Acheté

Voici le modèle applicatif pour une transaction spirit en générale, qui s'applique également dans ce cas :



1.2 - La mission

Quelques temps après avoir été formée sur Spirit, il m'a été confié une tâche à reporter sur plusieurs transactions. Les transactions se ressemblent fortement je vais donc en détailler qu'une seule : **Vendre sur Bourse France**.

- 1) Premièrement j'ai dû effectuer une montée de version d'un service GOAL.
- 2) Dans un second temps, faire des modifications sur un composant Spirit, dans ce cas un tableau (uiTable).
- 3) Et pour finir rajouter/modifier des balises dans la zone éditique, qui gère l'impression.

Il est évident que je devais faire évoluer cette transaction et ne pas générer des régressions.

La première étape pour ma part était tout d'abord de comprendre la correctement la mission qui m'a été confiée. Pour cela, je me suis référée au DCIG qui m'a été fourni.

En cas d'évolution, de maintenance, etc... le DCIG permet de **retrouver les règles de gestion (RG)** déjà implémentées sans avoir à analyser le code. C'est d'autant plus intéressant quand il y a transfert de l'application ou changement de responsable.

Une fois la prise de connaissance finie, j'ai commencé par la montée de version du service GOAL. Pour cela, j'ai dû récupérer le nouveau stub grâce au code mnémonique qui m'a été fourni. Je me suis rendue sur l'URI fourni par le client. Un stub, est une doublure qui remplace un objet réel en fournissant des entrées indirectes à l'objet à tester, c'est une source JS de tous les schémas .json de validation. Pour des raisons de sécurité je n'ai malheureusement pas pu mettre d'exemple dans ce dossier. Une fois le fichier placé dans [app/js/services/stubs](#), j'ai modifié le fichier spirit-config qui permet l'appel des nouveaux services, qui eux aussi ont été modifiés grâce aux nouvelles données ajoutées voire supprimées, référencées dans le DCIG.

J'ai pris comme exemple le service créé :

```
<!-- app cree_bf -->
<service id="nego_bf_v13.13.0.cree_bf" type="goal">
    <param name="ims-transaction-code" value="YGOAV" />
    <param name="sdo-transaction-code" value="CGOVC" />
</service>
```

Le fichier package.json a également été mis à jour :

```
{
  "code": "CGOVC",
  "name": "CGOVC",
  "version": "0.0.47",
  "socle": "1.0.6",
  "theme": "v6",
  "services": [
    {
      "type": "goal",
      "stubs": [
        "1ug-1.0",
        "4c6-3.0",
        "4gv-3.0",
        "app-5.0"
      ]
    },
    {
      "type": "papyrus-pdf",
      "stubs": [
        "com.bnpp.spirit:papyrus.backend:1.4"
      ]
    }
  ]
}
```

Pour finir, les appels services ont été corrigé également pour qu'ils puissent s'effectuer correctement.

Une fois toute ses modifications mises en place, j'ai fait une première livraison sur l'environnement DEV, pour que les nouveaux services/modifications soient pris en compte.

Dans un deuxième temps, j'ai effectué les modifications sur un composant spirit, l'uiTable. Plus précisément, il m'a été demandé d'insérer de nouvelles lignes, un nouveau tableau et de modifier certains algorithmes déjà mis en place. Toujours pour des raisons de sécurités je ne vais pas pouvoir montrer l'intégralité du code.

Actuellement :

Frais estimés

	H.T. en EUR	T.V.A. en EUR	Impact sur le montant brut de l'opération
▼ Frais liés au service d'investissement et services auxiliaires	21,55	0,00	1,63 %
Courtage	21,55	0,00	
Rétrocessions perçues par la banque	BNP Paribas ne perçoit pas de rétrocession sur cet instrument financier		
▼ Frais liés à l'instrument financier	3,00	0,00	0,00 %
Taxe sur la transactions financières 0,2 %	2,00	0,00	
Frais indirects	1,00	0,00	
<hr/>			
En EUR			
Montant brut estimé de l'opération	1 500,00		
Montant estimé des frais TTC	24,55		
Montant net estimé de l'opération TTC	1 524,55		

Cible :

Frais estimés

Type de frais	Montant des frais (en EUR)	Frais en % du montant brut de l'opération
▼ Frais liés au service d'investissement et services auxiliaires	32,11	11,07
Courtage	32,11	
Frais de change	N/A	
▼ Rétrocessions perçues par la banque	0,33	0,11
▼ Frais liés à l'instrument financier	3,20	1,10
Frais sur les transactions financières 0,2 %	0,10	
Frais indirects	3,10	
Total des frais sur la 1ère année de détention	35,64	12,28
<hr/>		
Les frais calculés sont en TTC lorsque la TVA est applicable		
<hr/>		
	Total des frais en EUR	Total des frais en %
Année 1	35,64	12,28
Années suivantes	3,53	1,21
<hr/>		
Montant de l'opération avant les frais (estimé en EUR)	400,00	
Montant des frais liés à l'opération (estimé en EUR) prélevés à l'exécution	32,11	
Montant total prélevé (estimé en EUR)	432,11	

Comme vu ci-dessus, la ligne *rétrocession perçue par la banque*, ne fais plus partie des *frais liés au service d'investissement et services auxiliaires*. Et une icône de dossier est ajouté devant. De plus le tableau ne comporte plus 4 mais 3 colonnes.

Dans une fonction :

```
var prepareTreeGrid = function () {
    var createData = function () {
        var res = {
            data: []
        };
    };
}
```

Je déclare une variable de type objet, ses colonnes et d'autres paramètre attendu. Ici par exemple iconClass : icon-folder-yellow qui permet de mettre l'icône du dossier.

```
// on ajoute la ligne rétrocession perçue par la banque
var retroPercue = {
    typeFrais: ["span", "Rétrocessions perçues par la banque", "i", [
        id: "tooltipRetro",
        class: "treegrid-item-name icon-info vertical-bottom",
        "data-original-title": "Frais indirects récurrents, non prélevés sur le compte du client. Ils correspondent au montant versé par les producteurs"],
    montFrais: ["div", {"class": "text-center"}, col2or3value(data.g_inducement.b_aff_retr_perc_bg, column: "m_estmn_ttc_retrokes"),
        fraisMontBrutOpe: ["div", {"class": "text-center"}, col2or3value(data.g_inducement.b_aff_retr_perc_bg, column: "t_imp_retrokes"),
        hideInput: true,
        iconClass: "icon-folder-yellow",
        data: []
    ],
    res.data.push(retroPercue);
}
```

Dans la colonne montFrais et FraisMontBrutOpe, on peut voir qu'il y a une fonction qui se nomme `col2or3value`. C'est une fonction que j'ai mise en place car dans le DCIG fourni je me suis aperçue que beaucoup de données étaient testées de la manière suivante :

- | Colonne 2 |
|---|
| <ul style="list-style-type: none"> Si <code>b_aff_retr_perc_bg</code> est valorisé à '1'
le local restitue le contenu de la donnée <code>m_estmn_ttc_retrokes</code> Si <code>b_aff_retr_perc_bg</code> est valorisé à '2'
le local restitue 'NA' Si <code>b_aff_retr_perc_bg</code> est valorisé à '3'
le local restitue 'ND' |

De ce fait j'ai créé une fonction qui comporte un switch :

```
var col2or3value = function (b_aff, column) {
    var colValue;
    switch (b_aff) {
        case "1" :
            switch (column) {
                case "m_ttc_frais_service" :
                    colValue = App.Utils.toEditic.m_ttc_;
                    break;
                case "t_frais_service" :
                    colValue = App.Utils.toEditic.t_frai;
                    break;
                case "m_ttc_frais_tran" :
                    colValue = App.Utils.toEditic.m_ttc_;
                    break;
                case "m_ttc_frais_chng" :
                    colValue = App.Utils.toEditic.m_ttc_;
                    break;
                case "m_estmn_ttc_retrokes" :
                    colValue = App.Utils.toEditic.m_estmn;
                    break;
                case "t_imp_retrokes" :
                    colValue = App.Utils.toEditic.t_imp_;
                    break;
                case "m_ttc_frais_instr" :
                    colValue = App.Utils.toEditic.m_ttc_;
                    break;
                case "t_frais_instr" :
                    colValue = App.Utils.toEditic.t_frai;
                    break;
                case "m_ttc_tftf" :
                    colValue = Spirit.UtilNumber.format(1);
                    break;
                case "m_estmn_ttc_frais_imd" :
                    colValue = Spirit.UtilNumber.format(1);
                    break;
                case "m_tot_frais_1ere_annee" :
                    colValue = App.Utils.toEditic.m_tot_;
                    break;
                case "t_tot_frais_1ere_annee" :
                    colValue = App.Utils.toEditic.t_tot_;
                    break;
                default :
                    colValue = "";
            }
            break;
    }
}
```

En premier paramètre de la fonction `col2or3value`, je lui passe la donnée où j'attends une réponse du mainframe qui doit me renvoyer « 1 » « 2 » ou « 3 ». Si 1, j'affiche avec la donnée renseignée, si 2 ou 3 je renseigne avec NA ou ND. En deuxième paramètre je lui envoie une chaîne de caractères que j'ai défini dans les cases du switch, ce qui permet de renvoyer le bon montant ou taux.

```

    case "2" :
        colValue = "NA";
        break;
    case "3" :
        colValue = "ND";
        break;
    default :
        colValue = "";
    }
App.Utils.toEditic[column] = colValue;
return colValue;
};

```

Pour finir cette étape, à la fin je restitue toute ma ligne.

```
res.data.push(retroPercue);
```

Voici comment est créé le tableau des frais estimés :

```

var fraisEstimesTable = new Spirit.UiTTreeGrid("fraisEstimes_tree_container", {
    data: createData(),
    headers: [
        { "name": "Type de frais", "type": "string", "sortable": false, className: "col-sm-5 text-center" },
        { "name": "Montant des frais (en EUR)", "type": "number", "sortable": false, className: "col-sm-3 text-center" },
        { "name": "Frais en % du montant brut de l'opération", "type": "number", "sortable": false }
    ],
    arrayElement: ["typeFrais", "montFrais", "fraisMontBrutOpe"]
});
addTotalFraisLine();

```

Il comporte 3 colonnes, qui comprend un `name`, un `type`, si la colonne est triable ou non avec `sortable` et une `className` qui comprend une class bootstrap3 permettant d'ajuster la largeur des colonnes.

`arrayElement` permet de définir les colonnes plus simplement et est de type tableau [].

Dans l'appel des frais estimés : `App.Steps.fraisEstimes = function () {`

Je teste si je dois afficher les tableaux :

```

data = App.Data.dataOutControles;

var indicFraisNonEstimes = data.g_frais.b_aff_frais !== "1";
var indicBlocMessage = data.g_frais.b_aff_msg_frais_ind === "1" && data.g_frais.b_aff_frais === "1";

Spirit.Dom.populateDom(App.Tpl.tplFraisEstimes({ indicFraisNonEstimes: indicFraisNonEstimes, indicBlocMessage: indicBlocMessage }), "fraisEstimes_container");

// si les frais sont estimés
if (!indicFraisNonEstimes) {
    prepareTreeGrid();
    if (data.g_mentions.q_list_mentions > 0) {
        prepareMention5142();
    }
    prepareTableIllumImpactPerf();
    if (data.g_mentions.q_list_mentions > 0) {
        prepareMention3();
    }
    prepareTableMontantOpe();
}

```

Pour finir, j'ai également dû modifier dans le service éditique (qui gère l'impression), quelques balises. Voici un aperçu :

```
[ "NOM_SIGN_BANQ", App.userContext.nomCollaborateur || ""],  
[ "PRENOM_SIGN_BANQ", App.userContext.prenomCollaborateur || ""],  
[ "L_BIC_ADRP_1", adresse.bic.trim()],  
[ "TOP_OI", res.g_ordre_non_habit.i_ordre_non_habit.trim()],  
[ "N_VILLE_AGCE", adresse.ville2.trim()],  
[ "TOP_TAB_IMP", res.g_frais.b_aff_frais === 1 ? "O" : "N"]]
```

Les Châtaignettes

Une ancienne collègue, Léanna CHANIOL, m'a contacté pour savoir si j'étais intéressée de réaliser un site pour un de ses clients.

J'ai également travaillé sur ce projet avec ma partenaire Chloé KACZMAREK qui est également webdesigner.

J'ai commencé par établir une maquette, suite au brief envoyé par Léanna.

La maquette du site web est divisée en deux colonnes. La colonne gauche contient :

- Action harre**: Barre de navigation avec "Horaire d'ouverture popin" et "Contact".
- Biscuiterie Les Châtaignettes logo.
- Menu à voir: Nos boutiques, Contact.
- Navigation: Accueil, Les châtaignettes, Nos spécialités à la châtaigne, Les spécialités d'Ardèche.
- Les Châtaignettes**: Biscuits à la farine de châtaigne, Fabrication artisanale.
- Spécialités à la châtaigne** et **Spécialités ardéchoises**: Boutons "x" indiquant une redirection vers PA.
- VIDEO**: Bouton "Découvrez notre histoire" avec lien "/notre-histoire/".
- H1 - Biscuiterie Les Châtaignettes, le cœur savoureux de l'Ardèche**: Texte de présentation.
- Carte: Indication de la localisation avec les villes Saint-Péray, Bourg, Vallon Pont d'Arc.

La colonne droite contient :

- Une question ? Écrivez-nous !**: Bouton "/contact/".
- Section "picto": Quatre cases avec pictos et boutons "x".
- H2 - L'Ardèche, le Pays de la Châtaigne**: Texte de présentation.
- H2 - Visiter l'Ardèche c'est bien, la déguster c'est mieux**: Texte de présentation.
- H2 - Nos spécialités**: Texte de présentation.
- H3 - Gâteau à la châtaigne**, **H3 - Farine de châtaigne**, **H3 - Crème de marron**, **H3 - Confiture de châtaigne**: Sections avec des textes courts et boutons "x".

Par la suite, nous avons travaillé sous le CMS WordPress, avec l'aide du plugin BeTheme qui permet d'avoir une interface plus compréhensible pour le client que de simple bloc HTML.

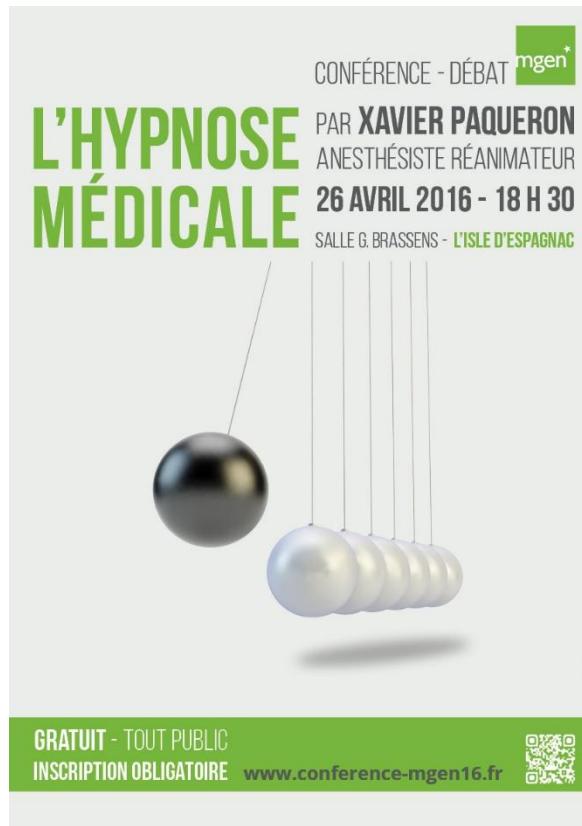
Voici le résultat de la page d'accueil du site : <http://www.les-chataignettes.fr/>

The screenshot displays the homepage of the website 'Les Chataignettes'. At the top, there's a navigation bar with links to 'Accueil', 'Nos biscuits', 'Spécialités à la châtaigne', 'Spécialités d'Ardèche', 'Nos recettes', and 'Contact'. The main header features the brand name 'Les Chataignettes' in a large, stylized font, with 'Spécialistes d'Ardèche' underneath. Below the header is a large image of a chestnut. A sidebar on the left contains a vertical menu for 'BISCUTERIE' with options like 'Biscuits à la farine de châtaigne', 'Farine de châtaigne', 'Crème de marron', and 'Confiture de châtaigne', along with a small image of a product and a quote: 'Il était une fois...'. The central content area has a section titled 'Biscuiterie Les Chataignettes, le cœur savoureux de l'Ardèche' with descriptive text about the brand's history and products. Below this are two images: one of the shop interior and another of shelves filled with products. At the bottom of the page, there are four icons representing different product categories: biscuits, farine, crème de marron, and confiture. The footer section includes a 'Qui sommes-nous?' (About Us) section with a video thumbnail, a 'Visiter l'Ardèche c'est bien, la déguster c'est mieux' (Visiting Ardèche is good, tasting it is better) section with a yellow flag graphic, and a 'Nos clients nous font confiance' (Our clients trust us) section featuring customer reviews with stars.

Conférence sur l'hypnose médicale

Pour valider ma deuxième année de DUT MMI, j'ai participé à un projet tuteuré avec un vrai client, dans ce cas présent la MGEN de Niort. En annexe de ce dossier se trouve le cahier des charges.

J'ai réalisé l'affiche et le carton d'invitation de cette conférence.



EAT THE WORD

Avec deux de mes camarades de U-DEV, Jimmy BOUMANS et Damien ECKLY, nous avons eu pour idée de concevoir un site e-commerce de A à Z. Ce projet n'a pas abouti pour faute de temps mais nous avions tout de même conçu un MCD un script SQL. Voici ma version du script.

```
#-----
#      Script MySQL.
#-----

/*CREATE DATABASE ecommerce;*/

USE ecommerce;

#-----
# Table: categorie
#-----


CREATE TABLE categorie(
    id Int Auto_increment NOT NULL,
    libelle_categorie Varchar (250) NOT NULL,
        CONSTRAINT categorie_PK PRIMARY KEY (id)
);

#-----
# Table: tva
#-----


CREATE TABLE tva(
    id Int Auto_increment NOT NULL,
    taux Decimal (3,2) NOT NULL,
    actif INT NOT NULL,
        CONSTRAINT tva_PK PRIMARY KEY (id)
);

#-----
# Table: type_adresse
#-----


CREATE TABLE type_adresse(
    id Int Auto_increment NOT NULL,
    libelle Varchar (50) NOT NULL,
        CONSTRAINT type_adresse_PK PRIMARY KEY (id)
);

#-----
# Table: civilite
#-----


CREATE TABLE civilite(
    id Int Auto_increment NOT NULL,
    libelle Varchar (30) NOT NULL,
        CONSTRAINT civilite_PK PRIMARY KEY (id)
```

```

);

#-----
# Table: cp_ville
#-----

CREATE TABLE cp_ville(
    id Int Auto_increment NOT NULL,
    code Char (5) NOT NULL,
    ville Varchar (250) NOT NULL,
    CONSTRAINT cp_ville_PK PRIMARY KEY (id)
);

#-----
# Table: adresse
#-----


CREATE TABLE adresse(
    id Int Auto_increment NOT NULL,
    ligne1 Varchar (250) NOT NULL,
    ligne2 Varchar (250),
    complement Varchar (250),
    id_cp_ville Int NOT NULL,
    CONSTRAINT adresse_PK PRIMARY KEY (id),
    CONSTRAINT adresse_cp_ville_FK FOREIGN KEY (id_cp_ville) REFERENCES
cp_ville(id)
);

#-----
# Table: moyen_paiement
#-----


CREATE TABLE moyen_paiement(
    id Int Auto_increment NOT NULL,
    valeur Varchar (250) NOT NULL,
    libelle Varchar (250) NOT NULL,
    CONSTRAINT moyen_paiement_PK PRIMARY KEY (id)
);

#-----
# Table: reduction
#-----


CREATE TABLE reduction(
    id Int Auto_increment NOT NULL,
    Valeur_Pourcentage Float NOT NULL,
    Valeur_Reduction Varchar (150) NOT NULL,
    Code_Promo Varchar (150) NOT NULL,
    CONSTRAINT reduction_PK PRIMARY KEY (id)
);

#-----
# Table: marque
#-----
```

```

CREATE TABLE marque(
    id Int Auto_increment NOT NULL,
    nom_marque Varchar (150) NOT NULL,
    CONSTRAINT marque_PK PRIMARY KEY (id)
);

#-----
# Table: produit
#-----

CREATE TABLE produit(
    id Int Auto_increment NOT NULL,
    reference_produit Varchar (10) NOT NULL,
    designation_produit Varchar (250) NOT NULL,
    prix_unitaire_ht_produit Double NOT NULL,
    quantite_stock_produit Int NOT NULL,
    est_dispo INT NOT NULL,
    date_dispo Date,
    id_categorie Int NOT NULL,
    id_reduction Int,
    id_marque Int NOT NULL,
    CONSTRAINT produit_PK PRIMARY KEY (id),
    CONSTRAINT produit_categorie_FK FOREIGN KEY (id_categorie) REFERENCES
categorie(id),
    CONSTRAINT produit_reduction0_FK FOREIGN KEY (id_reduction) REFERENCES
reduction(id),
    CONSTRAINT produit_marque1_FK FOREIGN KEY (id_marque) REFERENCES
marque(id)
);

#-----
# Table: mot_de_passe
#-----

CREATE TABLE mot_de_passe(
    id Int Auto_increment NOT NULL,
    mot_de_passe Varchar (300) NOT NULL,
    CONSTRAINT mot_de_passe_PK PRIMARY KEY (id)
);

#-----
# Table: client
#-----

CREATE TABLE client(
    id Int Auto_increment NOT NULL,
    nom Varchar (50) NOT NULL,
    prenom Varchar (50) NOT NULL,
    date_de_naissance Date NOT NULL ,
    id_civilite Int NOT NULL,
    id_mot_de_passe Int NOT NULL,
    CONSTRAINT client_PK PRIMARY KEY (id),
    CONSTRAINT client_civilite_FK FOREIGN KEY (id_civilite) REFERENCES
civilite(id),

```

```

        CONSTRAINT client_mot_de_passe0_FK FOREIGN KEY (id_mot_de_passe)
REFERENCES mot_de_passe(id)
);

#-----
# Table: devis
#-----

CREATE TABLE devis(
    id Int Auto_increment NOT NULL,
    date Date NOT NULL,
    duree_valide Date NOT NULL,
    est_panier int NOT NULL,
    est_liste_souhait int NOT NULL ,
    id_client Int NOT NULL,
        CONSTRAINT devis_PK PRIMARY KEY (id),
        CONSTRAINT devis_client_FK FOREIGN KEY (id_client) REFERENCES client(id)
);

#-----
# Table: ligne_devis
#-----


CREATE TABLE ligne_devis(
    id Int Auto_increment NOT NULL,
    description_ligne_devis Varchar (250) NOT NULL,
    quantite_ligne_devis Int NOT NULL,
    id_produit Int NOT NULL,
    id_tva Int NOT NULL,
    id_devis Int NOT NULL,
        CONSTRAINT ligne_devis_PK PRIMARY KEY (id),
        CONSTRAINT ligne_devis_produit_FK FOREIGN KEY (id_produit) REFERENCES
produit(id),
        CONSTRAINT ligne_devis_tva0_FK FOREIGN KEY (id_tva) REFERENCES tva(id),
        CONSTRAINT ligne_devis_devis1_FK FOREIGN KEY (id_devis) REFERENCES
devis(id)
);

#-----
# Table: facture
#-----


CREATE TABLE facture(
    id Int Auto_increment NOT NULL,
    date Date NOT NULL,
    est_paye INT NOT NULL,
    id_devis Int NOT NULL,
    id_adresse Int NOT NULL,
    id_client Int NOT NULL,
        CONSTRAINT facture_PK PRIMARY KEY (id),
        CONSTRAINT facture_devis_FK FOREIGN KEY (id_devis) REFERENCES devis(id),
        CONSTRAINT facture_adresse0_FK FOREIGN KEY (id_adresse) REFERENCES
adresse(id),
        CONSTRAINT facture_client1_FK FOREIGN KEY (id_client) REFERENCES
client(id),
        CONSTRAINT facture_devis_AK UNIQUE (id_devis)

```

```

);

#-----
# Table: ligne_facture
#-----

CREATE TABLE ligne_facture(
    id Int Auto_increment NOT NULL,
    description Varchar (250) NOT NULL,
    quantite_ligne_facture Int NOT NULL,
    id_produit Int NOT NULL,
    id_tva Int NOT NULL,
    id_facture Int NOT NULL,
        CONSTRAINT ligne_facture_PK PRIMARY KEY (id),
        CONSTRAINT ligne_facture_produit_FK FOREIGN KEY (id_produit) REFERENCES
produit(id),
        CONSTRAINT ligne_facture_tva0_FK FOREIGN KEY (id_tva) REFERENCES
tva(id),
        CONSTRAINT ligne_facture_facture1_FK FOREIGN KEY (id_facture) REFERENCES
facture(id)
);

#-----
# Table: moyen_comm
#-----


CREATE TABLE moyen_comm(
    id Int Auto_increment NOT NULL,
    valeur Varchar (250) NOT NULL,
    libelle Varchar (250) NOT NULL,
    id_client Int NOT NULL,
        CONSTRAINT moyen_comm_PK PRIMARY KEY (id),
        CONSTRAINT moyen_comm_client_FK FOREIGN KEY (id_client) REFERENCES
client(id)
);

#-----
# Table: commentaire
#-----


CREATE TABLE commentaire(
    id Int Auto_increment NOT NULL,
    titre Varchar (30) NOT NULL,
    description Varchar (250) NOT NULL ,
    id_client Int NOT NULL,
    id_produit Int NOT NULL,
        CONSTRAINT commentaire_PK PRIMARY KEY (id),
        CONSTRAINT commentaire_client_FK FOREIGN KEY (id_client) REFERENCES
client(id),
        CONSTRAINT commentaire_produit0_FK FOREIGN KEY (id_produit) REFERENCES
produit(id)
);

#-----
# Table: commande
#-----
```

```

#-----#
CREATE TABLE commande(
    id Int Auto_increment NOT NULL,
    demande_retour_commande TinyINT NOT NULL,
    id_client Int NOT NULL,
    id_facture Int NOT NULL,
        CONSTRAINT commande_PK PRIMARY KEY (id),
        CONSTRAINT commande_client_FK FOREIGN KEY (id_client) REFERENCES
client(id),
        CONSTRAINT commande_facture0_FK FOREIGN KEY (id_facture) REFERENCES
facture(id)
);

#-----#
# Table: POSSEDER
#-----#

CREATE TABLE POSSEDER(
    id Int NOT NULL,
    id_adresse Int NOT NULL,
    id_client Int NOT NULL,
        CONSTRAINT POSSEDER_PK PRIMARY KEY (id,id_adresse,id_client),
        CONSTRAINT POSSEDER_type_adresse_FK FOREIGN KEY (id) REFERENCES
type_adresse(id),
        CONSTRAINT POSSEDER_adresse0_FK FOREIGN KEY (id_adresse) REFERENCES
adresse(id),
        CONSTRAINT POSSEDER_client1_FK FOREIGN KEY (id_client) REFERENCES
client(id)
);

#-----#
# Table: LIVRER
#-----#

CREATE TABLE LIVRER(
    id Int NOT NULL,
    id_facture Int NOT NULL,
    date Date NOT NULL,
    preuve Char (10) NOT NULL,
    est_livre INT NOT NULL,
        CONSTRAINT LIVRER_PK PRIMARY KEY (id,id_facture),
        CONSTRAINT LIVRER_adresse_FK FOREIGN KEY (id) REFERENCES adresse(id),
        CONSTRAINT LIVRER_facture0_FK FOREIGN KEY (id_facture) REFERENCES
facture(id)
);

#-----#
# Table: UTILISER
#-----#

CREATE TABLE UTILISER(
    id Int NOT NULL,
    id_client Int NOT NULL,
    date Date NOT NULL,
        CONSTRAINT UTILISER_PK PRIMARY KEY (id,id_client),

```

```
CONSTRAINT UTILISER_moyen_paiement_FK FOREIGN KEY (id) REFERENCES
moyen_paiement(id),
CONSTRAINT UTILISER_client0_FK FOREIGN KEY (id_client) REFERENCES
client(id)
);
```

ANNEXES

FICHE TUTEUR

CERTIFICATION PROFESSIONNELLE
EVALUATION ENTREPRISESTAGIAIRE

Nom : BENQUET
 Prénom : Emeline
 Certification visée : CDAN

ENTREPRISE

Nom de la société : CGI
 Tuteur(trice) : Anthony DARDILHAC - Carole LAUGA
 Courriel : anthony.dardilhac@cgi.com
 carole.lauga@cgi.com

«tuteur_Civilite»,

Afin de parfaire la validation de la certification professionnelle de votre stagiaire, nous vous remercions de remplir exhaustivement cet original et nous le retourner dans les meilleurs délais. Ce document sera examiné par le jury en vue de l'attribution de la certification professionnelle de votre stagiaire.

L'intégration dans l'équipe de travail est-elle satisfaisante ?

Oui / Non

Le comportement en situation professionnelle est-il adapté à la culture de l'entreprise ?

Oui / Non

Le comportement relationnel est-il adapté au métier visé ?

Oui / Non

La capacité de travail fournie correspond-elle au niveau d'un professionnel du métier ?

Oui / Non

Le niveau de responsabilités attendu est-il satisfait ?

Oui / Non

Des questions sont-elles posées à bon escient ?

Oui / Non

Des difficultés en relation avec le niveau de responsabilités sont-elles surmontées ?

Oui / Non

Le niveau d'obligation de réserve demandé est-il pris en compte ?

Oui / Non

Les activités effectuées concourent-elles à la couverture du champ de compétences ?

Oui / Non

Le métier associé à la certification visée est-il compris ?

Oui / Non

Le stagiaire est-il apte à occuper la fonction visée même comme débutant ?

Oui / Non

Appréciation générale suite au stage en entreprise :

Emeline s'est très bien intégrée à l'équipe.
 Elle a su monter en compétences tout au long de son alternance.

Fait à LeHaillan , le 31/10/19



Signature du tuteur
et cachet de l'entreprise

Signature du stagiaire

Association ADIP régie
par la loi du 1^{er} juillet 1901
Siret : 409 801 677 00017 Code
APE : 85.59 A
N° organisme : 11 75 27 97 775

Centre de formation
44 bis, quai de Jemmapes 75010
Paris
Siège social
12, rue Alexandre Parodi 75010 Paris

Tél : 01 80 97 35 00
mlabhiti@groupe-igs.fr
www.ipi-ecoles.com

GROUPE IGS

© Institut de Poly-Informatique (2011)

CV



EMELINE BENQUET

DÉVELOPPEUSE / WEBDESIGNER

Au-delà d'être un métier, le secteur du numérique est une réelle passion. Ma curiosité naturelle et ma motivation me donnent envie de m'investir dans une nouvelle formation, ainsi qu'une nouvelle entreprise.

EXPÉRIENCES

2019 | CGI - Développeuse IHM (1an)

Développement IHM pour un client bancaire - Framework JS

2018 | Solocal Marketing Service - Webdesigner (2an)

Conception et intégration de maquette sur WordPress - relation client

2016 | IUT de Poitiers - Appel à projet LISEA (2 mols)

Conception charte graphique de l'application web Social Taxi

2013 - 2015 | Intermarché Bazas - Travail d'été

Hôtesse de caisse et mise en rayon

2014 - 2015 | Studio Efflorescence (2mols)

Création de : Flyers - Carte de visites

Affiches - Vitrine - Mise en page de magazine

2013 (2mols) | StickerzLab

Création de : Stickers - Bannières de site web - Création vitrine

Référencement - Gestion WordPress - Magento - Photographie

2013 | La Sodal - Imprimerie

Connaissance de la chaîne graphique

FORMATION

2018 - 2019 | EPSI Bordeaux

Formation UDEV - Licence Concepteur Développeur d'Application numérique

2014 - 2016 | IUT Angoulême

DUT Métiers du Multimédia et de l'Internet

2011 - 2014 | Lycée Saint Vincent de Paul - Bordeaux

Bac Pro. « Communication Visuelle option Plurimédia »

COMPÉTENCES

Logiciel de création

Photoshop - Illustrator - InDesign - After Effects - Final Cut Pro X

Bureautique

Powerpoint - Word - Excel - Utilisation Google drive

Langages

HTML / CSS - JS - JQuery - PHP (notion) - JAVA (débutante) - SQL

CONTACT

27 rue des écus
33110 Le bouscat
emeline.benquet@gmail.com

06 58 76 57 30
24 ans - Permis B

INTÉRÊTS



Basket



Fifre



Longboard



Musique - Festival

Contactez-moi !

Tableau de compétences

Commencer par le référentiel :

Bloc de compétences : Qualité et sécurisation du code réalisé

Compétences ou capacités qui seront évaluées	Critères d'évaluation	Exemples d'activités et tâches	Activités pratiquées	Origine de l'acquisition	Preuves apportées & ref. annexe
- Formaliser, identifier les résultats attendus.	La liste de contrôle des attendus fonctionnels est paraphée.	Étude de l'existant. Rédaction du cahier des spécifications fonctionnelles.	Rédaction d'un cahier des charges pour l'organisation d'une conférence sur l'hypnose médicale.	F(S)	Annexes p.57
Respecter des contraintes.	Un plan d'assurance qualité est observé.	Conception/architecture d'applications logicielles. Conception de services métiers.	Conception / architecture d'application logicielle	E(A)	p.37
Respecter les recommandations qualité de la norme en vigueur pour l'architecture des logiciels.	L'application est organisée en couches indépendantes.	Conception de services d'accès aux données.	Conception de services d'accès aux données	E(A)	p.36
Anticiper les évolutions.	Les règles métier sont encapsulées dans des services logiciels.	L'accès aux données est réalisé par des services logiciels indépendants du mode de stockage.	Détermination du nombre de tiers de l'application.		
Qualifier les risques	L'exécution de l'application est répartie				

IPI - Comment faire mon dossier de validation ?

CDAN

Respecter une norme de présentation des écrans et documents de sortie.	<p>entre un nombre d'ordinateurs adapté au contexte.</p> <p>Un formulaire d'estimation des risques est rempli.</p> <p>Une norme de présentation des données est respectée.</p> <p>Les interfaces Homme/Machine sont validées.</p>	<p>Réalisation d'une interface homme/machine (IHM)</p> <p>Estimation, qualification des risques sécurité.</p> <p>Réalisation des maquettes de sorties interactives.</p> <p>Réalisation des maquettes de sortie imprimée.</p>	<p>Réalisations d'une maquette pour un site internet sous WordPress.</p>	B(S)	p.38
Concevoir des programmes avec une orientation objets.	<p>Une programmation orientée objets est utilisée.</p> <p>Garantir un accès sécurisé aux données.</p>	<p>Programmation de logiciels.</p> <p>Le taux de réutilisation du code utile est > 80 %.</p> <p>Des gabarits sont utilisés.</p> <p>Une charte de nommage est utilisée.</p>	<p>Réalisation d'une affiche pour une conférence sur l'hypnose médicale.</p>	F(S)	p.40
			<p>Programmation de logiciels gérant des objets en JSON.</p>	E(A)	p.64

IPI - Comment faire mon dossier de validation ?

CDAN

			E(A)	
Livrer le logiciel déterminé.	<p>Le taux de documentation interne du code est > 8 % et < 15 %.</p> <p>Les anomalies d'accès aux données ne génèrent pas d'interruption de l'exécution et sont réperforées.</p> <p>Livrer le logiciel conforme aux attentes.</p>	<p>Programmation de l'accès aux données de l'entreprise.</p> <p>Tests unitaires.</p> <p>Préparation des jeux de tests.</p> <p>Des outils de contrôle automatique du code sont utilisés.</p> <p>Aucun défaut visible ne persiste.</p>	<p>Test unitaire, Règles du DCIG</p> <p>E(A)</p> <p>Utilisation de Grunt : un outil pour la création de tâches automatisées avec le langage JavaScript.</p> <p>E(A)</p> <p>Validation d'une étape dans le projet</p> <p>E(A)</p>	<p>p.34</p> <p>p.65</p> <p>p.20</p> <p>p.38</p>
	<p>Clôturer une mission.</p>	<p>Les contraintes spécifiques au projet sont respectées.</p> <p>Un manuel d'assurance qualité est respecté.</p> <p>Une méthode de recettage est utilisée.</p> <p>L'étape du projet est validée.</p> <p>Le PV de réception du logiciel est validé.</p>	<p>Respect des règles de gérance écrivent dans le DCIG.</p> <p>E(A)</p>	<p>p.21</p>
		<p>Mise en exploitation.</p> <p>Livraison du projet</p>	E(A)	

IPI - Comment faire mon dossier de validation ?

CDAN

Bloc de compétences : Audit, conception, méthode de projet



Compétences ou capacités qui seront évaluées	Critères d'évaluation	Exemples d'activités et tâches	Activités pratiquées	Origine de l'acquisition	Preuves apportées & ref. annexe
Formaliser des processus	<p>La procédure du service utilisateur est formalisée et validée.</p> <p>La procédure du service utilisateur est conforme aux règles du système de management des services de l'entreprise.</p> <p>La circulation du document résultat du traitement prévu est matérialisée dans un diagramme de workflow.</p> <p>Formaliser les règles de gestion et d'organisation des données de l'entreprise.</p>	<p>Étude de l'existant. Identification des procédures en place.</p> <p>Contrôle de la conformité des procédures utilisées avec la gouvernance de l'entreprise.</p> <p>Recensement des documents utilisés, identification de leur circulation et des acteurs concernés.</p> <p>Reconfiguration de procédure.</p> <p>La proposition de reconstruction de la procédure est validée.</p> <p>La base de données est modélisée.</p>	<p>Étude de l'existant. Identification des procédures en place.</p> <p>Contrôle de la conformité des procédures utilisées avec la gouvernance de l'entreprise</p> <p>Respect des règles de gestions édicté dans le DCG.</p> <p>Conception d'une base de données.</p> <p>La base de données est modélisée.</p>	<p>E(A)</p> <p>E(A)</p> <p>E(A)</p> <p>F(M)</p>	<p>p.34</p> <p>p.34</p> <p>p.65</p> <p>p.45</p>

IPI - Comment faire mon dossier de validation ?

CDAN	
Une méthode de conception par objets est utilisée.	Concevoir des éléments logiciels réutilisables.
Une méthode AGILE est utilisée.	Produire du logiciel en équipe. Programmation en équipe. Écriture de code.
Absence de signaux d'alertes au point de contrôle du projet.	Remonter les alertes au(x) décideur(s).
Les étapes du projet sont planifiées.	Estimer des délais.
Le projet est conforme au schéma directeur de l'entreprise et respecte les principes d'urbanisation du S.I.	Concevoir une solution logicielle.
Les spécifications fonctionnelles produites respectent le cahier des charges fourni.	Conception de la solution logicielle.
L'impact de modification est acceptable.	Anticiper des répercussions.
	Conception de l'architecture applicative. Programmation en équipe et écriture de code.
	Coordination de l'avancement.
	Planification des tâches du projet.
	Méthodes pseudo agiles Imputation en jour homme
	Conception de la solution logicielle. Ex. : Montée de version v6 vers sprint. Conforme aux spécificités attendue par le framework et écrite dans le DCIG
	E(M) E(A)
	E(S)
	E(M) p.34

Bloc de compétences : Réalisation d'applications logicielles

Compétences ou capacités qui seront évaluées	Critères d'évaluation	Exemples d'activités et tâches	Activités pratiquées	Origine de l'acquisition	Preuves apportées & ref. annexe
Encapsuler des solutions logicielles spécifiques dans des services logiciels génériques.	Le service d'accès aux données est opérationnel.	Programmation. Investigations documentaires fonctionnelles ou techniques complémentaires. Transcription des spécifications fonctionnelles en algorithmes.	Programmation Investigation documentaires fonctionnelles/technique complémentaire.	E(A)	p.34
Produire du logiciel générique réutilisable.	Des services logiciels internes sont réutilisables.	Transcription des algorithmes en code source. Compilation, déverrouillage du code source.	Transcription de spécification fonctionnelles en algorithmes.	E(A)	p.36
Produire du logiciel partageable.	Des services logiciels sont partageables en local.	Agglomération des différents éléments logiciels en unités de traitement, réalisation des tests unitaires.	Transcription des algorithmes en code source. Compilation, déverrouillage du code source.	E(A)	p.36
Intégrer des éléments logiciels hétérogènes et produire des exécutables livrables.	Des services logiciels sont partageables à distance.	Le logiciel est livrable, prêt pour la mise en production.		Mise à jour du planning de réalisation.	Mise à jour du planning de réalisation
Modifier un algorithme sans générer de dysfonctionnements.	La modification n'entraîne pas de régression fonctionnelle.			E(A)	
Contrôler des délais.	Le compte-rendu d'activité est renseigné,				

IP1 - Comment faire mon dossier de validation ?

CDAN

	les écarts sont constatés.		
--	----------------------------	--	--

Bloc de compétences : Communiquer avec les acteurs du projet

Compétences ou capacités qui seront évaluées	Critères d'évaluation	Exemples d'activités et tâches	Activités pratiquées	Origine de l'acquisition	Preuves apportées & ref. annexe
User d'une communication professionnelle tant en français qu'en anglais.	Le compte-rendu de la réunion est validé. Le score du TOEIC est > 749	Élaboration et rédaction de documents techniques, commerciaux ou internes à destination, des utilisateurs, des clients ou des collaborateurs, ...	Anglais : niveau B2 TOEIC < 749 Recherches de solutions de code en anglais car souvent plus de solutions.	F E	
Interagir efficacement dans un environnement de travail collaboratif.	Le document collectant l'expression des besoins des utilisateurs est validé.	Rédaction des spécifications fonctionnelles de la solution informatique.	Rédactions des spécifications fonctionnelles de la solution informatique.	E(S)	
	L'aide du logiciel est rédigée.	Écriture des interfaces homme/machine.	Relations avec les clients.	B(A)	
	La documentation du livrable est diffusée.	Animation de réunions de travail et interviews d'utilisateurs.	Réunion téléphonique pour la conception d'un site internet avec le client.	p.38	
	La présentation est appréciée.	Démonstrations, recettage de livrables.	Démonstration du site, site validé.		

IPI - Comment faire mon dossier de validation ?

CDAN

		Formation des utilisateurs au logiciel.	Formation des utilisateurs au logiciel.	B(A)	p.38
Réaliser des échanges de données informatisées (EDI). Automatiser des traitements.	Les utilisateurs sont opérationnels, le transfert des nouvelles compétences est validé.	<p>Réalisation d'un procédé d'échange de données informatisées.</p> <p>Rétro-documentation de logiciels et de bases de données.</p> <p>Consolidation, agrégation de données.</p> <p>Programmation de l'interface d'échange de données.</p> <p>Réalisation d'un environnement de tests.</p> <p>Création, configuration de machines virtuelles.</p> <p>Installation, configuration de serveurs d'applications, Web et base de données.</p> <p>Écriture de scripts systèmes pour adapter l'environnement d'exécution.</p>	<p>Réalisation de bouchons pour résoudre des anomalies</p>	E(A)	p.64

IPI - Comment faire mon dossier de validation ?

Cahier des charges



BENQUET Emeline
BOURIM Lamia
TURC Maud

CAHIER DES CHARGES

Conférence débat sur l'hypnose médicale

Tuteur IUT

Bernadette CHAULLET

Tuteurs extérieur

Mickaël BITEAU et Valérie DECROIX



SOMMAIRE

1. Présentation d'ensemble du projet

1.1	Présentation de l'entreprise :	4
1.2	Les objectifs du site :	4
1.3	La cible adressée par le site :	4
1.4	Objectifs quantitatifs :	4
1.5	Périmètre du projet :	4

2. Description graphique et ergonomique

2.1	Charte graphique :	5
2.2	Design :	5

3. Description fonctionnelle et technique

3.1	Arborescence du site :	5
3.2	Description fonctionnelle :	5
3.3	Contraintes techniques :	5
3.4	Hébergement :	6

4. Prestations attendues et modalités de sélection des prestataires

4.1	Prestations attendues :	6
4.2	Planning :	6
4.3	Méthodologie de suivi :	6



1. Présentation d'ensemble du projet

1.1 Présentation de l'entreprise :

- La MGEN est une mutuelle basée à Niort. Elle gère le régime obligatoire d'assurance maladie des professionnels de l'Éducation nationale, de l'Enseignement supérieur, de la Recherche, de la Culture et de la Communication, de la Jeunesse et des Sports. La MGEN organise depuis quelques années des conférences-débat. Chaque année un thème différent est proposé. Cette année, ce sera le thème de l'hypnose médicale. La conférence débat aura lieu à Niort dans le début du mois de juin 2016.
- Pour que la conférence débat sur l'hypnose médicale ait lieu et soit un évènement réussi, il faut faire venir du monde. Il faut donc créer une communication autour de cet évènement. Pour cela, il faut réaliser : des affiches, des cartons d'invitation et un site internet "one page". Il faut aussi pouvoir gérer les inscriptions à la conférence qui se feront en ligne à l'aide d'e-billet que le client présentera à son entrée lors de la conférence.

1.2 Les objectifs du site :

- Le site web a pour principal objectif de permettre une inscription en ligne pour faire venir un maximum de personnes à la conférence et simplifier la gestion de celle-ci. Cette inscription se devra d'être simple pour être accessible à la cible. Cela doit permettre à la MGEN d'élargir son public qui initialement était majoritairement constitué d'enseignants.
- Mais ce n'est pas son seul objectif, il doit donner des renseignements sur les conférenciers, ainsi que sur le thème de la conférence.

1.3 La cible adressée par le site :

- Les adhérents à la MGEN sont les premiers ciblés par cette conférence. Ce sont eux qui reçoivent des cartons d'invitation.
- Ils sont en majorité professeurs ou bien chercheurs. La tranche d'âge la plus visée est celle des plus de 60 ans, mais il serait bien d'attirer également les plus jeunes (35 à 50 ans).

1.4 Objectifs quantitatifs :

- Son ambition est aujourd'hui d'accueillir 900 personnes présentant une forte mixité « générationnelle et sociale ».

1.5 Périmètre du projet :

- Le site n'a pas besoin d'être multi langues, il vise un public français, adhérant à la MGEN ou dans les alentours de Niort.
- Il faut que le site soit adapté pour les supports mobiles, cela est un gain de temps pour les internautes qui ont un smartphone.



2. Description graphique et ergonomique

2.1 Charte graphique :

- Pour ce qui est des éléments de la charte graphique la seule obligation est de faire apparaître le logo de la MGEN.

2.2 Design :

- Il n'y a pas de contraintes typographiques, de visuel ou encore de couleur. Cela doit être en harmonie avec le thème de la conférence.

3. Description fonctionnelle et technique

3.1 Arborescence du site :

- Le site sera un « one page » c'est-à-dire un site constitué d'une seule page web sur laquelle est présente l'ensemble des contenus d'un site web. Cette page est divisée en plusieurs parties, dont les différents contenus sont accessibles en scrollant verticalement (ou à l'aide de liens internes ancrés).
- La section la plus importante est celle de l'inscription / dé-inscription (permettant d'annuler l'inscription) à la conférence, l'internaute doit y avoir accès rapidement, sans chercher. La deuxième section est celle consacrée aux intervenants, car ce sont pour eux que les internautes se déplacent. Il pourra y en avoir une troisième comprenant une petite vidéo sur le thème de l'hypnose médicale.

3.2 Description fonctionnelle :

- Le site doit permettre aux internautes, souhaitant aller à la conférence sur l'hypnose médicale, de s'inscrire en ligne grâce un formulaire. Une fois ce formulaire rempli correctement, il générera un e-billet au format pdf. L'internaute recevra un e-mail contenant son e-billet, qu'il présentera par la suite à son arrivée à la conférence, que ce soit par le biais de son smartphone ou alors d'un papier qu'il aura imprimé.

3.3 Contraintes techniques :

- Le site sera un « one page » comme dit précédemment, il devra également être responsive design, c'est-à-dire s'adapter à tous supports numériques que ce soit ordinateurs, tablettes ou smartphones.
- Mettre en place le développement des programmes utiles à l'accueil du public (scan des e-billets ou recherche de la personne si absence de e-billet).
- Si possible le site doit être consultable sur l'ensemble des navigateurs, ou du moins les plus côtoyés comme Google chrome, Mozilla firefox, Internet explorer.
- Faire une proposition pour une stratégie de communication via les réseaux sociaux.



3.4 Hébergement :

- Un nom de domaine existe, c'est celui de la conférence de l'année précédente :
<http://www.stid-niort.fr/mgen79>
- La MGEN possède un site internet : www.mgen.fr.
- Les ressources disponibles sont donc : le site web de la MGEN, le site de la dernière conférence qu'a donné la MGEN sur le thème du sommeil l'année dernière.

4. Prestations attendues et modalités de sélection des prestataires

4.1 Prestations attendues :

Voici les prestations attendues :

- Élaboration de la charte graphique de la conférence
- Création d'une maquette pour l'affiche et le carton d'invitation
- Développement du site web, de l'inscription en ligne avec envoi d'e-billet individuels (graphisme en cohérence avec l'affiche)
- Développement des programmes utiles à l'accueil du public (scan des e-billets ou recherche de la personne si absence de e-billet)
- Mettre en place une stratégie de communication

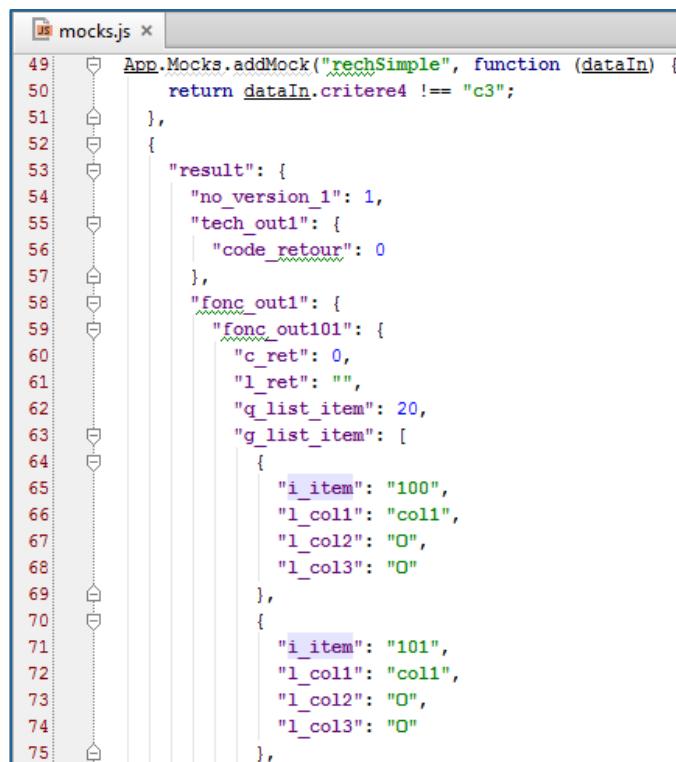
4.2 Planning :

- Une pré-soutenance aura lieu le jeudi 7 janvier 2016 pour présenter le projet et voir son avancement.
- Tous les mois un rendez-vous sera fixé pour voir l'avancée du projet.
- Le projet devra être livré mi-avril.

4.3 Méthodologie de suivi :

- On adoptera une méthodologie de développement agile.
- Pour ce qui est de l'organisation du projet, nous auront des contacts par mail, téléphone et devrons fixer quelque rendez-vous sur Niort et sur Angoulême.

Exemple Bouchon



```
49     App.Mocks.addMock("rechSimple", function (dataIn) {
50         return dataIn.criteres4 !== "c3";
51     },
52     {
53         "result": {
54             "no_version_1": 1,
55             "tech_out1": {
56                 "code_retour": 0
57             },
58             "fong_out1": {
59                 "fong_out101": {
60                     "c_ret": 0,
61                     "l_ret": "",
62                     "q_list_item": 20,
63                     "q_list_item": [
64                         {
65                             "i_item": "100",
66                             "l_col1": "col1",
67                             "l_col2": "O",
68                             "l_col3": "O"
69                         },
70                         {
71                             "i_item": "101",
72                             "l_col1": "col1",
73                             "l_col2": "O",
74                             "l_col3": "O"
75                         }
76                     ]
77                 }
78             }
79         }
80     }
81 );
```

DCIG

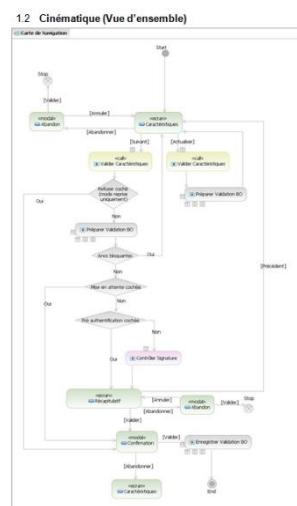
Rôle du DCIG dans la phase de développement des IHM :

- Un développeur, surtout IHM, n'a pas nécessairement les connaissances fonctionnelles et ne devrait, généralement, pas en avoir besoin. Il faut donc lui transmettre un support lui permettant de **réaliser l'application dans les meilleures conditions**.
 - En cas d'évolution, de maintenance, etc... le DCIG permet de **retrouver les règles de gestion (RG)** déjà implémentées sans avoir à analyser le code. C'est d'autant plus intéressant quand il y a transfert de l'application ou changement de responsable.

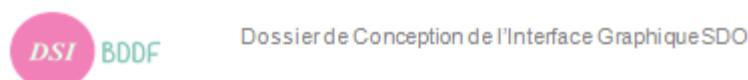
Rappeler en quelques lignes le rôle de l'application.



- . Toute la cinématique avec les différents écrans doit être décrite.
 - Les appels métiers et fenêtres modales doivent être tous représentés



Les écrans :



2.3 Ecran / Fonctionnalité XXXX

[Renseigner ce paragraphe pour chaque écran/Fonctionnalité]

2.3.1 Rôle et objet de l'écran / fonctionnalité

[Détailler le rôle de l'écran, intégrer la copie d'écran correspondante]

Cette partie est à répéter pour chaque écran de la transaction.

Si des écrans contiennent des modales, il peut être nécessaire d'ajouter des sections supplémentaires.

L'objectif est de montrer le rendu attendu.

Pour cela:

- Indiquer en quelques mots la **fonctionnalité assurée par cet écran**
- **Insérer l'impression d'écran** issue de la maquette réalisée en atelier avec les ergonomes du CEIHM.

2.3.1 Rôle et objet de l'écran / fonctionnalité

Cet écran permet la gestion des activités de juridiction avec une recherche des activités existantes et l'accès à la création, modification, suppression de ces activités

The screenshot shows a software interface titled 'Gérer Activités de juridiction'. At the top, there is a navigation bar with tabs: 'Activités de juridiction' (which is active), 'Pouvoirs', 'Applications', 'Fonctions Métier', and 'Habilitations'. On the far right of the top bar is a 'Aide' link. Below the navigation bar, the main title 'Gérer Activités de juridiction' is displayed in blue. Underneath the title, there is a section labeled 'Critères de recherche' with three input fields: 'Code fonctionnel:', 'Libellé activité:', and 'Identifiant:'. Each field has a magnifying glass icon to its right. To the right of these fields is a button bar with three buttons: '+ Créer une activité' (with a plus sign icon), 'Modifier' (with a pencil icon), and 'Lier / Délier' (with a circular arrow icon). A note below the search fields says 'Saisir un critère de recherche pour afficher la liste des activités correspondantes.'

L'objectif de ce tableau est de préciser au développeur le type de composant attendu et les formats d'affichage.

2.3.2 Description des champs de l'écran

Composant	Saisissable	Obligatoire	Type	Format	Taille	Commentaire
[Nom du champ/composant]	[Oui / Non]	[Oui / Non]	[Champ texte, Liste déroulante, ...]	[Numérique, Alphabétique,...]		

Colonne Composant : permettre de faire le lien avec les éléments de la maquette

Colonne Saisissable : les valeurs possibles sont oui, non, oui/non. Pour cette troisième valeur il faut expliciter la RG dans la colonne commentaire.

Colonne Obligatoire : les valeurs possibles sont oui, non, oui/non. Pour cette troisième valeur il faut expliciter la RG dans la colonne commentaire.

L'astérisque * signalant la caractéristique obligatoire du champ doit être visible sur la maquette en cas de valeur oui et oui/non.

Colonne Type : de nombreux types possibles :

Champ texte

Zone déroulante

Bouton radio

Case à cocher etc...

Colonne Format : en fonction du type :

Numérique

Alphabétique/ Alphanumérique

Booléen etc...

Mais aussi des formats plus spécifiques comme mail, RIB etc...

Colonne Taille : en fonction du type et du format :

Un champ texte numérique peut être : entier, décimal (5,2) etc...

Un nombre de caractère limite peut être appliqué à un champ texte.

Il faut retenir que le DCIG est un document très complet, qui permet de décrire au développeur IHM la transaction à développer. Il contient toutes les données en entrées ou sortie de services.

Il détails ce que chaque composant de l'écran permet de faire.

Sans ce document un développeur IHM ne peut pas développer.

CONCLUSION

Pour conclure ce dossier, je dirai que cette année a été pour moi très formatrice tant professionnellement qu'humainement. Il me reste encore beaucoup de chose à apprendre techniquement parlant mais cette première expérience m'a plu, même si c'était difficile.

Le fait de travailler en ESN et pour un gros client bancaire était également une première pour moi. Communication, organisation et savoir être sont pour moi les principales qualités que j'ai retenues pour faire avancer un projet correctement.

Je remercie encore une fois le groupe CGI de m'avoir permis de faire cette formation cette année.