

# BTS Services informatiques aux organisations

## *Solutions logicielles et applications métiers*

### Épreuve E5 : Éléments de questionnement

Avril 2012

*Ce document est un recueil de questions qui permettent d'évaluer des compétences relevant de l'épreuve E5. En fonction de ses objectifs pédagogiques, la professeure ou le professeur pourra en sélectionner et les agencer de façon à produire un sujet réalisable par les étudiants dans le temps de l'épreuve.*

*Le questionnaire est structuré en missions pour lesquelles les activités du référentiel (et donc les compétences), dont l'évaluation est visée, sont indiquées sous leur titre.*

#### **Liste des documents**

**Document 1A :** Schéma relationnel de la base de données.

**Document 1B :** Copie d'écran de l'enregistrement d'une commande dans l'application GestSac.

**Document 1C :** Cas d'utilisation "Enregistrement d'une commande".

**Document 1D :** Extrait des règles de développement.

**Document 1E :** Extrait de la classe Commande.

**Document 1F :** Extrait de la classe de tests unitaires de la classe Commande.

**Document 1G :** Classe HashMap du framework JAVA.

**Document 1H :** Classe Assert du framework JAVA.

**Document 1I :** Fiche d'incident n°GS53.

**Document 2A :** Cahier des charges fonctionnel de la nouvelle vitrine *web*.

**Document 2B :** Architecture de l'application vitrine.

**Document 2C :** Chargement en mémoire de l'ensemble des objets utiles.

**Document 2D :** Extrait du contrôleur et des vues.

**Document 2E :** Fichier de configuration web.xml.

**Document 2F :** Extrait de la classe Ecouteur.

**Document 4A :** Proposition commerciale de e-commerce.

**Document 4B :** Coûts du serveur et de l'abonnement Internet.

**Document 4C :** Coûts de formation et de rémunération.

**Document 4D :** Découpage temporel du projet.

**Document 4E :** Extrait du contrat de service avec le prestataire actuel.

# L'entreprise Shop'n Bag

## Contexte organisationnel

Fondée au début des années 2000, Shop'n Bag est une entreprise composée d'une vingtaine de salariés. Son siège social se situe dans la capitale. Spécialisée dans la maroquinerie grand public, Shop'n Bag s'adresse essentiellement à une clientèle plutôt féminine.

Le dirigeant de l'entreprise s'appuie sur une équipe de stylistes fidèle depuis le lancement de la société et des partenaires industriels pour assurer la production.

Pour vendre ses produits, deux canaux de commercialisation sont exploités : un site *web* marchand accessible à tous les internautes et un réseau de revendeurs implantés sur tout le territoire.

## Cartographie des applications

### Le site *web* marchand

Pour des raisons stratégiques, Shop'n Bag a préféré externaliser la gestion de son site *web*. La société prestataire SSII-PDES s'occupe entièrement de la maintenance et de la mise à jour du site marchand. Le service logistique de Shop'n Bag récupère par courriel les commandes enregistrées et payées sur le site puis expédie les produits aux clients internautes.

### L'application de vente aux revendeurs

Pour gérer son réseau de revendeurs, Shop'n Bag a développé en interne et exploite une application, nommée GestSac, permettant de gérer les produits, les revendeurs et leurs commandes et de répondre aux spécificités de la maroquinerie avec notamment la gestion des couleurs et des matières. L'application GestSac est installée sur l'ordinateur de la salle d'exposition.

De nombreux revendeurs se déplaçant chaque jour à la salle d'exposition du siège social, l'application GestSac est utilisée quotidiennement. On compte environ une centaine de commandes mensuelles. Le règlement des commandes, qui n'entre pas dans les attributions de l'application, est suivi par le comptable et enregistré à l'aide d'un progiciel de gestion comptable.

### Les autres applications

En plus de l'application GestSac, l'entreprise utilise des logiciels achetés à des éditeurs (suite bureautique, gestion comptable, gestion de base de données, gestion de la relation clients).

## Le service informatique

Le service informatique est composé de deux personnes qui assurent la gestion du parc informatique de l'entreprise aussi bien sur le plan matériel que logiciel. Notamment il assure la maintenance de l'application GestSac.

L'un des deux informaticiens est chargé du suivi du contrat liant Shop'n Bag avec le prestataire gérant le site marchand.

## **Production de nouveaux services**

Positionné sur un marché caractérisé par une vive concurrence, Shop'n Bag doit renouveler son offre à un rythme mensuel. Si l'activité du site marchand est florissante, le réseau de revendeurs est stable et l'entreprise souhaite l'étendre aux pays limitrophes.

Anticipant la montée en puissance des commandes liées à l'ouverture de ces nouveaux marchés et à l'intégration de nouveaux partenaires, elle a lancé dernièrement une réflexion visant à faire évoluer sa communication avec les revendeurs et à leur éviter de se déplacer jusqu'au siège social. L'enjeu étant de soutenir la montée en puissance de Shop'n Bag dans les pays européens, l'entreprise cherche une solution offrant les services suivants :

- Partage d'informations avec ses revendeurs (présentation des produits, suivi de commandes)
- Enregistrement de commandes en ligne
- Possibilité de prise de contact pour des revendeurs potentiels.

Jusqu'à présent, Shop'n Bag n'avait pas ressenti le besoin d'utiliser Internet dans ses relations BtoB car dans son activité les partenaires sont peu nombreux et se connaissent bien. Or, le développement important des technologies de communication a changé sa vision. Le réseau Internet lui semble être le meilleur vecteur pour promouvoir de nouveaux produits, élargir ses marchés et être présent quotidiennement auprès de ses revendeurs.

De ce fait le déploiement d'une vitrine *web* a été décidé.

Pour développer le site vitrine, la société n'envisage pas de s'appuyer sur le prestataire déjà en charge de son site marchand. En effet, même si le site marchand fonctionne bien et n'en demeure pas moins rentable, il se trouve que Shop'n Bag connaît des difficultés dans les relations qu'elle entretient avec ce prestataire (problèmes de coûts, de délais et de sécurité). De plus, pour développer une expertise en interne, l'entreprise a décidé à titre expérimental de développer elle-même cette vitrine. Pour cela, elle souhaite faire appel à l'expertise de deux développeurs d'une société d'ingénierie pendant trois mois.

L'exploitation du site vitrine servira aussi de test pour évaluer la capacité du service informatique à gérer ce type de solution. La compétence acquise pourrait alors permettre éventuellement de rapatrier en interne la gestion du site marchand et ainsi se désengager du prestataire.

*Vous faites partie du service informatique et êtes chargé(e) de la maintenance de l'application GestSac. Vous allez être associé(e) à l'étude et la mise en place du site vitrine et au rapatriement du site marchand.*

**Première mission : Maintenance de l'application GestSac**

**Deuxième mission : Étude de la solution de site vitrine**

**Troisième mission : Analyse des performances du site vitrine**

**Quatrième mission : Choix d'une solution pour le rapatriement du site marchand**

# **Première mission :**

## **Maintenance de l'application GestSac**

### ***D2.3 Gestion des problèmes et du changement***

A2.3.1 Identification, qualification et évaluation

A2.3.2 Proposition d'amélioration d'un service d'un problème

Lorsqu'un revendeur souhaite acheter des sacs, il doit se rendre au siège social à la fois pour voir les produits et pour passer commande. La commerciale ou son assistant se connecte alors à l'application GestSac, choisit dans le menu le choix "Gestion des commandes" puis "Enregistrer une commande". Le fonctionnement de l'application est décrit dans les documents 1A à 1D.

L'application GestSac donne satisfaction auprès des commerciaux. Les incidents de niveau de gravité « bloquant » et « majeur » sont corrigés au fur et à mesure mais certains incidents de niveau de gravité « mineur » sont en attente de traitement pour la prochaine version de l'application. Afin de rendre plus fiable la couche métier, il a été décidé de prendre en charge les incidents de niveau de gravité « mineur » pour lesquels la couche métier est concernée.

Comme les informaticiens en place souhaitent s'appuyer sur la couche métier de l'application existante (développée sous la plateforme Java) pour le futur site vitrine, ils entreprennent une revue de code des classes composant cette couche métier afin de la fiabiliser. Cette revue de code concerne la vérification du respect des règles de développement et la correction d'incidents mineurs. Les efforts portent actuellement sur la classe Commande dont un extrait est fourni dans le document 1E.

Vous participez à cette revue de code et disposez de la documentation technique suivante :

- Un extrait des règles de développement, document 1D,
- Un extrait de la classe Commande, document 1E,
- Un extrait de la classe de tests unitaires de la classe Commande, document 1F,
- La classe HashMap du framework JAVA, document 1G,
- La classe Assert du framework JAVA, document 1H,
- Fiche d'incident n° GS53 de niveau de gravité « mineur », document 1I.

1. 1	<b>Exposer de manière justifiée les règles de développement qui ne sont pas respectées dans l'extrait de code de la classe Commande.</b>
1. 2	<b>Procéder à l'analyse du problème exposé dans la fiche d'incident n°GS53 en rédigeant le texte renseignant la rubrique « Cause du problème » de la fiche d'incident sur votre copie.</b>

Vos collègues vous indiquent que les tests unitaires de la classe Commande n'ont pas permis de détecter plus tôt le problème relaté dans la fiche d'incident n°GS53.

1. 3	<b>Apporter les modifications nécessaires à la classe Commande et à la classe de test associée pour vous assurer que le problème ne se produise plus.</b>
---------	---

La perspective de mise en place d'un site vitrine oblige le service informatique à réfléchir d'une part à l'évolution fonctionnelle de l'application GestSac qui servira à définir le cahier des charges de la future application en ligne et d'autre part aux changements organisationnels induits par ce nouveau service.

Les besoins fonctionnels portent sur

- L'enregistrement des revendeurs : l'application doit permettre au moment de la commande la saisie d'un nouveau revendeur.
- La gestion des stocks : un commercial doit être en mesure de signaler à un revendeur que la quantité qu'il souhaite commander n'est pas en stock.

La documentation technique suivante doit vous guider pour circonscrire les besoins d'évolution :

- Le schéma relationnel de la base de données, document 1A,
- La copie d'écran de l'enregistrement d'une commande dans l'application GestSac, document 1B,
- Le cas d'utilisation "Enregistrement d'une commande", document 1C.

1. 4	Justifier en quoi l'application actuelle répond ou non aux besoins fonctionnels énoncés ci-dessus.
---------	--

Si l'application GestSac satisfait ses utilisateurs, on peut s'interroger sur son usage avec d'autres utilisateurs que les commerciaux pour lesquels elle a été conçue.

1.5	Expliquer ce qui dans l'interface de l'application actuelle pourrait être amélioré pour faciliter son usage par de nouveaux utilisateurs.
-----	---

Malgré la mise en place du site vitrine, certains revendeurs continueront à se rendre à la salle d'exposition du siège social.

1.6	Indiquer les modifications techniques, humaines et organisationnelles engendrées par l'arrivée du nouveau canal de distribution.
-----	--

## **Deuxième mission :** **Étude de la solution du site vitrine**

---

### ***D1.1 Analyse de la demande*** ***D4.1 Conception et réalisation d'une solution applicative***

*A1.1.1 Analyse du cahier des charges d'un service à produire*

*A4.1.3 Conception ou adaptation d'une base de données*

*A4.1.1 - Proposition d'une solution applicative*

*A4.1.4 Définition des caractéristiques d'une solution applicative*

*A4.1.7 Développement, utilisation ou adaptation de composants logiciels*

Vos collègues du service informatique, assistés de deux développeurs d'une société d'ingénierie, ont commencé à développer le site vitrine en Java, en utilisant le modèle MVC (Modèle Vue Contrôleur).

Afin de faciliter l'exploitation du code, vous êtes chargé(e) de documenter certains éléments qui viennent d'être développés.

Vous avez à votre disposition la documentation technique suivante :

- Cahier des charges fonctionnel de la nouvelle vitrine *web*, document 2A,
- Squelette de l'application, document 2B,
- Chargement en mémoire de l'ensemble des objets utiles, document 2C,
- Extrait du contrôleur et des vues, document 2D,
- Fichier de configuration *web.xml*, document 2E,
- Extrait de la classe *Ecouteur*, document 2F,
- La classe *HashMap*, document 1G,
- Le schéma relationnel de la base existante, document 1A.

2. 1	<b>En analysant les extraits de code fournis en document 2C, citer le ou les élément(s) manquant(s) dans les classes métiers du diagramme de classes.</b>
2. 2	<b>Documenter le code de la classe <i>Vitrine</i>.</b>

Vos collègues sollicitent votre aide pour mener à bien la mise en œuvre du site vitrine et vous demandent de modifier la structure de la base de données.

2. 3	<b>Présenter les modifications à opérer sur la structure de la base de données pour répondre au cahier des charges (authentification et suivi de l'activité du site).</b>
---------	---

Ils vous demandent ensuite de participer au développement de certaines parties du code. Vous êtes chargé(e) de coder l'affichage des produits (de la famille choisie). La vue (*choixProduit.jsp*) attend les attributs suivants :

- ✓ *laFamille* : objet de la classe *Famille*, correspondant à la famille choisie
- ✓ *lesProduits* : dictionnaire (*HashMap*) des produits de la famille choisie

La servlet "Control" reçoit toutes les demandes des clients. Son premier travail est donc de tester les paramètres reçus pour connaître la demande (faut-il afficher les familles, les produits ...). Une partie de ce contrôleur et l'affichage des différentes familles de produit sont présentés dans la documentation.

2. 4	<b>Dans le langage de votre choix ou en pseudo-code, écrire les deux méthodes <i>ajoutProduit</i> et <i>chargeProduits</i> utilisées dans la classe <i>Vitrine</i>.</b>
2.	<b>Dans le langage de votre choix ou en pseudo-code, écrire la méthode</b>

5	<b><i>affiProduits</i> de la classe Control.</b>
---	--

Les revendeurs sont très sensibles aux évolutions de la mode. Ils ne peuvent pas se permettre de prendre du retard dans leurs approvisionnements. Il est donc décidé de ne proposer que des produits en stock.

Quand un client prépare sa commande sur le site, il ajoute dans son panier des produits. Ce panier est mémorisé sous la forme d'un dictionnaire (*HashMap*) contenant comme clé la référence du produit, et comme valeur la quantité choisie.

La principale difficulté vient du fait qu'un internaute peut interrompre sa navigation à tout moment, dans ce cas il faut remettre en stock les produits dont la commande n'a pas été validée. Vous êtes chargé(e) de cette étude.

Le fonctionnement désiré est le suivant :

- ✓ quand un client met un article dans son panier, la quantité en stock est diminuée dans l'objet de la classe Produit correspondant,
- ✓ quand un client valide son panier, la base de données est mise à jour (quantité en stock des produits, ajout de la commande ...),
- ✓ si la session est annulée par le client, ou si son temps est écoulé (time out), les quantités réservées par le client sont remises dans les quantités en stock des objets de la classe Produit.

2. 6	<b>En étudiant la méthode <i>affiFamilles</i> présentée en document, citer les objets mis en session lors de la visite d'un internaute.</b>
---------	---

Le "time out" des sessions est géré au niveau du fichier de configuration de l'application (web.xml). Les lignes de ce fichier indiquent que le temps limite d'une session est de 30 minutes, et qu'un "listener", mis en oeuvre par la classe Ecouteur dont un extrait est fourni, sera averti à chaque ouverture ou fermeture de session.

2. 7	<b>Dans le langage de votre choix ou en pseudo-code, compléter la méthode <i>sessionDestroyed</i> de la classe Ecouteur.</b>
---------	--

## Troisième mission

### Analyse des performances des sites de Shop'n Bag

#### **D4.1 - Conception et réalisation d'une solution applicative**

##### *A4.1.3 Conception ou adaptation d'une base de données*

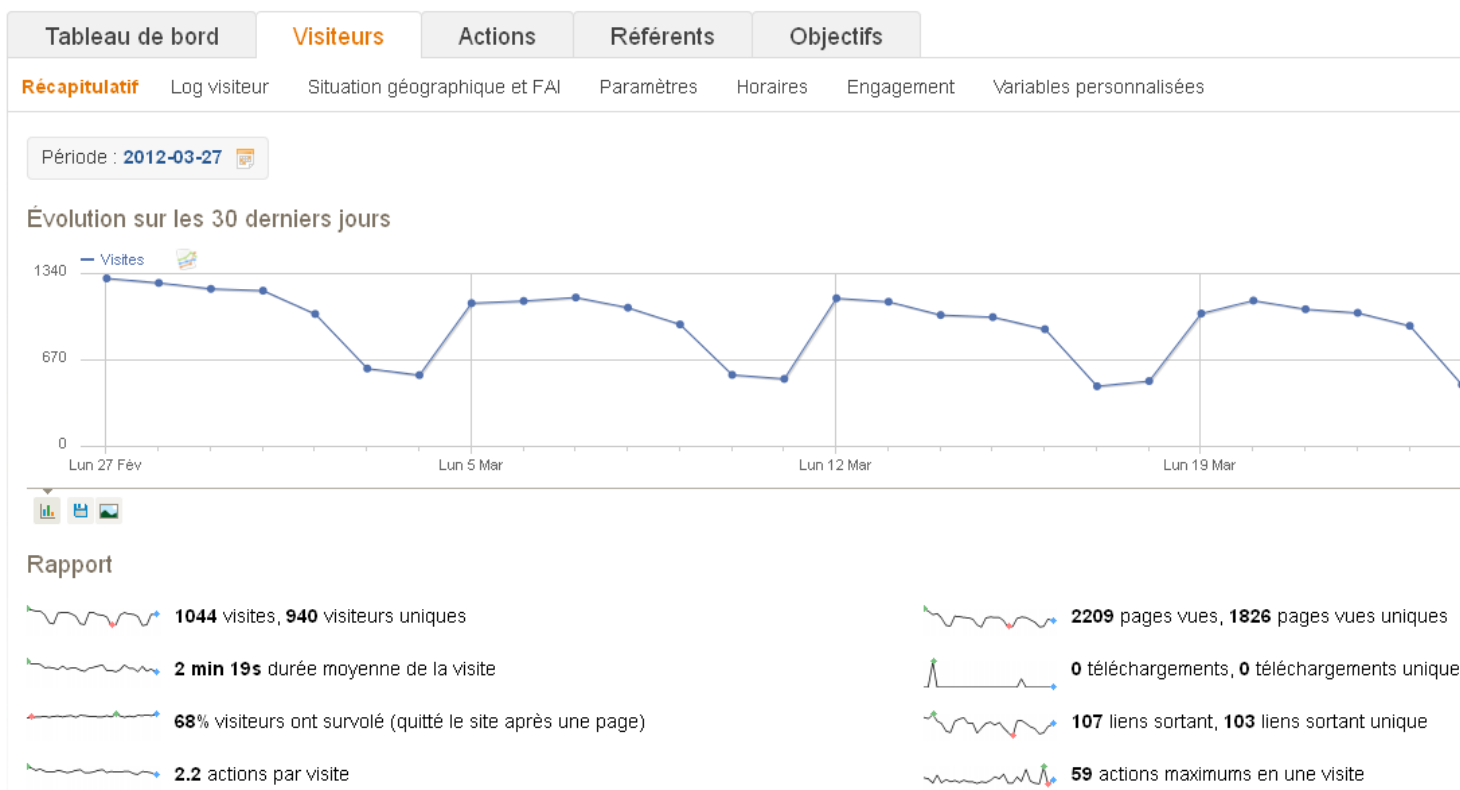
La présence sur le web de Shop'n Bag repose maintenant sur deux sites, le site marchand qui s'adresse aux internautes et qui est géré par le prestataire et le site vitrine qui s'adresse aux revendeurs.

Après 18 mois, l'expérience est techniquement considérée comme un succès. Du point de vue fonctionnel, l'outil couvre l'ensemble des besoins. Shop'n Bag peut désormais présenter ses produits en ligne et piloter la gestion des commandes avec ses revendeurs de manière efficace et fiable.

La direction décide d'enrichir l'ensemble des indicateurs mis en place afin d'évaluer le retour sur investissement réalisé dans la mise en place des deux sites. L'analyse des performances est en effet indispensable pour déterminer si les objectifs ont été atteints et le cas échéant revoir la stratégie de développement.

Shop'n bag s'appuie sur le produit *Piwik* pour établir les statistiques de fréquentation de son site dédié aux revendeurs, ce qui lui permet d'obtenir des indicateurs graphiques à partir d'un entrepôt de données alimenté régulièrement par des marqueurs placés sur les pages du site.

L'interface graphique propose de nombreux indicateurs à travers des pages *web* comme celle-ci-dessous :





L'interface graphique fournit notamment le tableau ci-après, dont les données brutes sont récupérables à partir d'une vue sur l'entrepôt de données :

Pages (date, urlPage, nbVues, nbVuesUniques, tauxRebond, tpsMoyenPage, tauxSortie)

Tableau de bord

Visiteurs

Actions

Référents

Objectifs

Pages

Pages d'entrée

Pages de sortie

Titres des pages

Liens sortants

Téléchargements

Période : 2012-03-27

Pages

URL de la page	Vues	Vues uniques	Taux de rebond	Temps moyen sur la page	Taux de sortie
<a href="#">/index</a>	139	100	15%	1 min 20s	25%
<a href="#">/list.php?2</a>	145	74	30%	1 min 16s	36%
<a href="#">/read.php?2,74422</a>	43	38	84%	38s	95%
<a href="#">/index.php</a>	47	28	67%	1 min 27s	18%
<a href="#">/list.php?5</a>	69	23	60%	2 min 36s	26%
<a href="#">/list.php?6</a>	27	23	0%	1 min 59s	22%
<a href="#">/login.php</a>	35	18	33%	54s	22%
<a href="#">/read.php?2,829</a>	18	17	88%	24s	94%
<a href="#">/read.php?2,73759,83138</a>	18	17	73%	1 min 15s	71%
<a href="#">/register.php</a>	29	16	0%	1 min 13s	19%
<a href="#">/read.php?2,72150</a>	19	15	85%	1 min 37s	93%
<a href="#">/read.php?5,56981</a>	16	15	86%	51s	93%

Le taux de rebond exprime le pourcentage de visites qui ont commencé sur la page et qui ont directement quitté le site.

Le taux de sortie exprime le pourcentage de visites qui ont quitté le site après avoir visualisé la page.

Vous contribuez à l'élaboration d'un tableau de bord destiné aux commerciaux.

3.1	<b>Proposer, dans le langage de votre choix, la requête qui permettra d'identifier les pages qui ont provoqué un abandon immédiat de la visite dans plus de 80% des cas sur le dernier mois en précisant pour chacune le nombre de visites.</b>
-----	---

Shop'n Bag souhaite d'abord mettre en place des statistiques sur son site marchand en s'appuyant sur le même produit déployé sur le site vitrine. Elle s'interroge sur les indicateurs de performance financière à retenir permettant de justifier l'investissement mis en place.

3.2	<b>Proposer deux indicateurs pertinents permettant d'évaluer la performance financière du site web marchand en justifiant leur pertinence.</b>
-----	--

Avec la mise en place de son site vitrine, Shop'n Bag avait notamment pour objectif principal d'étendre son réseau de partenaires. Elle visait une augmentation de 30% du nombre de ses revendeurs sur les deux premières années. Elle souhaite mesurer cette augmentation depuis l'ouverture en s'appuyant sur l'application de gestion des commandes GestSac que vous avez eu l'occasion d'étudier au cours de votre première mission.

La date de référencement d'un nouveau revendeur coïncide avec la date de la première commande qui le concerne.

3.3	<b>Indiquer comment il est possible d'obtenir cette information à partir de l'application de gestion des commandes en vous appuyant sur la représentation ou le langage de votre choix.</b>
-----	---

Afin d'avoir une meilleure visibilité de l'évolution de son portefeuille de revendeurs, Shop'n bag souhaite pouvoir maintenant évaluer cette augmentation chaque mois.

3.4	<b>Proposer une solution permettant de simplifier le calcul de cet indicateur pour les prochains mois.</b>
-----	--

## **Quatrième mission :** **Choix d'une solution** **pour le rapatriement du site marchand**

---

### ***D5.1 Gestion des configurations***

*A5.1.5 Évaluation d'un élément de configuration ou d'une configuration*

*A5.1.6 Évaluation d'un investissement informatique*

*A5.1.4 Étude de propositions de contrat de service (client, fournisseur)*

Il s'agit ici d'étudier l'opportunité pour Shop'n Bag de rapatrier le site *web* marchand en interne.

Avant d'envisager cette possibilité, l'équipe informatique a pris contact avec une société proposant des solutions de e-commerce. Elles sont vendues sous forme de *pack* tout en un comprenant à la fois l'hébergement et les applications en ligne de gestion d'une boutique.

#### **Solution de e-commerce clef en main**

Ces solutions sont vendues sous forme d'abonnement mensuel dont le coût est fonction des options choisies.

Après avoir débattu des différentes solutions présentées dans la proposition commerciale de e-commerce, l'équipe informatique a déterminé que seule la boutique "Excellence" proposée à 49,99 € HT/mois est à examiner.

#### **Solution du site marchand en interne**

Shop'n Bag ne souhaitant pas s'engager à long terme avec un nouveau partenaire, la société étudie donc la possibilité d'un rapatriement en interne de son site marchand.

Dans cette optique, l'équipe informatique a déterminé les besoins techniques et humains nécessaires et s'est penchée sur les conséquences du rapatriement en terme organisationnel.

Sur le plan technique, la société souhaite utiliser le serveur *web* sur lequel est stocké le site vitrine. Or il s'avère que sa capacité et sa puissance sont insuffisantes pour soutenir la charge du site marchand. L'achat d'un nouveau serveur est donc nécessaire.

Pour faire face à l'accroissement du trafic induit par la mise en ligne du site marchand, la société doit également modifier son abonnement internet pour offrir un trafic symétrique de 8 Mb/s.

Le paiement en ligne sera assuré par une passerelle de paiement auprès de la banque partenaire de Shop'n Bag pour un montant de 250 € HT/an.

Le contrat en vigueur avec le prestataire ne permet pas à Shop'n Bag de récupérer le code source du site marchand. Il faut donc le redévelopper en interne. La charge de ce développement est estimée à 6 mois homme. La nouvelle version devra être en ligne au plus tard dans 3 mois, date de renouvellement du contrat avec le prestataire SSII-PDES. Quinze jours de test sont inclus dans ce temps de développement.

Afin de renforcer l'équipe actuelle, Shop'n Bag envisage de faire appel à deux développeurs d'une société d'ingénierie engagés au forfait pour une durée de 3 mois.

De plus, suite aux problèmes récents de sécurité sur le site marchand, les informaticiens ont demandé une formation complémentaire sur la sécurité des développements et des serveurs *web*.

Sur le plan organisationnel, les responsabilités seront partagées entre les deux informaticiens de l'équipe actuelle :

- un premier informaticien sera responsable des serveurs (maintenance, sauvegarde et qualité de service),
- le second informaticien sera responsable des programmes (modification du code source et sécurisation des sites *web* vitrine et marchand)

La fonction de *webmaster* (mise à jour de nouveaux produits, mise en place de promotions ...) a été estimée à 4 jours homme par mois, elle sera partagée par l'équipe et imputée au coût du projet.

Vous avez à votre disposition la documentation technique suivante :

- Proposition commerciale de e-commerce, document 4A,
- Coûts du serveur et de l'abonnement Internet, document 4B,
- Coûts de formation et de rémunération, document 4C,
- Découpage temporel du projet, document 4D,
- Extrait du contrat de service avec le prestataire actuel, document 4E.

Vos collègues du service informatique ont retenu une configuration de serveur et vous demandent votre avis concernant ses éléments de sécurité.

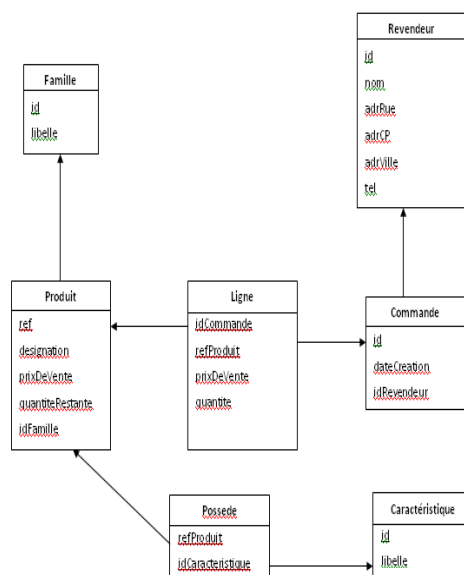
4.1	<b>Apprécier le niveau de sécurité de la configuration du serveur.</b>
-----	--

Vous êtes associé(e) à la réflexion concernant le choix d'investissement informatique. Les différentes composantes de l'investissement seront amorties sur trois ans à partir de la mise en ligne du site.

4.2	<b>Calculer le coût de l'investissement en ressources humaines (développement et formation) nécessaire au rapatriement du site marchand du mois de janvier à avril inclus.</b>
4.3	<b>Évaluer le coût annuel d'exploitation pour l'hébergement interne du site marchand à partir du moment de sa mise en ligne.</b>
4.4	<b>Argumenter sur le fait que Shop'n Bag doit ou non rapatrier son site marchand en interne. Justifier par des critères quantitatifs et qualitatifs.</b>

# Documents de la mission 1

## DOCUMENT 1A - Schéma relationnel de la base de données



L'attribut `prixDeVente` dans la relation `Ligne` permet de conserver le prix de vente au moment de la transaction

## DOCUMENT 1B - Copie d'écran de l'enregistrement d'une commande dans l'application GestSac

Application GestSac

### ENREGISTREMENT D'UNE COMMANDE

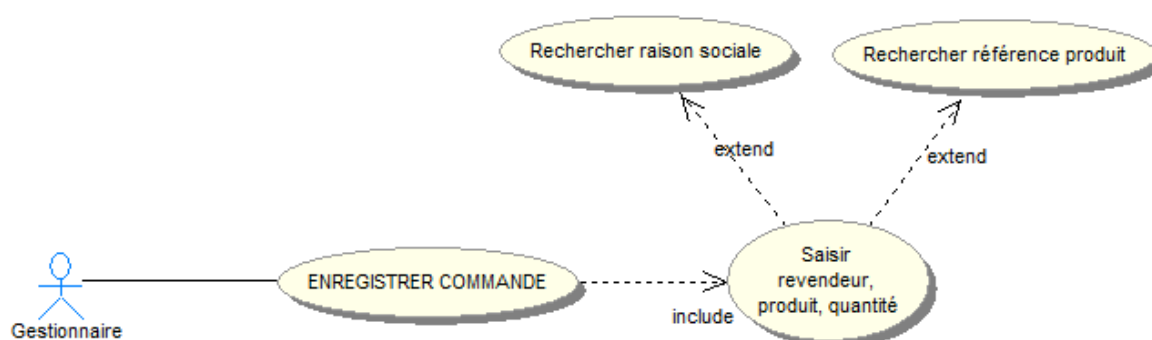
Revendeur : 
 Date :

Produit	Désignation	Quantité	Prix	Total		
AS232	Sac aspect vieilli -2 anses - 1 poche - Coloris Bleu	<input type="text" value="30"/>	<input type="text" value="8"/>	<b>240</b>	Annuler Ligne	Valider Ligne
AS233	Sac aspect vieilli -2 anses - 1 poche - Coloris Rouge	<input type="text" value="0"/>	<input type="text" value="8"/>	<b>0</b>	Annuler Ligne	Valider Ligne
GT125	Sac imitation croco - Poche avec bijou de sac - Coloris bronze	<input type="text" value="10"/>	<input type="text" value="12"/>	<b>120</b>	Annuler Ligne	Valider Ligne
HN341	Sac toucher velours - 2 anses - fermeture éclair - Coloris noir	<input type="text" value="15"/>	<input type="text" value="10"/>	<b>150</b>	Annuler Ligne	Valider Ligne

**510 €**

*F1 : Rechercher Revendeur - F2 : Rechercher Produit*

## DOCUMENT 1C - Cas d'utilisation "Enregistrement d'une commande".



### Cas d'utilisation : **Enregistrement d'une commande dans l'application GestSac**

#### Scénario nominal :

1. Le commercial saisit la raison sociale du revendeur.
2. Le système recherche le revendeur à partir de sa raison sociale et l'affiche.
3. Le commercial saisit la référence et la quantité du produit à commander.
4. Le système recherche le produit à partir de la référence saisie, affiche sa désignation et son prix puis vérifie le stock restant. S'il est inférieur à la quantité souhaitée, le système modifie la quantité saisie par la quantité restante. Pour terminer, le système valide la ligne de commande, met à jour les totaux de la ligne et de la commande, et place le curseur à la ligne suivante pour une nouvelle saisie de produit à commander.
5. --- Le commercial peut recommencer au point 4 pour continuer la saisie d'autres produits ---
6. Le commercial valide la commande.
7. Le système enregistre les lignes de commandes, la commande, met à jour les stocks de produit selon les quantités commandées et informe l'utilisateur que la commande a bien été enregistrée.

#### Extensions :

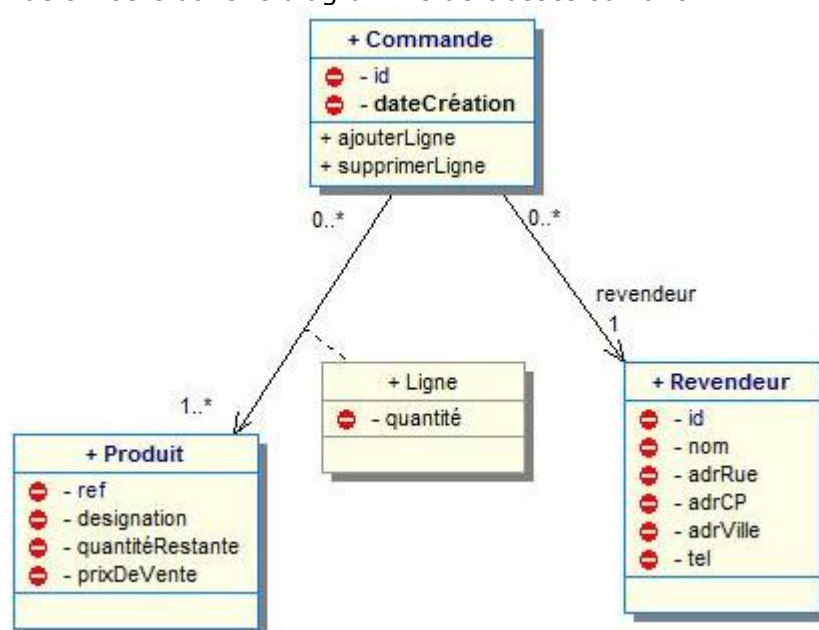
- 2. . Le système ne trouve pas le revendeur :
  - Le système informe l'utilisateur qu'il n'a pas trouvé le revendeur.
  - Retour à l'étape 1.
- 4. Le système ne trouve pas le produit :
  - Le système informe l'utilisateur qu'il n'a pas trouvé le produit.
  - Retour à l'étape 3.

## DOCUMENT 1D – Extrait des règles de développement

1. Toute méthode publique d'une classe sera précédée d'une documentation qui comprendra au minimum le résumé, la description des paramètres et du résultat suivant le format JavaDoc.
2. Le nom (identificateur) d'une classe respectera la notation Pascal : la première lettre du nom de méthode et la première lettre de chaque mot présent dans l'identificateur sont en majuscules. Par exemple, LigneCommande respecte la notation Pascal.
3. Le nom d'une méthode est un verbe, ou un groupe verbal. Les méthodes qui permettent de lire (resp. écrire) directement une variable privée d'instance sont préfixées par get (resp. set), suivi du nom de la variable.
4. Le nom (identificateur) des méthodes, paramètres formels et des variables locales respectera la notation Camel : la première lettre du nom est en minuscules et la première lettre de chaque mot présent dans l'identificateur est en majuscules. Par exemple, uneQte respecte la notation Camel.
5. Le nom des méthodes, paramètres et variables doit être le plus explicite possible et informer de leur rôle. Il faut privilégier la lisibilité à la concision.
6. Le nom des méthodes, paramètres et variables ne contient que des lettres non accentuées ou des chiffres : le tiret bas, trait d'union ou tout autre caractère non alphanumérique sont interdits.

## DOCUMENT 1E – Extrait de code de la classe Commande

La classe Commande s'insère dans le diagramme de classes suivant :



Classe Produit – Méthodes publiques (extrait)

```
public class Produit
{
    public Produit(String uneRef, String uneDesignation, int uneQteRestante,
        double unPUHT) // constructeur
    {
    }
    public int getQuantiteRestante() // accesseurs sur la quantité restante
    {
    }
    public void setQuantiteRestante(int value)
    {
    }
}
```

```

public class Commande {
    int            Id;
    Date            DateCreation;
    Revendeur       revendeur;
    HashMap<Produit, Integer> lignes;

    /**
     * Crée une instance de classe Commande.
     * @param i      l'id de commande
     * @param d      la date de commande
     * @param r      le revendeur
     */
    public Commande(int i, Date d, Revendeur r) {
        Id            = i;
        DateCreation  = d;
        lignes        = new HashMap<Produit, Integer>();
        revendeur     = r;
    }

    /**
     * Fournit la liste des lignes de la commande sous forme de dictionnaire
     * en clé, l'instance de produit, en valeur, la quantité commandée
     * @return liste des lignes de commande
     */
    public HashMap<Produit, Integer> getLignes() {
        return lignes;
    }

    /**
     * Ajoute une ligne de commande à l'instance de commande courante
     * La quantité de la ligne est ajustée à la qté du produit restant en stock.
     * Si le produit figure déjà dans une ligne de commande, la quantité demandée
     * remplace la quantité actuelle.
     * @param unProd : instance de classe Produit
     * @param uneQte : nombre de produits commandés
     */
    public void ajouterLigne(Produit unProd, int uneQte){
        int qteRestante = unProd.getQuantiteRestante();
        if ( uneQte > qteRestante ) {
            uneQte = qteRestante;
        }
        if ( uneQte >= 0 ) {
            lignes.put(unProd, uneQte);
        }
    }

    public void supprimer_ligne(Produit unProd){
        if (lignes.containsKey(unProd)) {
            lignes.remove(unProd);
        }
    }
}

```



## DOCUMENT 1F – Extrait de la classe de tests unitaires de la classe Commande

```
/**
 * Classe de test de la classe Commande
 * Hérite de la classe TestCase issue du framework de tests unitaires Junit
 * Permettra d'enchaîner l'exécution de toutes les méthodes commençant par test
 * L'exécution s'arrête à la première affirmation non respectée, avec le
 * message d'erreur associé qui s'affiche. Rien n'est affiché concernant les
 * affirmations respectées, l'exécution se termine avec le message Test réussi
 * si toutes les affirmations ont été respectées.
 */
public class TestCommande extends TestCase {
    /**
     * Méthode de test pour ajouterLigne
     * Cas de test sur un produit n'ayant pas encore fait l'objet d'une ligne de commande
     */
    public void testAjouterLigneNouveauProduit() {
        // création d'une commande et d'un produit
        Commande commande = new Commande(1, new Date(), new Revendeur(100,"Au joli sac"));
        Produit produit = new Produit("AS232", "Sac aspect vieilli - 2 anses", 10, 8);

        // cas de test d'une ligne de commande d'un produit non encore commandé
        commande.ajouterLigne(produit, 2);
        HashMap<Produit, Integer> lesLignes = commande.getLignes();
        Assert.assertEquals("Une ligne supplémentaire", lesLignes.size(), 1);
        Assert.assertNotNull("Ligne ajoutée accessible", lesLignes.get(produit));
        int qte = lesLignes.get(produit);
        Assert.assertEquals("Quantité de la ligne égale à celle demandée", qte, 2);

        // cas de test d'une nouvelle ligne avec une quantité en stock insuffisante
        Produit autreProduit = new Produit("HN341", "Sac toucher velours - 2 anses", 5, 10);

        commande.ajouterLigne(autreProduit, 10);
        lesLignes = commande.getLignes();
        Assert.assertEquals("Une ligne supplémentaire", lesLignes.size(), 2);
        Assert.assertNotNull("Ligne ajoutée accessible", lesLignes.get(autreProduit));
        qte = lesLignes.get(autreProduit);
        Assert.assertEquals("Quantité égale à celle en stock", qte, 5);
    }
    /**
     * Méthode de test pour ajouterLigne
     * Cas de test d'un produit déjà commandé dans la commande
     */
    public void testAjouterLigneProduitDejaCommande() {
        // création d'une commande, d'un produit et d'une première ligne de commande
        Commande commande = new Commande(1, new Date(), new Revendeur(100,"Au joli sac"));
        Produit produit = new Produit("AS232", "Sac aspect vieilli - 2 anses", 10, 8);

        commande.ajouterLigne(produit, 2);

        // cas de test d'une ligne existante avec une quantité différente
        commande.ajouterLigne(produit, 5);
        HashMap<Produit, Integer> lesLignes = commande.getLignes();
        Assert.assertEquals("Pas de ligne supplémentaire", lesLignes.size(), 1);
        Assert.assertNotNull("Ligne modifiée accessible", lesLignes.get(produit));
        int qte = lesLignes.get(produit);
        Assert.assertEquals("Quantité égale à celle demandée", qte, 5);

        // cas de test d'une ligne existante avec une quantité en stock insuffisante
        commande.ajouterLigne(produit, 15);
        lesLignes = commande.getLignes();
        Assert.assertEquals("Pas de ligne supplémentaire", lesLignes.size(), 1);
        Assert.assertNotNull("Ligne modifiée accessible", lesLignes.get(produit));
        qte = lesLignes.get(produit);
        Assert.assertEquals("Quantité égale à celle en stock", qte, 10);
    }
}
```

## DOCUMENT 1G – Présentation de la classe HashMap du framework Java

*/\*\**

*La classe HashMap permet de mémoriser un dictionnaire d'éléments (clé, valeur). Toute valeur (de type ValueType) peut être extraite à partir de sa clé (de type KeyType) à une clé présente dans le dictionnaire correspond une et une seule valeur*

*\*/*

**Class HashMap <KeyType, ValueType>**

```
public void put (KeyType key, ValueType value)
    // ajoute un élément (key, value). Remplace la valeur existante si
    // l'élément existe déjà pour la clé spécifiée.

public ValueType get (KeyType key)
    // retourne la valeur correspondant à la clé spécifiée, ou null si la clé
    // spécifiée est inexistante.

public ValueType remove (KeyType key)
    // retire du dictionnaire l'élément correspondant à la clé spécifiée et
    // retourne la valeur qui y était associée. Ne fait rien et retourne null
    // si la clé était inexistante.

public Boolean containsKey (KeyType key)
    // retourne true si l'élément dont la clé est passée en paramètre est
    // présent dans le dictionnaire, false sinon

public Set<KeyType> keySet ()
    // retourne un ensemble des clés présentes dans le dictionnaire d'éléments
```

## DOCUMENT 1H– Présentation de la classe Assert du framework Java

*/\*\**

*La classe Assert réunit un ensemble de méthodes statiques d'assertion utiles pour l'écriture des tests. Une méthode d'assertion a pour rôle de vérifier si une affirmation est ou non respectée. Dans le cas où elle ne l'est pas, la méthode déclenche une exception. Si non, elle ne fait rien.*

*\*/*

**Class Assert**

```
public static void AssertEquals(String message, int expected, int actual)
    // vérifie si les valeurs entières expected et actual sont égales.
    // Si elles ne le sont pas, déclenche une exception de type
    // AssertException comportant le message d'erreur spécifié.

public static void AssertEquals(String message, Object expected, Object actual)
    // vérifie si les deux objects expected et actual sont égaux.
    // S'ils ne le sont pas, déclenche une exception de type
    // AssertException comportant le message d'erreur spécifié.

public static void AssertNull(String message, Object obj)
    // vérifie si l'objet obj est une référence nulle. S'il ne l'est pas,
    // déclenche une exception de type AssertException comportant le message
    // d'erreur spécifié.

public static void AssertNotNull(String message, Object obj)
    // vérifie si l'objet obj n'est pas une référence nulle. S'il l'est,
    // déclenche une exception de type AssertException comportant le message
    // d'erreur spécifié.
```

## Document 1I - Fiche d'incident n° GS53

### Description incident

Date signalement :  
25/02/2012

Rédacteur :  
Assistant commercial  
Arnaud Belloir

Niveau de gravité :  
☐Bloquant ☐Majeur ☐Mineur

Environnement concerné

Matériel :  
Poste Caisse n°2

Logiciel:  
GestSac

Description du problème avec éventuelles captures d'écran, messages d'erreurs :

J'ai constaté que certaines lignes présentaient une quantité nulle dans les commandes comme dans l'exemple ci-dessous.

N° commande	Référence produit	Quantité commandée	Prix de vente
345126	AS232	30	8
345126	AS233	0	8
345126	GT125	10	12

### Cause du problème

Date analyse :

Rédacteur :

Description de la cause :

### Actions réalisées

Date réalisation :

Rédacteur :

Version de correction :

# Documents de la mission 2

## DOCUMENT 2A – Cahier des charges fonctionnel de la nouvelle vitrine web

### CAHIER DES CHARGES FONCTIONNEL VITRINE WEB

- Présentation générale du problème

1. Projet

1. Finalités

L'objectif est de mettre en ligne une application « vitrine » pour nos clients professionnels, les revendeurs. Ils pourront voir nos produits et passer commande.

2. Espérance de retour sur investissement

En servant mieux nos partenaires actuels, nous visons une augmentation substantielle de nos commandes. Nous visons aussi à étendre notre clientèle de boutiques de province (leur déplacement vers notre « show-room » parisien est actuellement un handicap).

2. Contexte

1. Situation du projet par rapport aux autres projets de l'entreprise

Ce projet, important dans ses objectifs par lui-même, est stratégique pour notre entreprise. C'est notre premier développement web interne. Sa réussite conditionne notre poursuite dans ce domaine, peut être jusqu'au rapatriement de notre site grand-public.

2. Études déjà effectuées

Le service informatique a réalisé une refonte du code existant de l'ancienne version de notre logiciel de prises de commandes. Dans le but de minimiser les coûts et les délais de développement, les informaticiens en place décident de conserver une partie du système d'information existant : le site web s'appuiera sur la base de données existante.

3. Caractère confidentiel s'il y a lieu

Notre prestataire SSII-PDES ne doit pas être informé de ce projet : de difficiles négociations sont en cours concernant notre site grand-public ...

- Expression fonctionnelle du besoin

- 

1. Fonctions de service principales

1. Authentification

Nos clients étant des professionnels, ils sont déjà enregistrés dans notre base de données, ou pourront s'inscrire auprès de notre service commercial. Il n'y a pas lieu de prévoir un formulaire d'inscription. Par contre le service commercial fournira à nos partenaires un « login » et un mot de passe leurs permettant de s'authentifier sur le site. Il faut prévoir les ajouts à la base concernant ces informations.

2. La vitrine

La présentation de nos produits se fera de façon hiérarchique : une première page (choixFamille) présentera les familles de produits que nous proposons, et après ce choix une page (choixProduit) permettra de mettre les produits choisis dans le « panier » du client.

3. Les sessions

Les clients pourront naviguer librement sur le site en complétant leur panier. Il est donc indispensable de gérer des sessions.

2. Fonctions de service complémentaires

Pour permettre une analyse précise de l'activité du site, il faut prévoir de stocker dans la base de données l'ensemble des connexions des clients : on enregistrera la date et l'heure de chaque connexion.

3. Contraintes

La base de données existante devra être utilisée. Les tables existantes, encore utilisées par les applications locales ne devront pas être modifiées. Seul l'ajout de nouvelles tables est possible.

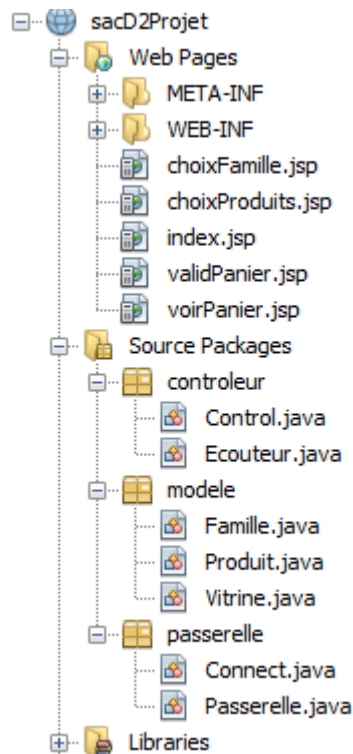
L'application existante devra autant que possible être réutilisée, les deux grandes nouveautés seront la présentation de la vitrine, et la gestion du « panier » du client.

## DOCUMENT 2B – Architecture de l'application vitrine

Le modèle MVC choisi est basé sur une Servlet contrôleur qui reçoit toutes les demandes des clients (get ou post). Suivant la demande, le contrôleur recherche les données utiles dans le modèle (ensemble des classes métier). Il met en forme ces données pour les envoyer aux vues qui seront affichées sur le navigateur du client.

Les vues seront des pages JSP, constituées uniquement de HTML et de "tag" JSTL (balises prévues pour traiter les données reçues depuis la Servlet contrôleur).

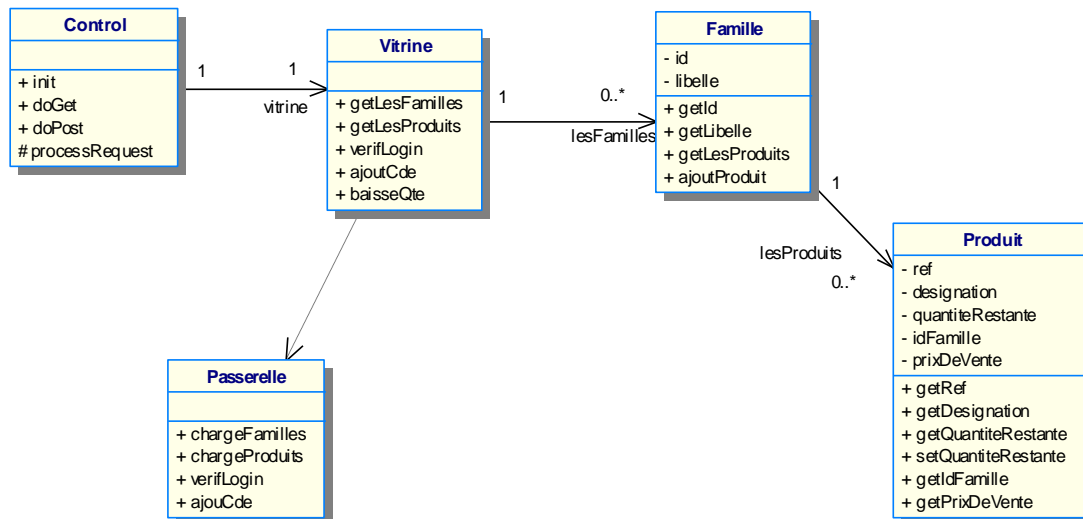
Squelette de l'application :



Au démarrage de l'application la méthode "init" du contrôleur, automatiquement appelée, a pour objectif de charger en mémoire l'ensemble des objets utiles.

## Extrait du diagramme de classes

La classe *Passerelle* contient toutes les méthodes à portée de classe (statiques) qui accèdent à la base de données. Les constructeurs et la classe *HashMap* ne sont pas représentés ici. Les méthodes privées des classes ne sont pas représentées.



## Extraits de code de l'application :

//classe Control (extrait)

```

public class Control extends HttpServlet {
    private Vitrine vitrine;
    /**
     * Méthode appelée au chargement de l'application
     * charge en mémoire les objets utiles
     * @param config la configuration de la servlet
     */
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        vitrine = new Vitrine();
    }
}
  
```

//classe Vitrine (extrait)

```

public class Vitrine {
    // dictionnaire des familles, en clé l'id et en valeur l'objet de la classe Famille
    private HashMap <Integer, Famille> lesFamilles;
    // dictionnaire des produits, en clé la référence et en valeur l'objet de la classe Produit
    private HashMap <String, Produit> lesProduits;
    /**
     * Constructeur charge en mémoire les objets utiles
     */
    public Vitrine() {
        lesFamilles = Passerelle.chargeFamilles();
        lesProduits = Passerelle.chargeProduits();
        for (String refProd : lesProduits.keySet()) {
            Produit unProduit = lesProduits.get(refProd);
            int idFamille = unProduit.getIdFamille();
            lesFamilles.get(idFamille).ajoutProduit(unProduit);
        }
    }
}
  
```

*//classe Passerelle (extrait)*

```
public class Passerelle {  
    /**  
     * Charge les familles depuis la base de données  
     * @return dictionnaire des familles (id, Famille)  
     */  
  
    public static HashMap<Integer, Famille> chargeFamilles() {  
        HashMap<Integer, Famille> lesFamilles = new HashMap<Integer, Famille>();  
        Connection con = Connect.get();  
        Statement req;  
        req = con.createStatement();  
        ResultSet rs = req.executeQuery( "select * from famille");  
        while ( rs.next() ) {  
            int idFamille = rs.getInt("id");  
            String libelleFamille = rs.getString("libelle");  
            Famille uneFamille = new Famille(idFamille, libelleFamille);  
            lesFamilles.put(idFamille, uneFamille);  
        }  
        rs.close();  
        return lesFamilles;  
    }  
}
```

## DOCUMENT 2D – Extrait du contrôleur et des vues

Ci dessous un extrait du code de la méthode du contrôleur (classe Control) qui reçoit les demandes des internautes (une demande est gérée grâce au paramètre reçu request). Pour récupérer les paramètres envoyés par une méthode GET ou POST, on utilise la méthode request.getParameter.

**protected void processRequest(HttpServletRequest request, HttpServletResponse response)** throws ServletException, IOException {

```
    String page;  
    //récupération du paramètre action  
    String action = request.getParameter("action");  
  
    if (action == null) {  
        //début, affichage page de connexion  
        page = "index.jsp";  
    } else {  
  
        // test du paramètre action :  
        // 1 : retour du login  
        // 2 : retour du choix de la famille de produit  
        // 3 : retour du choix des produits  
        // 4 : panier validé  
  
        int valAction = Integer.parseInt(action);  
  
        switch (valAction) {  
  
            case 1:  
                String login = request.getParameter("login");  
                String mdp = request.getParameter("mdp");  
                int numCli = vitrine.verifLogin(login, mdp);  
                if (numCli == 0) {  
                    request.setAttribute("erreur", "Login incorrect");  
                    page = "index.jsp";  
                }  
            }  
        }  
    }  
}
```

```

        } else {
            page = affiFamilles(request, numCli); // Méthode privée qui récupère les familles
        }
        break;

    case 2:
        page = affiProduits(request); // Méthode privée qui récupère les produits d'une famille
        break;
        // partie coupée
    }
}
// Appel de la vue à afficher
request.getRequestDispatcher(page).forward(request, response);
}

```

**et** la méthode `affiFamilles` de la classe `Control` :

```

private String affiFamilles(HttpServletRequest request, int numCli) {
    String page;
    // Début session
    HttpSession session = request.getSession();
    session.setAttribute("idCli", numCli);
    session.setAttribute("vitrine", vitrine);
    session.setAttribute("panier", new HashMap<String, Integer>());
    // Préparation de l'affichage des familles
    request.setAttribute("lesFamilles", vitrine.getLesFamilles());
    page = "choixFamille.jsp";
    return page;
}

```

**Et enfin la vue `choixFamille.jsp` (simplifiée) :**

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Choix famille</title>
  </head>
  <body>
    <h1>Sac</h1>
    <!-- Parcourir tous les éléments d'une collection (attribut du request) -->
    <c:forEach var="ligne" items="{lesFamilles}" >
      <a href="Control?idFamille={ligne.key}&action=2"> {ligne.value.libelle} </a>
      <br/>
    </c:forEach>

  </body>
</html>

```

Notes : On voit ici que la vue a récupéré l'attribut de l'objet `request` nommé `"lesFamilles"`, et grâce au tag JSTL `<c:forEach ... >` a pu afficher les différentes familles de produit. Le paramètre `action` est valorisé à 2 pour prévenir le contrôleur que la famille a été sélectionnée.



## DOCUMENT 2E – Fichier de configuration web.xml

```
<session-config>
    <session-timeout>30</session-timeout>
</session-config>
<listener>
    <listener-class>controleur.Ecouteur</listener-class>
</listener>
```

## DOCUMENT 2F – Extrait de la classe Ecouteur

```
public class Ecouteur implements HttpSessionListener {
    /**
     * Méthode appelée à la fin de la session du client
     * @param se évènement lié à la session en cours de fermeture
     */

    public void sessionDestroyed(HttpSessionEvent se) {
        HttpSession session = se.getSession();
        //récupération du panier
        HashMap<String, Integer> panier = session.getAttribute("panier");
        //récupération de la vitrine
        Vitrine v = session.getAttribute("vitrine");
        //à compléter
    }
}
```

# Documents de la mission 4

## DOCUMENT 4A – Proposition commerciale de e-commerce<sup>1</sup>

**e-Commerce**

### DÉVELOPPEZ VOS VENTES AVEC VOTRE E-BOUTIQUE !

Vos premiers pas dans le monde de l'e-commerce	Idéal pour développer votre activité sur Internet	Solution complète incluant toutes les fonctions
<b>1&amp;1 e-Boutique Classique</b>	<b>1&amp;1 e-Boutique Avancée</b>	<b>1&amp;1 e-Boutique Excellence</b>
<b>6 MOIS GRATUITS*</b>	<b>3 MOIS GRATUITS*</b>	<b>3 MOIS GRATUITS*</b>
puis 19,99 € HT/mois (23,91 € TTC/mois)	puis 29,99 € HT/mois (35,87 € TTC/mois)	puis 49,99 € HT/mois (59,79 € TTC/mois)
<a href="#">&gt; Commander</a>	<a href="#">&gt; Commander</a>	<a href="#">&gt; Commander</a>

\* **Offre « 6 mois à 0 € »** soumise à un engagement de 12 mois.  
A l'issue des 6 premiers mois, l'e-Boutique **Classique** est à son prix habituel de 19,99 € HT/mois (23,91 € TTC/mois). Frais de mise en service de 9,99 € HT (11,95 € TTC).

\* **Offre « 3 mois à 0 € »** soumise à un engagement de 12 mois.  
A l'issue des 3 premiers mois, les e-boutiques sont à leurs prix habituels : **Avancée** à 29,99 € HT/mois (35,87 € TTC/mois) ; **Excellence** à 49,99 € HT/mois (59,79 € TTC/mois). Frais de mise en service de 9,99 € HT (11,95 € TTC).

### Création et configuration

Définissez le nombre d'articles, de catégories, d'outil de création par étapes de l'e-boutique, d'une interface d'administration, Assistant de paramétrage Référencement sur des moteurs de produits connus

### Espace Web

Bénéficiez d'un espace d'hébergement pour votre boutique de 2 à 20 Go selon l'abonnement, gérez autant de noms de domaine que vous souhaitez (fr, .eu, .be, .com), trafic mensuel illimité, Comptes e-mail (POP/IMAP)

### Méthodes de paiement

Permettez à vos clients de payer en ligne (PayPal, Paybox) ou hors ligne (chèques, espèces)

### Sécurité et statistiques

Cryptage SSL, Statistiques (Commandes, visiteurs)

### Gestion de l'e-Boutique

Accès administrateur, Utilisateurs supplémentaires, Gestion des images (GIF, JPG, PNG, BMP, WBMP), Calcul de la TVA, Configuration des prix de vente,

### Gestion des commandes,

Bénéficiez de nombreux outils pour vous faciliter la gestion des commandes (attribution automatique de numéros de clients, édition de factures, suivi du statut des commandes...), création de dossiers de commandes supplémentaires, Annulation de commande, Exportation des données de commandes (CSV et XML), Affichage du dossier de commandes envoyées.

### Gestion des clients,

Espace client, Restriction d'accès, Assistance par mail et par téléphone, vidéos de démo, Foire aux questions

### Gestion des articles

Produits téléchargeables (Proposez des articles numériques en téléchargement tels que ebooks, photos, musique, etc.), Quantité minimale de la commande, Gestion des stocks

<sup>1</sup> Source : <http://commander.1and1.fr>



Dell PowerEdge™ T710  
Tower Chassis, Up to 8x 3.5" HDDs (6Gbps)

18 000 € HT

## 2 Processeurs

Intel Xeon E5645, 6C, 2.40GHz, 12M Cache, 5.86GT/s, 80W TDP, Turbo, HT, DDR3-1333MHz

### Mémoire:

32GB Memory for 2 CPUs, DDR3, 1066MHz (8x4GB Dual Ranked LV RDIMMs), Using 1333MHz DIMMs

### 1ère carte contrôleur RAID ou SCSI:

PERC H700 Integrated RAID 5 Controller, 512MB Cache

### 3 disques durs

300GB, SAS 6Gbps, 3.5-in, 15K RPM Hard Drive (Hot Plug)

### Alimentation:

Power Supply, Redundant (2 PSU), 1100W

### Onduleurs:

Dell 3U Rack/Tower UPS + NMC, 2700W, 230V, incl. Cable Pack ~17 min Runtime

### Unités de sauvegarde sur bande:

PowerVault Half Height 1500GB LTO-5 Internal Tape Backup with PERC H200A Controller

### Services de support technique:

3 ans de garantie de base - Intervention le jour ouvrable suivant incluse - Aucune extension de garantie sélectionnée

## SDSL Entreprise

*Une connexion symétrique et 100% garantie pour vos applications critiques*

Nos liaisons SDSL permettent de garantir la bande passante disponible et de proposer des débits symétriques en assurant un taux de disponibilité accru. Elles assurent le cadre nécessaire au bon fonctionnement des applications qui exigent des temps de réponses réduits et constants.

512K	<b>SDSL Entreprise</b>	à partir de 79 € HT/mois
1M	<b>SDSL Entreprise</b>	à partir de 112 € HT/mois
2M	<b>SDSL Entreprise</b>	à partir de 119 € HT/mois
3M	<b>SDSL Entreprise</b>	à partir de 159 € HT/mois
4M	<b>SDSL Entreprise</b>	à partir de 219 € HT/mois
5M	<b>SDSL Entreprise</b>	à partir de 239 € HT/mois
6M	<b>SDSL Entreprise</b>	à partir de 329 € HT/mois
8M	<b>SDSL Entreprise</b>	à partir de 419 € HT/mois
16M	<b>SDSL Entreprise</b>	à partir de 699 € HT/mois

## DOCUMENT 4C- Coûts de formation et de rémunération

### Formations :

Désignation	Durée	Tarif
Sécurité des réseaux TCP/IP	5 jours	3400 € HT
Sécurité du développement web	4 jours	2700 € HT

Coût journalier d'un développeur au forfait : 500 € HT

### Rémunération des salariés

Intitulé poste	Coût toutes charges comprises pour l'entreprise
Informaticiens Shop'n Bag	48 K €/an

## DOCUMENT 4D – Découpage temporel du projet

On compte vingt jours ouvrés par mois.

Janvier	Février	Mars	Avril	Mai
Formation	Développement	Développement	Développement & Tests	Mise en ligne du site

## DOCUMENT 4E– Extrait du contrat de service avec le prestataire actuel

### Article 1 – Durée

Le présent contrat entre en vigueur à compter de la date de la signature des deux parties pour une durée de 12 mois, reconductible tacitement pour une période de 1 an. Chacune des parties peut y mettre fin à tout moment, sous réserve d'en aviser l'autre partie avant l'échéance, par lettre recommandée avec accusé de réception.

### Article 2 – Prestations

Les prestations comprises dans le contrat sont les suivantes :

Hébergement :

- L'hébergement sur serveur dédié
- La passerelle de paiement en ligne
- Le référencement

Opérations marketing

- Le webdesign du site e-commerce
- La mise en ligne du catalogue produits et de la base clients
- La transmission quotidienne des commandes

Maintenance technique

- La sauvegarde
- La maintenance corrective

Toute maintenance évolutive sera facturée en sus. L'étude des évolutions demandées est facturée forfaitairement 300 € HT, et donnera lieu à un devis.

### Article 3 – Qualité de service

Le service e-commerce sera assuré à un taux de disponibilité (SLA) de 99%. La garantie du temps d'intervention (GTI) est de 4 heures en semaine ouvrée. Le débit garanti pour l'accès au site d'e-commerce est de 8Mbits/s.

### Article 4 – Tarification

L'ensemble des prestations est fourni pour 400 € HT / mois.