

PPE 2017-18

Lycée Gaston Berger

1) Organisation du PPE

a) Présentation générale

La durée hebdomadaire du PPE (Projet Personnel Encadré) est de quatre heures. Vous disposez approximativement de 20 semaines pour réaliser vos projets.

Cet enseignement doit vous permettre de mettre en œuvre les notions vues en cours et également de faire des recherches pour en découvrir de nouvelles.

Pour cela, il vous est proposé un contexte professionnel dans lequel vous serez acteur pour réaliser différentes situations professionnelles. On vous demandera donc en équipe (3 membres par groupe) d'analyser un cahier des charges d'un nouveau service en tenant compte des exigences de qualité. Vous aurez donc à élaborer un dossier de choix de solutions techniques, de rédiger les spécifications techniques de la solution retenue, de réaliser cette solution et de définir les tests et les niveaux d'habilitation associés au service.

Parallèlement, ce module invite également à contribuer au processus « P5 - Gestion du patrimoine informatique » à travers la mise en place d'un outil de gestion des configurations, l'évaluation de l'investissement nécessaire à la mise en place du service, la mise en place et l'exploitation d'un dispositif de veille technologique, ainsi que l'étude d'une technologie, d'un outil ou d'une méthode afin de proposer une solution actualisée.

b) Lien avec l'examen

Epreuve E5 - Production et fourniture de services informatiques (coeff 5)

Tout d'abord, le sujet proposé est en lien direct avec le référentiel du BTS. Travailler en PPE vous permet donc de conforter les notions vues en cours et donc de préparer au mieux l'étude de cas.

Epreuve E6 – Parcours de professionnalisation (coeff 3)

Par ailleurs, le module PPE vous permet également d'alimenter votre portefeuille de compétences. Vous devez donc identifier les compétences mises en œuvre et analyser votre pratique afin de pouvoir correctement illustrer votre pratique. Attention, vous devez effectuer régulièrement cette démarche car vous serez beaucoup plus efficace que si vous l'effectuez de manière irrégulière et très ponctuelle.

Epreuve E4 – Conception et maintenance de solutions informatiques (coeff 4)

Cette épreuve repose directement sur les travaux effectués pendant le module PPE. En effet, cette épreuve consiste à modifier une situation professionnelle réalisée durant le module PPE. Il convient donc que les spécifications techniques soient à jour et que les solutions techniques soient opérationnelles.

c) Travail en groupe

Chaque groupe est formé de trois membres dont un chef de projet. La formation des groupes est irréversible.

Vous devez impérativement tenir un planning de projet en utilisant un logiciel. Vous devrez donc établir un planning prévisionnel et faire le point sur votre avancement régulièrement.

Vous devez donc mettre en œuvre les compétences suivantes :

C1.4.1.1 Établir son planning personnel en fonction des exigences et du déroulement du projet

C1.4.1.2 Rendre compte de son activité

C1.4.2.1 Suivre l'exécution du projet

C1.4.2.2 Analyser les écarts entre temps prévu et temps consommé

C1.4.2.3 Contribuer à l'évaluation du projet

C1.4.3.1 Recenser les ressources humaines, matérielles, logicielles et budgétaires nécessaires à l'exécution du projet et de ses tâches personnelles

C1.4.3.2 Adapter son planning personnel en fonction des ressources disponibles

Par ailleurs, pour chaque technologie utilisée, chaque membre du groupe doit réaliser une tâche. En effet, l'épreuve E4 peut consister à effectuer des modifications sur des travaux effectués par un autre membre du groupe. Les technologies utilisées doivent donc être maîtrisées par l'ensemble du groupe.

d) Veille technologique et étude d'une technologie, d'un composant d'un outil ou d'une méthode

Le cadre technologique du PPE n'étant pas imposé vous allez devoir effectuer des recherches afin de prendre les bonnes décisions. Par ailleurs, les autres membres du groupe devront pouvoir bénéficier de vos recherches. Il conviendra donc de les formaliser et de les diffuser.

Chaque membre doit donc mettre en œuvre les compétences suivantes :

Veille technologique

C5.2.2.1 Définir une stratégie de recherche d'informations

C5.2.2.2 Tenir à jour une liste de sources d'information

C5.2.2.3 Évaluer la qualité d'une source d'information en fonction d'un besoin

C5.2.2.4 Synthétiser et diffuser les résultats d'une veille

Etude d'une technologie, d'un composant d'un outil ou d'une méthode

C5.2.4.1 Se documenter à propos d'une technologie, d'un composant, d'un outil ou d'une méthode

C5.2.4.2 Identifier le potentiel et les limites d'une technologie, d'un composant, d'un outil ou d'une méthode par rapport à un service à produire

e) Exemples de compétences que vous allez pouvoir mettre en œuvre avec la situation professionnelle 1

1) Qu'est-ce qu'un projet web ?	Les contraintes d'un projet web Le cycle de vie d'un projet web	
2) La conduite de projet web	La planification	C1.4.1.1 Établir son planning personnel en fonction des exigences et du déroulement du projet C1.4.1.2 Rendre compte de son activité C1.4.3.1 Recenser les ressources humaines, matérielles, logicielles et budgétaires nécessaires à l'exécution du projet et de ses tâches personnelles
	Le suivi	C1.4.2.1 Suivre l'exécution du projet C1.4.2.2 Analyser les écarts entre temps prévu et temps consommé C1.4.2.3 Contribuer à l'évaluation du projet
	Le pilotage et le management	C1.4.3.1 Recenser les ressources humaines, matérielles, logicielles et budgétaires nécessaires à l'exécution du projet et de ses tâches personnelles C1.4.3.2 Adapter son planning personnel en fonction des ressources disponibles
3) La phase de lancement	Le cahier des charges	C1.1.1.1 Recenser et caractériser les contextes d'utilisation, les processus et les acteurs sur lesquels le service à produire aura un impact
4)		
5) La phase de conception	La conception fonctionnelle	C1.2.4.1 Recenser les tests d'acceptation nécessaires à la validation du service et les résultats attendus C1.2.5.1 Recenser les utilisateurs du service, leurs rôles et leur niveau de responsabilité C1.2.5.2 Recenser les ressources liées à l'utilisation du service C1.2.5.3 Proposer les niveaux d'habilitation associés au service C4.1.2.1 Définir les spécifications de l'interface utilisateur de la solution applicative C4.1.2.2 Maquetter un élément de la solution applicative
	La conception graphique	C4.1.2.1 Définir les spécifications de l'interface utilisateur de la solution applicative C4.1.2.2 Maquetter un élément de la solution applicative
	La conception technique	C1.1.2.2 Recenser les composants de l'architecture technique sur lesquels le service à produire aura un impact
6) La phase de réalisation	La réalisation	C4.1.6.2 Mettre en place et exploiter un environnement de test
	Définition des environnements	C1.3.1.1 Mettre en place l'environnement de test du Service C4.1.6.2 Mettre en place et exploiter un environnement de test
	Les tests	C1.3.1.3 Rédiger le rapport de test C4.2.3.1 Élaborer et réaliser des tests d'intégration et de non régression de la solution mise à jour
	Les tests unitaires	C4.2.2.3 Élaborer et réaliser les tests unitaires des composants mis à jour
7) La phase d'exploitation	La documentation d'un projet	C4.1.9.1 Produire ou mettre à jour la documentation technique d'une solution applicative et de ses composants logiciels C4.1.10.1 Rédiger la documentation d'utilisation, une aide en ligne, une FAQ C4.1.10.2 Adapter la documentation d'utilisation à chaque contexte d'utilisation
8)		
9) Les bonnes pratiques	Le référencement L'ergonomie Le développement des pages web Le graphisme Les formulaires Les pages d'accueil La réglementation	C4.1.2.3 Concevoir et valider la maquette en collaboration avec des utilisateurs C5.2.1.2 Identifier et partager les bonnes pratiques à intégrer

f) Environnement de travail collaboratif

Afin de travailler en équipe, vous devrez utiliser un outil collaboratif pour les documents : Google Drive, Office 365...

Vous devez également utiliser une solution pour gérer les versions de votre code : SVN, GIT...

2) Contexte du PPE

La société CashCash vend des terminaux « point de vente » (TPV) à des hypermarchés, supermarchés et petits commerçants. CashCash compte de nombreux sites en France (centres régionaux et agences). Elle dispose d'un SI (système d'information) composé de plateformes hétérogènes (Mac OS, Linux, Windows, IOS, Android, etc) construit sur un réseau IP.

GESTION DE LA CLIENTÈLE

Gestion commerciale

Les TPV commercialisés par CashCash sont de deux types :

les petites caisses autonomes destinées aux petits commerçants.

les caisses bâties sur des processeurs Intel Atom , reliées à un serveur via un réseau ; ces caisses peuvent aussi travailler de façon autonome.

À ces machines, sont connectés des périphériques tels que les lecteurs optiques pour lire les codes à barres des produits, les lecteurs de cartes bancaires et les modems permettant de se connecter aux centres de gestion des cartes bleues.

Un matériel est identifié par son numéro de série et caractérisé par la date de sa vente, la date de son installation, son prix de vente et son emplacement dans le magasin du client (exemples : fruits et légumes, boulangerie...).

Chaque matériel appartient à un et un seul type, caractérisé par une référence interne et le libellé du type (exemple : la référence A506 correspond au type « souris sans fil logitech S »).

Chaque type de matériel est rattaché à une famille de produits. Une famille de produits est identifiée par le code de la famille et possède un libellé (exemples : PC, petite caisse ; CP, caisse avec processeur ; SE, serveur ; LO, lecteur optique ; CB, lecteur de cartes ; MO, modem ; etc).

Le client peut souscrire, auprès de CashCash, un contrat de maintenance pour la totalité ou une partie des matériels acquis auprès de cette société.

Un client est identifié par son numéro et se caractérise par sa raison sociale, son numéro de SIREN, son code APE, son adresse postale, un numéro de téléphone et un numéro de télécopie. Certains clients possèdent également une adresse de courrier électronique.

Tous les contrats de maintenance sont repérés par un numéro de contrat et mentionnent la date de signature et la date d'échéance. Un contrat couvre un ou plusieurs matériels achetés.

Tout matériel est garanti pendant un an à compter de sa date d'installation chez le client. À la fin de la période de garantie, le client peut souscrire un contrat de maintenance pour ce matériel, s'il ne l'a pas encore fait, ou demander à l'inclure dans le contrat existant, sans modification de la date d'échéance initialement prévue.

CashCash intervient pour maintenir et réparer les matériels à la demande des clients.; cela concerne tous les matériels inclus dans le contrat.

La durée des contrats est d'un an. Les contrats arrivant à expiration peuvent être renouvelés. La procédure de renouvellement d'un contrat se caractérise par l'enregistrement de la date du renouvellement et la modification de la date d'échéance du contrat ; on conserve toutefois, sans la modifier, la date à laquelle le contrat a été signé pour la première fois.

Le montant annuel facturé pour un contrat de maintenance est déterminé en appliquant un pourcentage au prix de vente des matériels couverts. Ce taux est de 10 %.

Gestion des interventions

Les interventions sont réalisées à partir des cinquante sept agences de la société CashCash.

Chaque client est rattaché à une agence. Pour faciliter la facturation des interventions, on enregistre la distance kilométrique qui sépare le client de son agence et la durée moyenne d'un déplacement exprimée en minutes.

Chaque agence est localisée dans l'une des dix régions géographiques couvertes par la société CashCash. Une agence est décrite par un numéro d'agence, qui l'identifie au sein de sa région, ainsi que par son nom, son adresse postale, son numéro de téléphone, son mail et son numéro de télécopie.

Chaque région (ou centre régional) est identifiée par son code et caractérisée par son nom.

Les appels téléphoniques des clients sont reçus au centre régional par des assistants. Chaque assistant doit être capable d'aider le client dans son diagnostic afin de faciliter l'intervention ultérieure. L'assistant dispose d'un système informatisé de suivi des interventions : il peut ainsi déterminer quel technicien interviendra et indiquer au client quel jour et à quelle heure s'effectuera l'intervention de ce technicien. L'assistant prévient le technicien concerné : la communication peut se faire à tout moment car chaque technicien possède un téléphone mobile.

Une intervention est toujours réalisée par un technicien de l'agence dont dépend le client. Le technicien se rend chez le client, au jour et à l'heure dits, effectue l'intervention et complète la « fiche d'intervention ». Dans une fiche « vierge », certaines rubriques sont en fait déjà renseignées : le numéro de la fiche qui l'identifie et le matricule du technicien.

Après l'intervention, le technicien complète la fiche avec les informations suivantes : numéro, nom et adresse du client, date et heure de la visite ainsi que, pour chaque

matériel vérifié, numéro de série du matériel, libellé du type de matériel, temps passé par le technicien pour vérifier ce matériel et un commentaire décrivant le travail effectué.

Comme tous les employés de la société CashCash, les techniciens et les assistants sont identifiés par un numéro de matricule et caractérisés par leur nom, leur prénom, leur adresse personnelle et la date de leur embauche dans la société.

Les assistants sont affectés à un centre régional, les techniciens à une agence.

Pour chaque technicien, il faut connaître sa qualification, la date d'obtention de cette qualification, son mail et le numéro de son téléphone mobile.

On vous signale des contraintes fortes au niveau des données :

Respecter la contrainte sur les interventions : pour un client donné une intervention ne peut être effectuée que sur un matériel couvert par un contrat signé par ce même client

Un technicien n'intervient que sur le matériel des clients gérés par son agence de rattachement.

3) Situations professionnelles à mettre en œuvre

Situation professionnelle 1	Application web de gestion des interventions
------------------------------------	---

La société CASHCASH possède 57 agences sur le territoire national. On comprend bien que le nombre d'interventions à gérer sera très important. De plus, chaque client étant rattaché à une agence, la société CASHCASH privilégie la proximité lors des déplacements. Par ailleurs, la réactivité de la société CASHCASH est essentielle pour assurer la confiance qui la lie à ses clients. Son système d'information doit donc pouvoir gérer un maximum d'informations sur les interventions. Une application web a été choisie puisqu'elle permet de s'adapter à l'hétérogénéité des solutions techniques d'accès (STA) des salariés. Cela peut être aussi bien des smartphones, des PC plus ou moins performants, des tablettes. Enfin, ces STA fonctionnent sur des systèmes d'exploitation différents (Windows, Linux, Android...).

Deux acteurs interviennent sur la gestion des interventions :

Les assistants assurent des tâches en back office

Les agents assurent des tâches en front office

Des droits doivent donc être accordés. Une authentification du salarié sera donc nécessaire.

Les assistants devront pouvoir rechercher une fiche client par rapport au numéro du client pour la visualiser ou la modifier.

Les assistants pourront affecter les visites à un technicien (faisant partie de la même agence que le client).

Les assistants pourront générer une fiche intervention au format pdf juste après l'affectation ou ultérieurement.

Les assistants pourront consulter les interventions par date ou par agent. C'est à ce moment-là qu'ils pourront à nouveau éditer la fiche d'intervention.

Les techniciens peuvent consulter les interventions qui leur sont affectées. Les plus proches de l'agence seront à traiter en priorité.

Les techniciens peuvent valider les interventions en saisissant un commentaire et le temps passé.

Les assistants ont un outil statistique qui leur permet de visualiser les informations suivantes par technicien : le nombre d'interventions réalisées, le nombre de kilomètres parcourus et la durée passée à contrôler le matériel, le tout pour un mois donné

Situation professionnelle 2	Création d'un client lourd pour la gestion du matériel
------------------------------------	---

Annexe à utiliser :

Exemple de fichier XML à générer pour un client

Descriptif des classes pour le client lourd

La société CASHCASH a vendu et déployé de nombreux équipements à ses clients. Ces équipements sont fournis avec ou sans contrat de maintenance. Certains de ces clients, les hypermarchés par exemple, possèdent de nombreux terminaux. La maintenance du matériel acheté est donc assurée en interne ou par la société CASHCASH. Par soucis de rationalisation, ces clients souhaiteraient pouvoir intégrer les données concernant les matériels (date d'installation, date d'échéance du contrat de maintenance...) à leur propre système d'information. Le format de données privilégié est le XML.

Par ailleurs, l'expertise technique de la société CASHCASH montre qu'il est fréquemment intéressant pour un client d'externaliser la maintenance de son matériel. La société CASHCASH s'attend donc à couvrir de nombreux matériels déjà vendus par des contrats de maintenance. Les commerciaux transmettent en général par téléphone ou par mail les informations utiles.

Enfin, pour d'autres clients, les petits commerçants par exemple, une relance plus classique par courrier est envisagée. La durée des contrats est d'un an. Les contrats arrivant à expiration peuvent être renouvelés. La procédure de renouvellement d'un contrat se caractérise par l'enregistrement de la date du renouvellement et la modification de la date d'échéance du contrat ; on conserve toutefois, sans la modifier, la date à laquelle le contrat a été signé pour la première fois.

La société CASHCASH souhaiterait posséder une application de type Client lourd avec interface graphique assurant deux fonctionnalités :

Le gestionnaire doit pouvoir générer le fichier xml pour un client donné

Le gestionnaire doit pouvoir couvrir par un contrat de maintenance les matériels déjà vendus.

Le gestionnaire doit pouvoir générer des courriers automatiques de relance au format pdf

Situation professionnelle 3	Développement d'une application mobile native
------------------------------------	--

La société CASHCASH a de nombreux techniciens qui sont auprès de leurs clients. L'enquête de septembre 2017 de l'ARCEP (Autorité de Régulation des Communications Electroniques et des Postes) a révélé la présence de nombreuses zones blanches sur le territoire national. Cela rend l'accès difficile au système d'information pour les techniciens qui sont sur le terrain. Le problème principal est qu'ils ne peuvent pas transmettre les informations collectées sur place. Ils sont obligés de faire des saisies papiers puis de ressaisir les données. Cela engendre à la fois des erreurs et une perte de temps.

La société CASHCASH compte investir dans de nouveaux smartphones pour ses techniciens. Le choix du modèle vous appartient donc.

On vous soumet quelques contraintes techniques :

Donner le schéma de la BDD SQLite locale qu'un technicien devra utiliser pour sauvegarder les éléments de son intervention et le code permettant la sauvegarde.

Synchroniser les sauvegardes locales des techniciens et les données sur le serveur de BDD.

Consulter via son Smartphone la BDD sur le serveur.

Situation professionnelle transversale	Gestion des incidents
---	------------------------------

Afin de

- faciliter les échanges entre CASHCASH et VDEV pour ce qui concerne des anomalies ou des incidents de fonctionnement relatifs aux applications que VDEV va réaliser pour CASHCASH,
- garantir la traçabilité de chaque demande (nature de la demande, nom du technicien VDEV qui a pris en charge cette demande, statut et informations de suivi de la demande,),

VDEV a retenu la solution du logiciel GLPI pour permettre la déclaration des incidents et leur suivi.

La direction de VDEV vous demande :

- de mettre en place cette solution
- de la paramétrer en fonction des objectifs ci-dessous.
- d'effectuer l'ensemble des tests qui permettent de valider votre paramétrage par rapport aux objectifs fixés.

Le paramétrage à effectuer doit répondre aux objectifs suivants :

- Tous les clients de VDEV doivent être en mesure d'utiliser cet outil. CASHCASH est l'un d'entre eux, mais votre paramétrage doit permettre que d'autres clients y accèdent.
- Tous les salariés de CASHCASH (ou d'une entreprise Y pour laquelle VDEV a développé une application) qui utilisent vos applications doivent pouvoir saisir directement leurs anomalies ou incidents dans l'outil.

- Par contre la Direction de VDEV ne veut absolument pas qu'un client puisse avoir accès aux problèmes qui se sont posés chez un autre client !
- Au sein des équipes de développement de VDEV n'importe quel développeur peut être amené à prendre en charge un incident qui s'est produit chez n'importe quel client.
- Chez un client donné :
 - Un utilisateur ne peut accéder qu'aux tickets qu'il a lui-même saisi.
 - Mais, pour faciliter le suivi du projet et du contrat avec VDEV, un responsable chez le client est autorisé à avoir une vision de l'ensemble des problèmes qui ont été remontés au sein de son entreprise et uniquement au sein de celle-ci. Il ne peut donc pas accéder aux problèmes des autres clients conformément à la volonté de la Direction.
- Par ailleurs, au sein de VDEV, on doit pouvoir faire le point sur tous les problèmes signalés par n'importe quel client sur une application donnée.

4) Organisation de VDEV

VDEV est une ESN (Entreprise de services du Numérique). Depuis sa création, développe des applicatifs pour divers clients. Elle a par exemple développé des applications de gestion de terrasses pour différentes villes ou alors fait évoluer le système d'information d'une coopérative de producteurs de noix.

Les applicatifs développés sont également de natures diverses. Cela peut être une application Web permettant de réserver des terrasses par des établissements d'une ville. Cette application permet également la consultation des tarifs pratiqués. Cela peut être également une application conçue à l'aide d'un langage objet permettant de positionner des terrasses sur la carte d'une ville donnée, en fonction d'un certain nombre de critères. Enfin, VDEV a fait évoluer ses technologies pour tenir compte des terminaux mobiles.

Pour diversifier ses activités, VDEV, décide de s'intéresser désormais au marché des entreprises. Développer des applications pour des acteurs privés peut être générateur de profit selon les dirigeants de VDEV. Après une longue campagne de communication et marketing, un premier contrat est en passe d'être finalisé avec l'entreprise CASHCASH.

Contraintes techniques et organisationnelles au sein de VDEV

Organisation du projet :

Cycle en V ou méthodes agiles (SCRUM)

Pour la couche données :

VDEV souhaite retrouver les spécifications et contraintes techniques suivantes ;

- Un dictionnaire de données détaillé
- Un modèle conceptuel de données comportant les extensions Merise 2
- Un modèle relationnel
- un SGBD ayant des fonctionnalités avancées (contraintes de domaine, implémentation de PL/SQL...).
- Un script de création de la base et de ses objets avec un jeu d'essai conséquent
- Une gestion très stricte au niveau de la couche donnée vous est demandée.
- Implémenter la BDD dans deux environnements différents :
 - a- Windows / Mysql
 - b- Linux / Postgre

Pour une application de type client léger :

VDEV fournit des solutions adaptative (responsive).

Par ailleurs, VDEV améliore sans cesse l'ergonomie et l'attractivité des solutions web. Vous devez faciliter les saisies avec par exemple des champs auto complétés. Enfin, vous devez utiliser des contrôles graphiques modernes comme les DatePicker.

Fort de leur expérience dans le développement, les responsables de VDEV vous suggèrent d'utiliser les technologies suivantes : Javascript, Ajax, JQuery.

VDEV souhaite retrouver les spécifications et contraintes techniques suivantes ;

- Une règle de nommage des variables
- Un diagramme de cas d'utilisation
- Une description textuelle des cas d'utilisation
- Un maquettage des IHM (les composants doivent être nommés)
- Le choix des Architecture logicielles retenues
- Une Charte graphique
- La compatibilité avec différents navigateurs.
- La visualisation correcte des informations sur terminaux mobiles
- Rapport de tests
- Un code commenté

VDEV souhaiterait développer idéalement ses solutions web sur un modèle MVC (Modèle Vue Contrôleur).

Gestion de certificats

<http://www.journaldunet.com/solutions/seo-referencement/1192830-risque-benefices-migration-https/>

Extrait de l'article du 14/03/17

C'est un fait, Google cherche à imposer aux sites Web de passer au HTTPS. Son moteur promet un bonus SEO aux pages en HTTPS et son navigateur a commencé à générer des nouvelles alertes lorsque certaines pages n'utilisent pas le chiffrement. Mais migrer vers ce protocole sécurisé peut être complexe. Quels sont les problèmes qui attendent les sites ? Les avantages ? Les conséquences en SEO sont-elles toujours bénéfiques ? Certains sites doivent-ils plus se presser que d'autres ? Le JDN a posé ces questions à trois consultants SEO, tous experts en technique.

Un boost SEO, vraiment ?

Google a annoncé en août 2014 que les pages en HTTPS allaient bénéficier d'un petit bonus SEO. Les professionnels du référencement naturel ont-ils pu observer la prime promise ? "Il ne faut pas s'attendre à avoir de bien meilleures performances en SEO en optant pour le HTTPS", prévient Madeline Pinthon, consultante SEO chez iProspect. "Il n'y a pas de progression systématique pour les sites passés au HTTPS. Lorsqu'on en observe une, elle est à peine visible. Et s'il y a de plus en plus de résultats HTTPS en première page, c'est parce que de plus en plus de sites sont passés au HTTPS", argumente-t-elle. L'agence Résonéo a pu effectuer plusieurs migrations vers le HTTPS, "et nous n'avons jamais vu rien de transcendant en termes de progression SEO", témoigne Aymeric Bouillat, consultant SEO et expert technique au sein de cette entreprise.

VDEV souhaiterait idéalement que ses solutions web soient bien référencées.

Pour une application de type client lourd :

VDEV souhaite retrouver les spécifications et contraintes techniques suivantes ;

- Un diagramme de classe
- Une code commenté
- Une gestion fine des erreurs
- Une documentation technique au format HTML
- Un diagramme de cas d'utilisation
- Une description textuelle des cas d'utilisation
- Un diagramme de classe UML
- Un maquettage des IHM (les composants doivent être nommés)
- Le choix des Architecture logicielles retenues
- Rapport de tests
- Tests unitaires

Vdev souhaite retrouver un certain nombre de règles de développement

Règles de développement pour un client lourd

- 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.
- 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.
- 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.
- 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.
- 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.
- 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.

Pour une application mobile :

VDEV souhaite retrouver les spécifications et contraintes techniques suivantes ;

- Un diagramme de cas d'utilisation
- Une description textuelle des cas d'utilisation
- Un maquetage des IHM (les composants doivent être nommés)
- Le choix des Architecture logicielles retenues
- Code commenté
- Rapport de tests

5) Annexes utiles

Annexe - Exemple de fichier XML à générer pour un client (materielClientcli15432.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<listeMateriel>
<materiels idClient="cli15432" >
  <sousContrat>
    <materiel numSerie="00213">
      <type refInterne="A506" libelle=" souris sans fil logitech S  />
      <famille codeFamille="SE" libelle="serveur" />
      <quantite>50</quantite>
      <date_vente>10-03-17</date_vente>
      <date_installation>12-05-17</date_installation>
      <prix_vente>32.5</prix_vente>
      <emplacement>"Boulangerie" </emplacement>
      <nbJourAvantEcheance>94</nbJourAvantEcheance >
    </materiel>
    <materiel numSerie="00214">
      <type refInterne="B106" libelle=" clavier sans fil logitech S  />
      <famille codeFamille="SE" libelle="serveur" />
      <date_vente>10-04-17</date_vente>
      <date_installation>17-05-17</date_installation>
      <prix_vente>18.5</prix_vente>
      <emplacement>"Boulangerie" </emplacement>
      <nbJourAvantEcheance>94</nbJourAvantEcheance >
    </materiel>
  </sousContrat>
  <horsContrat>
    <materiel numSerie="00215">
      <type refInterne="A506" libelle=" Datalogic QW2100 Datalogic
QuickScan Lite/>
      <famille codeFamille="PC" libelle="petite caisse" />
      <date_vente>10-03-17</date_vente>
      <date_installation>16-06-17</date_installation>
      <prix_vente>68.26</prix_vente>
      <emplacement>"Boulangerie" </emplacement>
    </materiel>
  </horsContrat>
</listeMateriel>
```

Annexe – Descriptifs des classes pour le client lourd

Classe PersistanceSQL

```
Public
  // Constructeur
  PersistanceSQL (ipBase : chaîne, port : entier, nomBaseDonnee : chaîne)
  // Construit un objet PersistanceSql. Cet objet permettra de charger les
  données depuis une base
  // de données ou de les sauvegarder dans la base.

  procédure RangerDansBase (unObjet : Objet)
  // Stocke les données de l'objet dans la base de données.

  fonction ChargerDepuisBase (id : chaîne, nomClasse : chaîne) : Objet de la
  classe NomClasse.
  // Retourne l'objet de la classe NomClasse dont l'identifiant est "id".
  Cet objet est chargé
  // depuis la base de données, ainsi que l'ensemble de ses objets liés (voir
  l'exemple d'utilisation
  // ci-dessous). Retourne NULL si aucun objet de cette classe ne possède
  cet identifiant.
```

Fin classe

Exemple d'utilisation :

```
// persist est une instance de PersistanceSQL
Client leClient ← persist.chargerDepuisBase ("2", "Client")
// leClient est l'instance de la classe Client dont l'identifiant est 2.
```

Toutes les commandes du distributeur sont automatiquement chargées dans la collection *leClient.lesMateriels*. Toutes les données utiles sont également chargées.

Les constructeurs, accesseurs et mutateurs n'ont pas été décrits. Ils sont bien évidemment à ajouter.

Vous pourrez ajouter toute méthode que vous jugerez utile.

Classe Client

Privé

```
    numClient : chaîne
    raisonSociale : chaîne
    siren : chaîne
    codeApe : chaîne
    adresse : chaîne
    telClient : chaîne
    email : chaîne
    dureeDeplacement : entier
    distanceKm : entier
    lesMateriels : Collection de <Materiel> // Tous les matériels du client.
    leContrat : ContratMaintenance // peut être nul si le client ne possède
pas de contrat
```

Public

```
    fonction getMateriels() : Collection de <Materiel>
    // Retourne l'ensemble des matériels du client

    fonction getMaterielsSousContrat() : Collection de <Materiel>
    // Retourne l'ensemble des matériels pour lesquels le client a souscrit un
contrat de maintenance qui
    // est encore valide (la date du jour est entre la date de signature et la
date d'échéance)

    fonction estAssure() : booléen
    // Retourne vrai si le client est assuré, faux sinon
```

Fin classe

Classe ContratMaintenance

Privé

```
    numContrat : chaîne
    dateSignature : date
    dateEcheance : date
    lesMaterielsAssures : Collection de <Materiel> // Tous les matériels sous
contrat de maintenance
```

Public

```
    fonction getJoursRestants () : entier
    // Renvoie le nombre de jours avant que le contrat arrive à échéance

    fonction estValide () : booléen
    // indique si le contrat est valide (la date du jour est entre la date de
signature et la date d'échéance)

    procedure ajouteMateriel (Materiel unMateriel)
    // ajoute unMatériel à la collection lesMaterielsAssures si la date de
signature du contrat est // antérieure à la date d'installation du
matériel
```

Fin classe

Classe Materiel

Privé

```
numSerie : entier  
dateVente : Produit  
prixVente : réel  
emplacement : chaîne  
leType : TypeMateriel
```

Public

```
fonction xmlMateriel () : chaîne  
// Retourne la chaîne correspondant au code XML représentant le matériel  
(voir annexe).
```

Fin classe

Classe TypeMateriel

Privé

```
referenceInterne : chaîne  
libelleTypeMateriel : chaîne  
laFamille : Famille
```

Public

// libre

Fin classe

Classe Famille

Privé

```
codeFamille : chaîne  
libelleFamille : chaîne
```

Public

//libre

Fin classe

Classe GestionMateriels

```
Privé
    donnees : PersistenceSQL
    // Attribut qui permet de rendre les objets métiers accessibles.

Public
    //Constructeur
    GestionMateriels (lesDonnees : PersistenceSQL)
    // Construit un objet GestionMateriels avec un modèle de persistance
    associé.

    fonction getClient (idClient : entier) : Client
    // Retourne l'objet Distributeur qui possède l'identifiant idDistributeur
    passé en paramètre,
    // retourne null si aucun Distributeur ne possède cet identifiant.

    fonction XmlClient (unClient : Client) : chaîne
    // Retourne une chaîne de caractères qui représente le document XML de la
    liste des matériels
    // du client passé en paramètre comme le montre l'exemple de l'annexe.

    fonction statique XmlClientValide (xml : chaîne) : booléen
    // Retourne un booléen Vrai si le fichier xml respecte la DTD référencée
    dans le fichier XML, Faux sinon
```

Fin classe