BREVET DE TECHNICIEN SUPÉRIEUR
SERVICES INFORMATIQUES AUX ORGANISATIONS
Option : Solutions logicielles et applications métiers

# U7 – CYBERSÉCURITÉ DES SERVICES INFORMATIQUES

SESSION 2025

_____

Durée : 4 heures
Coefficient : 4

_____

Matériel autorisé :
Aucun matériel ni document est autorisé.

Dès que le sujet vous est remis, assurez-vous qu'il est complet.

Le sujet comporte 22 pages, numérotées de 1/22 à 22/22.

# Puces’in

Ce sujet comporte 22 pages dont un dossier documentaire de 12 pages.

**Barème**

| | | |
|---|---|---|
| DOSSIER A | Sécurisation des applications *VerifColis* et *Puces’In Rachète* | 32 points |
| DOSSIER B | Gestion des droits sur les données à caractère personnel | 18 points |
| DOSSIER C | Résolution d'incidents sur le site web *pucesin.com* | 30 points |
| | TOTAL | 80 points |

## Dossier documentaire


***Documents communs à plusieurs dossiers*** ..... ***11***

**Document commun 1 : Processus de rachat de livres** ..... **11**

**Document commun 2 : Base de données pucesin** ..... **12**

**Document commun 3 : Extrait de la documentation de MySQL** ..... **14**

***Documents associés au dossier A*** ..... ***14***

**Document A1 : Formulaire de l'application client lourd VerifColis** ..... **14**

**Document A2 : Extrait des tables EtatLivre et Revendre** ..... **16**

**Document A3 : Extrait des classes métier** ..... **16**

**Document A4 : Extrait des classes techniques utilisées par l'interface de programmation d'application (API) REST** ..... **17**

**Document A5 : Résultats des tests de l'interface de programmation d'application (API) REST** ..... **18**

**Document A6 : Composition d'un jeton (token) JWT** ..... **19**

**Document A7 : Tests de l'interface de programmation d'application (API) réalisés suite à l'intégration des jetons (tokens) JWT** ..... **20**

***Document associé au dossier B*** ..... ***20***

**Document B1 : Diagramme de classes et schéma entité-association de la base de données** ..... **20**

***Documents associés au dossier C*** ..... ***21***

**Document C1 : Ticket S-C-1092** ..... **21**

**Document C2 : Ticket S-A-0095** ..... **21**

**Document C3 : Programme utilisant les données de la base Whois pour prévenir l'administrateur système avant l’expiration de la propriété du nom de domaine pucesin.com** ..... **22**

**Document C4 : Extrait du fichier auth.log des traces (logs) d'authentification réussies ou non du mois de mars 2024** ..... **22**

**Document C5 : Extrait du fichier des traces (logs) d'audit de MySQL du mois d'avril 2024** ..... **22**

# Présentation du contexte

## L'organisation cliente

Puces'In[1] est une entreprise française fondée sur le modèle de l'économie circulaire qui récupère, collecte, rachète et revend des livres d'occasion.

Le livre est non seulement un produit culturel mais aussi un objet qui a une histoire et qui transporte de nombreuses émotions. Ce type de bien doit circuler et être partagé. Lutter contre le gaspillage de manière large, et promouvoir l'économie circulaire fait partie intégrante de l'engagement de Puces'In.

Spécialisée dans les livres (hors livres numériques), Puces'In dispose de sept centres de dépôt, tri et revente répartis sur tout le territoire national.

Puces'In se différencie de ses concurrents en proposant une application mobile *Puces'In Rachète*, disponible sur les magasins d'applications (*store*), qui offre aux particuliers la possibilité de lire les code-barres des livres qu'ils veulent revendre, valider leur panier et imprimer un bordereau d'expédition (appelé également étiquette d'envoi) pour le colis à expédier à un centre de tri Puces'In.

À la réception des colis, les employés des centres de tri de Puces'In utilisent une application de type client lourd *VerifColis* pour en vérifier le contenu.

Les livres rachetés par Puces'In sont mis en vente, soit en rayon dans la partie revente des centres de Puces'In, soit directement sur le site web *pucesin.com* et les sites de marché (*marketplaces*) associés.

## L'entreprise prestataire de service

La société Solut est une entreprise de services du numérique (ESN) labellisée ExpertCyber[2], spécialisée dans le développement d'applications web et mobiles, et l'amélioration de la sécurité d'applications existantes.

Jeune recrue de cette société, vous intervenez sur la sécurisation des applications *Puces'In Rachète et VerifColis*, ainsi que sur la résolution d'incidents remontés depuis le site web de l'entreprise.

[1] Pour des raisons de confidentialité, les noms des entreprises et les données afférentes ont été modifiées.

[2] Le label ExpertCyber vise à reconnaître l'expertise des experts en cybersécurité assurant des prestations d'installation, de maintenance et d'assistance en cas d'incident.

## Dossier A – Sécurisation des applications *VerifColis* et *Puces'In Rachète*

Le processus de rachat de livres, décrit dans le dossier documentaire, utilise deux applications :

- l'application client lourd *VerifColis* utilisée par les employés du centre de tri pour traiter les colis de livres reçus. Cette application écrite en *C#* utilise la base de données *pucesin* ;

- l'application mobile *Puces'In Rachète* qui permet aux particuliers de revendre des livres à Puces'In. Cette application existant en version Android et IOS utilise une interface de programmation d'application (API) REST (*Representational State Transfer*) développée en *Java* qui communique avec la base de données *pucesin*.

Des failles ont été détectées dans ces deux applications. Vous devez intervenir pour les sécuriser et appliquer des contre-mesures aux failles détectées.

### Mission A1 – Sécurisation de l'application client lourd *VerifColis*

Lors du contrôle d'un colis contenant des livres à racheter, lorsqu'un livre est jugé dans un état « acceptable », l'employé du centre de tri propose un prix de rachat qui sera transmis au client. Ce prix de rachat proposé sera enregistré dans la base de données.

Le formulaire de l'application *VerifColis* utilisé pour réaliser ce traitement est fourni dans le dossier documentaire.

Le service Qualité de Puces'In a récemment réalisé une étude pour connaître le taux de réponse des clients suite à ces propositions de prix de rachat.

Cette étude a fait apparaître une faille de sécurité : des prix de rachat proposés par un employé ont été validés avec une valeur supérieure au prix de revente enregistré pour la publication du livre.

**Question A.1.1**

Identifier la principale conséquence de cette malveillance pour Puces'In.

Pour contrer cette faille, votre chef de projet souhaite une adaptation de la méthode *traiterUnLivre* pour prendre en compte la règle de gestion suivante : « Le prix de rachat proposé ne sera enregistré dans la base que s'il ne dépasse pas 10 % du prix de revente de ce livre dans un état jugé acceptable ».

**Question A.1.2**

Lister les modifications à apporter à la méthode *traiterUnLivre* pour qu'elle intègre la règle de gestion demandée. Le code n'est pas demandé.

## Mission A2 – Sécurisation de l'application mobile *Puces'In Rachète*

Depuis trois semaines, Puces'In reçoit un nombre important de colis vides de la part de vendeurs.

Identifiés comme étant des concurrents de Puces'In, ces vendeurs utilisent l'application mobile pour lire le code-barres de livres et valider leur panier dans le but d'obtenir uniquement un bordereau d'expédition qu'ils collent sur un colis vide.

**Question A.2.1**

Identifier et justifier deux conséquences de cette malveillance pour Puces'In.

Pour éviter ce comportement malveillant, il a été décidé de limiter le nombre de bordereaux d'expédition à quatre par mois par vendeur.

Pour cela :

- La méthode *getNbBordereauDuMois* vient d'être développée et ajoutée dans la classe VendeurDAO pour ce traitement. Elle fournit le nombre de bordereaux d'expédition générés durant le mois en cours pour le vendeur dont l'identifiant est passé en paramètre.
- La signature de la méthode *validerColis* de la classe ColisController, responsable de la validation du colis, a déjà été modifiée afin de retourner une valeur booléenne. Pour l'instant, le code de la méthode ne permet pas de contrôler le nombre de bordereaux.

**Question A.2.2**

Compléter la méthode *validerColis* de la classe ColisController utilisée par l'interface de programmation d'application (API) REST afin qu'elle prenne en compte la limite fixée pour le nombre de bordereaux d'expédition.

## Mission A3 – Gestion de l'interface de programmation d'application (API) REST

L'application mobile *Puces'In Rachète* communique avec la base de données en utilisant une interface de programmation d'application (API) REST (*Representational State Transfer*).

Des tests de l'interface de programmation d'application (API) REST, présentés dans le dossier documentaire, ont été réalisés.

Deux tests présentent une vulnérabilité.

**Question A.3.1**

a) Identifier les deux tests présentant une vulnérabilité.

b) Expliquer, pour chacun de ces deux tests, la vulnérabilité de l'interface de programmation d'application (API) mise en évidence.

c) Indiquer, pour chacun de ces deux tests, si la vulnérabilité concerne l'intégrité, la disponibilité ou la confidentialité.

Votre responsable technique décide d'augmenter la sécurité de l'interface de programmation d'application (API) REST suite à vos remarques. En plus de la mise en place d'un système de prévention d'intrusion (IPS), il décide d'implémenter deux règles pour contrôler les requêtes vers l'interface de programmation d'application (API) REST :

- Une requête d'authentification doit être réalisée avant toute autre requête à l'interface de programmation d'application (API) REST.

- Un utilisateur devra être limité en nombre de requêtes vers les identifiants URI (*Uniform Resource Identifier*) de l'interface de programmation d'application (API) REST.

**Question A.3.2**

Indiquer si la méthode *authentifierUtilisateur* de la classe VendeurDAO est sécurisée en justifiant votre réponse.

**Question A.3.3**

Proposer une solution technique pour limiter le nombre de requêtes. Sa mise en œuvre n'est pas demandée.

L'équipe de développement décide d'utiliser des jetons d'authentification (*tokens*) de type *JSON Web Token* (JWT) pour implémenter ces deux règles.

**Question A.3.4**

Expliquer en quoi l'utilisation d'une clé de hachage par la fonction HMAC (Hash-based message authentification code) lors de la signature d'un jeton d'authentification (*token*) JWT garantit son intégrité.

Elle décide également d'utiliser le protocole HTTPS pour la communication avec l'interface de programmation d'application (API) REST.

**Question A.3.5**

Expliquer pour quelle raison l'utilisation du protocole HTTPS est nécessaire malgré le mode d'authentification par jeton (*token*).

L'intégration des jetons d'authentification (*tokens*) de type *JSON Web Token* (JWT) dans l'interface de programmation d'application (API) REST est en phase de test.

Lors des tests, des requêtes ont été envoyées à l'interface de programmation d'application (API) REST. Leurs caractéristiques sont fournies dans le dossier documentaire.

Le premier test fournit une réponse satisfaisante mais le deuxième est en échec.

**Question A.3.6**

Donner la cause de l'échec du deuxième test en justifiant votre réponse.

## Dossier B – Gestion des droits sur les données à caractère personnel

### Mission B1 – Contrôle du respect des obligations légales

Pour utiliser les applications *Puces'In Rachète* et *pucesin.com*, les vendeurs et les acheteurs doivent fournir des données à caractère personnel.

Le RGPD (articles 15 à 22) permet à toute personne ayant fourni des données à caractère personnel d'exercer des droits auprès de l'organisation les ayant collectées, comme par exemple :

- le droit d'accès ;
- le droit à la limitation du traitement ;
- le droit à la portabilité ;
- le droit d'opposition.

Actuellement, Puces'In reçoit de la part des vendeurs et des acheteurs des demandes relatives à l'exercice de droits sur leurs données à caractère personnel.

Le traitement de ces demandes transmises par courriel est réalisé par Mme Sylvie Mathy. Le temps moyen de délai de réponse est de 2 mois, bien au-delà du délai réglementaire.

**Question B.1.1**

a) Donner deux droits, manquants dans la liste, que les vendeurs et les acheteurs pourront également exercer.

b) Donner le délai règlementaire dont dispose Mme Mathy pour répondre aux demandes d'exercice des droits sur les données à caractère personnel transmises par les vendeurs et les acheteurs.

Pour être traitée, une demande doit obligatoirement être accompagnée de la photocopie, au format PDF, d'un document officiel (passeport, carte d'identité, etc.) prouvant l'identité de la personne à l'origine de la demande.

Actuellement, Mme Mathy saisit dans un tableur les caractéristiques des demandes (nom et prénom du demandeur ainsi que le droit exercé).

Lorsque la demande est traitée, Mme Mathy en informe le demandeur par courriel en lui fournissant, le cas échéant, les informations nécessaires.

Ce tableur et les fichiers contenant les documents officiels sont enregistrés dans un répertoire situé sur son poste de travail. En cas de besoin (par exemple : la demande n'est pas explicite ou la photocopie ne permet pas d'identifier le demandeur, etc.), Mme Mathy est amenée à échanger avec le demandeur. Seuls les échanges réalisés par courriel sont conservés dans sa boîte de messagerie durant un mois.

Depuis 2018, Mme Mathy a conservé, par précaution, tous les fichiers prouvant l'identité du demandeur.

**Question B.1.2**

Indiquer si le choix de Mme Mathy de conserver, depuis 2018, les fichiers prouvant l'identité des demandeurs est règlementaire. Justifier votre réponse.

Mme Mathy est consciente que cette manière de gérer les demandes ne permet pas de :

- garantir que le délai règlementaire fixé pour l'exercice d'un droit est bien respecté ;
- réaliser des statistiques sur la durée de traitement des demandes ;
- garder la trace des échanges avec le demandeur lors du traitement de la demande.

Aussi, elle a demandé une application web permettant, entre autres, la saisie d'une demande d'exercice de droits.

**Question B.1.3**

Indiquer dans quel document le traitement de saisie des demandes d'exercice des droits sur les données à caractère personnel doit être décrit pour être conforme au RGPD.

### Mission B2 – Conception de la base de données "Gestion des droits sur les données à caractère personnel"

En plus de proposer la possibilité de saisir une demande d'exercice de droit, la nouvelle application web permettra à Mme Mathy de gérer correctement le délai réglementaire : lorsqu'elle se connectera à l'application, elle obtiendra les demandes non encore traitées, triées par date de saisie de la plus ancienne à la plus récente. D'autres fonctionnalités lui permettront de saisir les échanges avec les demandeurs et d'obtenir des statistiques sur le délai de traitement des demandes.

Une ébauche de représentation de la base de données utilisée par l'application a été réalisée. Vous allez la compléter pour permettre :

- la garantie de la gestion du délai règlementaire de traitement d'une demande ;
- la réalisation de statistiques sur la durée de traitement d'une demande par type de droit exercé ;
- la conservation des dates de changement de l’état d’une demande ;
- la conservation des échanges réalisés pour traiter la demande. Pour chaque échange on conservera la date de l'échange, la description de l'échange et un code indiquant la personne à l'origine de l'échange (D pour le demandeur, P pour le DPO de Puces'in).

**Question B.2.1**

Adapter la représentation de votre choix (diagramme de classes ou schéma entité-association) fournie dans le dossier documentaire.

## Dossier C – Résolution d'incidents sur le site web *pucesin.com*

Plusieurs tickets d'incident ont été enregistrés pour le site web principal de Puces'In à savoir *pucesin.com*. Votre chef de projet vous demande de participer à la résolution de certains d'entre eux.

### Mission C1 - Identification d'attaques et contre-mesures

Vous traitez le ticket S-C-1092 fourni dans le dossier documentaire.

**Question C.1.1**

a) Nommer la technique utilisée par l'attaquant.

b) Nommer l'attaque dont l'employé a été victime.

**Question C.1.2**

Donner deux moyens de sensibiliser les employés de Puces'In à ce type d'attaque.

Vous traitez le ticket S-A-0095 fourni dans le dossier documentaire.

**Question C.1.3**

Proposer deux contre-mesures que Puces'In aurait dû prendre pour éviter ce type d'attaque.

À la lecture de ce ticket, l'administrateur système du site *pucesin.com* a souhaité mettre en place un programme de vérification automatique du renouvellement de la propriété du nom de domaine. Pour ce faire, il a demandé à un stagiaire de développer un programme qui exploite les données de la base *Whois*, pour le prévenir 3 mois puis 1 mois avant la date d'expiration de la propriété du nom de domaine *pucesin.com*.

Après une première expérimentation, il constate qu'un rappel supplémentaire lui serait bien utile une semaine avant.

**Question C.1.4**

Compléter le programme pour que sa demande soit prise en compte.

## Mission C2 – Analyse de traces (*logs*)

À la fin du mois d'avril, vous recevez un courriel de la part du responsable du service comptabilité de Puces'In. Lors du mois de mars, le total des ventes indiqué par le site ne correspond pas aux sommes créditées sur les comptes de Puces'In.

En analysant les traces (*logs*) d'authentifications réussies ou non du mois de mars, vous identifiez quelques irrégularités.

**Question C.2.1**

Identifier le type d'attaque subi par le serveur hébergeant le site *pucesin.com.*

**Question C.2.2**

Identifier deux raisons possibles du succès de l'attaque.

**Question C.2.3**

Pour chaque principe de sécurité informatique, donner un risque possible dû à l'attaque.

La lecture du fichier des traces (*logs*) d'audit du SGBD MySQL du mois d'avril révèle plusieurs requêtes effectuées sur la base de données qui ont engendré une fuite de données.

**Question C.2.4**

a) Lister les acteurs devant être informés de la fuite de données.

b) Indiquer dans quel document la fuite de données doit être consignée.

c) Indiquer deux actions à réaliser pour garantir l'intégrité des preuves pour les autorités compétentes.

**Question C.2.5**

Écrire la requête permettant de supprimer les privilèges attribués abusivement par ces requêtes.

## Mission C3 – Sécurisation des accès à une base de données

Le prix de revente des livres de Puces'In varie avec la demande : par exemple, les bandes-dessinées et les mangas, en plus forte demande, ont un prix de revente plus élevé, tout comme d'autres livres d'auteurs populaires (Marc Levy, Michel Bussi, etc.). En revanche les livres de poche et les œuvres de référence, comme celles demandées par les enseignants en classe de première, ont un prix de revente très bas.

Pour déterminer les prix de revente, Puces'In a besoin de faire des statistiques sur les données. Pour ces statistiques, les informations suivantes sont nécessaires : livres commandés, date de publication et prix de vente neuf des publications des livres, libellés des types de publication.

Aussi, il est nécessaire de créer un compte utilisateur *statistiques* disposant d'un accès à la base de données nommée *pucesin* pour obtenir les informations utiles aux statistiques.

**Question C.3.1**

a) Écrire les requêtes permettant de créer le compte utilisateur *statistiques* qui ne peut se connecter qu'à partir de l'adresse 95.10.2.54 pour accéder uniquement aux tables utiles aux statistiques.

b) Proposer une solution technique différente, sans la mettre en œuvre, pour que le compte utilisateur *statistiques* ne puisse accéder qu'aux données nécessaires aux statistiques et à aucune autre.

**Documents communs à plusieurs dossiers**

## ***Document commun 1 : Processus de rachat de livres***

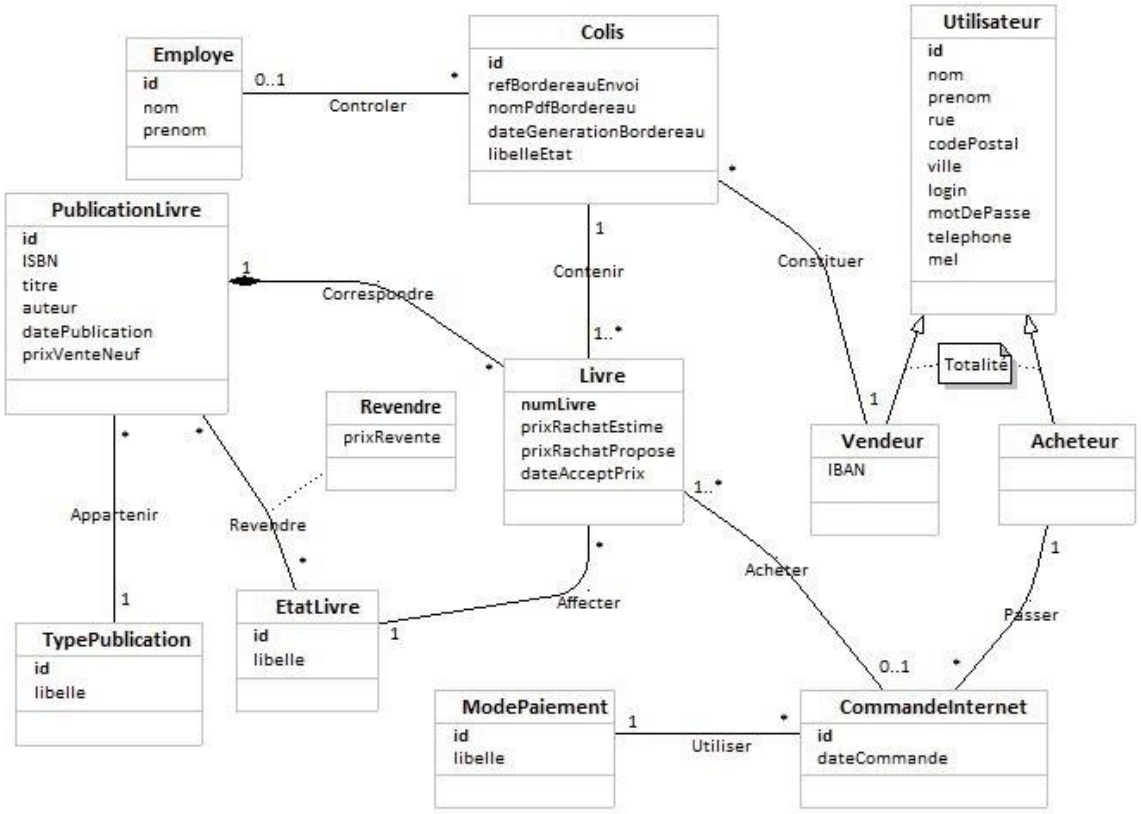
Un livre est un exemplaire d'une publication. Chaque publication de livre est caractérisée par un numéro d'identification unique ISBN (*International Standard Book Number*), un titre, un auteur, une date de publication, un type de publication (poche, broché, bande dessinée, manga, etc.) et un prix de vente neuf. Pour chaque publication d'un livre, plusieurs prix de revente sont définis, en fonction de l'état du livre (son niveau d'usure), et de la demande liée à la publication. Par exemple, les mangas étant plus demandés que les livres de poche, ils sont revendus plus chers.

Le processus de rachat de livres est le suivant :

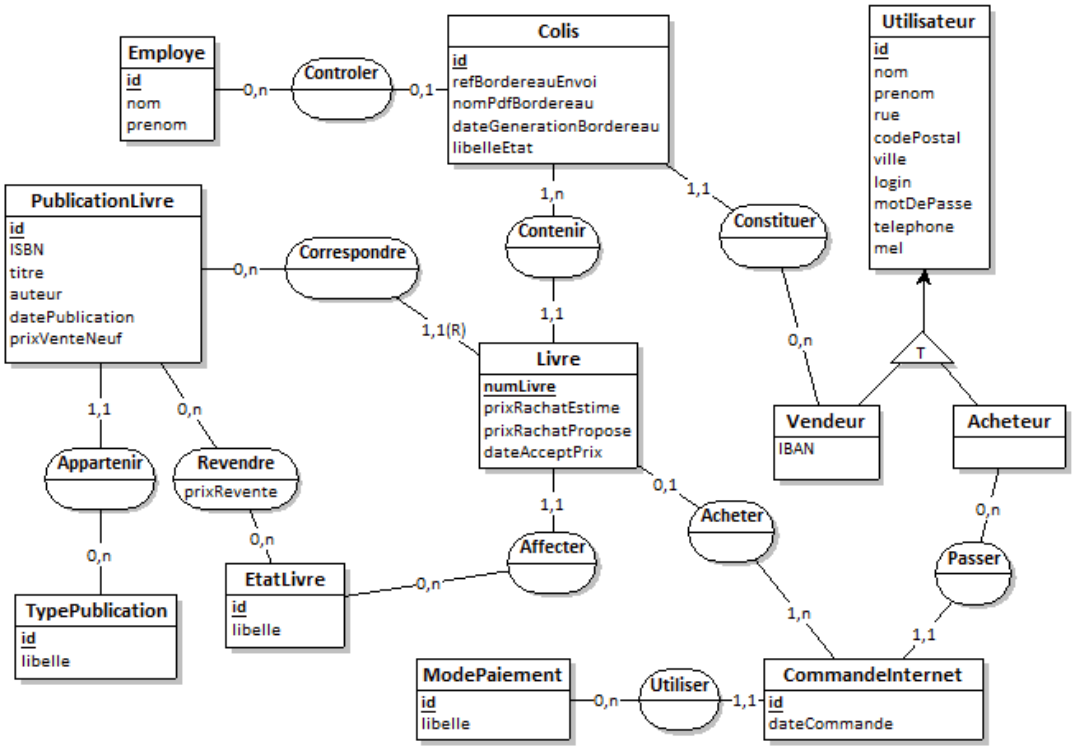
1. Le particulier qui souhaite revendre des livres à Puces'In doit lire le code-barres de chacun d'eux avec l'application *Puces'In Rachète* qu'il aura préalablement installée.
2. L'application affiche le prix de rachat estimé pour chaque livre. Ce prix correspond à 10 % du prix de revente d'un livre en « bon état » à la clientèle (un prix de revente est défini pour chaque publication de livre et chaque état). Exemple : si le prix de revente d'une publication pour un livre en « bon état » est fixé à 10 € alors le prix de rachat estimé d'un livre appartenant à cette publication sera de 1 €.
3. Une fois que le vendeur a traité tous les livres qu'il souhaite revendre, il valide son panier.
4. L'application lui propose alors de se connecter, ou de créer un compte. Dans ce dernier cas, son adresse postale ainsi que l'identifiant international de son compte bancaire (IBAN) lui seront demandés.
5. Une fois que le panier est validé, un bordereau d'expédition (appelé également étiquette d'envoi) est généré pour le vendeur, qui peut alors l'imprimer et le coller sur le colis contenant les livres du panier. Le bordereau d'expédition contient un code-barres permettant d'identifier le colis.
6. Le client expédie le colis à Puces'In qui le réceptionnera dans son centre de tri.
7. Les frais de port seront réglés par Puces'In dès que le client aura déposé le colis au bureau de poste ou au point relais de son choix.
8. À la réception du colis, l'état du colis est mis à jour dans la base de données et passe à l'état « reçu ».
9. Puis, un employé du centre de tri est chargé de renseigner l'état de chaque livre présent dans le colis. Cet état peut prendre une des valeurs suivantes : « comme neuf », « très bon état », « bon état », « acceptable », « refusé ». Si le livre manque dans le colis, alors il prendra l'état « absent ».
   Si un livre est dans un état « acceptable », alors l'employé du centre de tri saisira une nouvelle proposition de prix de rachat, enregistrée dans le champ prixRachatPropose, et l'enverra au client par messagerie. Si le client ne valide pas cette proposition dans un délai d'un mois, alors le livre sera retourné au client.
   Si un livre est dans un état « refusé », alors il sera également retourné au client.
10. Le client sera réglé :
    - immédiatement si dans le colis tous les livres ont un état supérieur à l'état « acceptable » (« comme neuf », « très bon état » ou en « bon état ») ;
    - ou suite à l'acceptation du client de la nouvelle proposition de prix de rachat si un des livres du colis est dans un état « acceptable ».
11. Les livres sont ensuite nettoyés.
12. Le service comptabilité de Puces'In procède alors au virement du montant à régler au client à l'aide d'un logiciel dédié.

## ***Document commun 2 : Base de données pucesin***

### Diagramme de classes




### Schéma entité-association



## Schéma relationnel

**Vendeur** (id, nom, prenom, rue, codePostal, ville, telephone, mel, login, motDePasse, IBAN)
Clé primaire : id

**Acheteur** (id, nom, prenom, rue, codePostal, ville, telephone, mel, login, motDePasse)
Clé primaire : id

**Employe** (id, nom, prenom)
Clé primaire : id

**Colis** (id, refBordereauEnvoi, nomPdfBordereau, dateGenerationBordereau, libelleEtat, idVendeur, idEmploye)
Clé primaire : id
Clés étrangères : idVendeur en référence à id de Vendeur
idEmploye en référence à id de Employe

**TypePublication** (id, libelle)
Clé primaire : id

**PublicationLivre** (id, ISBN, titre, auteur, datePublication, prixVenteNeuf, idTypePublication)
Clé primaire : id
Clé étrangère : idTypePublication en référence à id de TypePublication

**EtatLivre** (id, libelle)
Clé primaire : id
*Voir le document A2 pour les valeurs prises par le champ libelle*

**Revendre** (idPublication, idEtatLivre, prixRevente)
Clé primaire : idPublication, idEtatLivre
Clés étrangères :idPublication en référence à id de PublicationLivre
idEtatLivre en référence à id de EtatLivre
*Voir le document A2 pour une définition en extension de la table*

**Livre** (idPublication, numLivre, prixRachatEstime, prixRachatPropose, dateAcceptPrix, idEtatLivre, idColis, idCommande)
Clé primaire : idPublication, numLivre
Clés étrangères : idPublication en référence à id de PublicationLivre
idEtatLivre en référence à id de EtatLivre
idColis en référence à id de Colis
idCommande en référence à id de CommandeInternet

**ModePaiement** (id, libelle)
Clé primaire : id

**CommandeInternet** (id, dateCommande, idAcheteur, idModePaiement)
Clé primaire : id
Clés étrangères : idAcheteur en référence à id de Acheteur
idModePaiement en référence à id de ModePaiement

## ***Document commun 3 : Extrait de la documentation de MySQL***

**CURRENT_DATE**()

Cette fonction retourne la date courante du système sous le format « *année-mois-jour* ».

*par exemple elle retourne : 2024-03-10 si nous sommes le 10 mars 2024*

**Création d'un compte utilisateur**

```
CREATE USER 'jeffrey'@'localhost' IDENTIFIED BY 'password'; -- crée un
```

*utilisateur 'jeffrey' qui se connecte avec le mot de passe 'password' depuis la machine locale.*

**Attribution de droits à un compte utilisateur**

```
GRANT ALL ON base.table TO 'jeffrey'@'localhost';
```

*-- donne à l'utilisateur 'jeffrey' tous les droits sur la table « table » de la base « base ».*

**Révocation de droits à un compte utilisateur**

```
REVOKE UPDATE ON base.table FROM 'jeffrey'@'localhost';
```

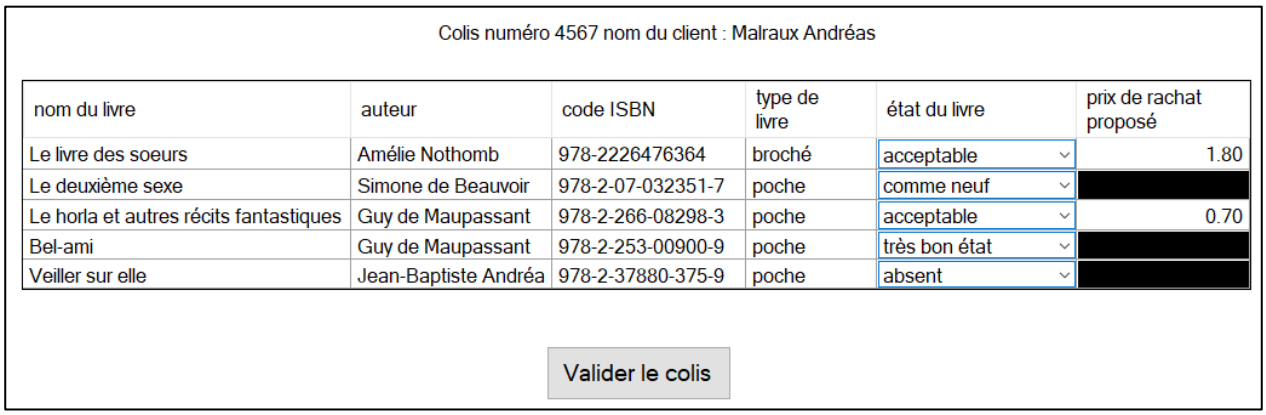
*-- supprime le privilège UPDATE sur les enregistrements à l'utilisateur 'jeffrey' dans la table « table » de la base « base ».*

## Documents associés au dossier A

## ***Document A1 : Formulaire de l'application client lourd VerifColis***

Lorsqu'un employé du centre de tri des livres lit le code-barres du bordereau d'expédition présent sur un colis, un formulaire s'affiche contenant les livres traités par le client.

Exemple de données affichées sur le formulaire :




Pour chaque livre du colis, l'employé du centre de tri doit :

- renseigner l'état du livre en sélectionnant une valeur dans la liste déroulante ;
- saisir un prix de rachat qui sera proposé au client si l'état du livre est « acceptable ».

Traitement du formulaire :

Au clic sur le bouton « Valider le colis » la méthode *traiterUnLivre* fournie ci-dessous est appelée pour chaque ligne de la grille de données.

Elle reçoit deux paramètres : un objet unLivre correctement valorisé avec les valeurs du livre d'une ligne et un objet contenant les caractéristiques utiles à la connexion à la base de données.

```
private int traiterUnLivre(Livre unLivre, MySqlConnexion uneConnexion) {
    int valeurRetournee;
    string reqSQL;

    // test si le livre est dans l'état « acceptable »
    if (unLivre.getLeEtatLivre().getId() == 4)
    {
        // mise en forme de la requête SQL
        // les paramètres de la requête sont précédés du caractère @
        reqSQL = "update Livre set idEtat = @pEtat, prixRachatPropose = @pPrix
                where numLivre = @pLivre and idPublication= @pPub";
    }
    else
    {
       // mise en forme de la requête SQL
       reqSQL = "update Livre set idEtat = @pEtat
                where numLivre = @pLivre and idPublication= @pPub";
    }
    // mise en forme de la requête SQL dans l'objet uneComSql
    MySqlCommand uneComSql= new MySqlCommand(reqSQL, uneConnexion);
    // renseigne les paramètres à partir des valeurs de l'objet unLivre
    uneComSql.Parameters.AddWithValue("@pEtat", unLivre.getLeEtatLivre().getId());
    if (unLivre.getLeEtatLivre().getId() == 4)
    {
       // si l'état du livre est acceptable, le prix de rachat proposé saisi est enregistré
       uneComSql.Parameters.AddWithValue("@pPrix", unLivre.getPrixRachatPropose());
    }
    uneComSql.Parameters.AddWithValue("@pLivre", unLivre.getNumLivre());
    uneComSql.Parameters.AddWithValue("@pPub", unLivre.getLaPublication().getId());
    // Exécution de la requête contenue dans l'objet uneComSql
    // la méthode ExecuteNonQuery retourne le nombre d'enregistrements modifiés
    valeurRetournee = uneComSql.ExecuteNonQuery();
    return valeurRetournee;
}
```

***Document A2* : Extrait des tables EtatLivre et Revendre**

La table EtatLivre contient les états possibles d'un livre :

| id | libelle |
|---|---|
| 1 | comme neuf |
| 2 | très bon état |
| 3 | bon état |
| 4 | Acceptable |
| 5 | Refusé |
| 6 | absent |

La table Revendre contient le prix de revente d'une publication en fonction de l'état du livre. Dans l'exemple ci-dessous, on peut voir les prix de revente enregistrés dans la table Revendre pour le livre d'Amélie Nothomb intitulé « Le livre des sœurs » (identifiant de publication = 256) :

| idPublication | idEtatLivre | prixRevente |
|---|---|---|
| 256 | 1 | 13.50 |
| 256 | 2 | 13.00 |
| 256 | 3 | 12.30 |
| 256 | 4 | 9.80 |

***Document A3* : Extrait des classes métier**

Seuls les classes métier, attributs et méthodes nécessaires aux traitements demandés sont fournis ci-dessous.

Les méthodes de lecture (*getters*) de mise à jour (*setters*) ne sont pas fournies mais existent pour chaque attribut d'une classe.

```
public class Livre  {
    private int numLivre ;
    private float prixRachatEstime ;
    private float prixRachatPropose ;
    private PublicationLivre laPublication ;
    private Colis leColis ;
    private EtatLivre leEtatLivre ;
    public Livre(…)  { … }
    // suite de la classe
}
```

```
public class PublicationLivre  {
    private int id ;
    private String isbn ;
    private String titre;
    private String auteur;
    private Date datePublication ;
    private float prixVenteNeuf ;
    private TypePublication leTypePublication ;
    public PublicationLivre(…) {…}
    // suite de la classe
}
```

```
public class Vendeur  {
    private int id ;
    private String nom ;
    private String prenom ;
    public Vendeur(…)  { … }
    public int getId()  { return id ; }
    // suite de la classe
}
```

```
public class TypePublication  { … }

public class EtatLivre  { … }
```

```
public class Colis  {
        private int id ;
        private String refBordereauEnvoi ;
        private String nomPdfBordereau ;
        private Date dateGenerationBordereau ;
        private String libelleEtat ;
        private Vendeur leVendeur ;
        private Employe leEmp ;
        private ArrayList<Livre> lesLivres ;
        public Colis(…)  { … }
        public int getId()  { return id ; }
        public String getRefBordereauEnvoi()  { … }
        public String getNomPdfBordereau()  { … }
        public Date getDateGenerationBordereau()  { … }
        public String getLibelleEtat()  { … }
        public Vendeur getLeVendeur()  { return leVendeur ; }
        // suite de la classe
}
```

***<u>Document A4</u> : Extrait des classes techniques utilisées par l'interface de programmation d'application (API) REST***

```
public class VendeurDAO extends DAO {
      public static ResultSet authentifierUtilisateur(String login , String mdp){
            ResultSet unResultSet = null ;
            // appel de la méthode de la classe mère qui donne un objet pour gérer la BDD
            Connection uneConnexion = getConnexion();
            try {
                  // mise en forme de la requête préparée
                  String sql ="Select id, nom, prenom from Vendeur
                                          where login = ? and motDePasse = ?";
                  PreparedStatement unStatement = uneConnexion.prepareStatement(sql);
                  // renseigne les deux paramètres
                  unStatement.setString(1, login);
                  unStatement.setString(2, hashpassword(mdp));
                  // exécute la requête préparée
                  unResultSet = unStatement.executeQuery();

             } catch (SQLException e) {
                        // gestion de l'exception (code non fourni)
             }
             // retourne le jeu d'enregistrement
             return unResultSet;
      }
      /* retourne le nombre de bordereaux d'expédition générés pour le mois en cours
      /*  pour le vendeur dont l'identifiant est passé en paramètre */
      public static int getNbBordereauDuMois(int unIdVendeur) { … }
}
```

```
public class ColisDAO  extends DAO {
   /* enregistre un colis avec ses livres */
   public static void enregistrerColisAvecLivres(Colis unColis) {…}
   ….
}
```

```
public class ColisController {
    …
    // validerColis permet la génération du bordereau d'expédition et la création des données
    // dans la base de données si le vendeur n'a pas déjà généré quatre bordereaux pour le mois
    //  en cours. LeColis est une instance de la Classe Colis.
    // La méthode retourne false si la limite a été atteinte.
    public Boolean validerColis(Colis leColis) {
        int  idVendeur = leColis.getLeVendeur().getId() ;  // récupération de l'identifiant du vendeur
        /* à compléter sur votre copie */
        genererPdfBordereau (leColis) ;  // génération du PDF
        // enregistrement des données dans la base
        ColisDAO.enregistrerColisAvecLivres(leColis) ;
        return true;
     }
}
```

***Document A5 : Résultats des tests de l'interface de programmation d'application (API) REST***

| Test n°1 : test permettant d'obtenir les caractéristiques de tous les livres | |
|---|---|
| Requête réalisée | Résultat retourné |
| GET api.pucesin.com/livres<br>Headers : - Cookie : none | Liste de tous les livres au format JSON |

| Test n° 2 : test permettant d'obtenir tous les colis pour tous les utilisateurs | |
|---|---|
| Requête réalisée | Résultat retourné |
| GET api.pucesin.com/colis<br>Headers : - Cookie : none | Caractéristiques de tous les colis pour tous les utilisateurs (caractéristiques : nom, prénom, ville, adresse électronique) au format JSON |

| Test n° 3 : test permettant d'obtenir les colis de l'utilisateur user1 | |
|---|---|
| Requêtes réalisées | Résultats retournés |
| POST api.pucesin.com/auth<br>Body :{ user: user1,<br>password:154#2Afvc115} | 200 - Ok<br>Headers : Set-Cookie:<jeton d'authentification> |
| GET api.pucesin.com/colis/user1<br>Headers : Cookie :<jeton d'authentification> | Liste des colis de l'utilisateur user1 |

| Test n°4 | |
|---|---|
| Requêtes réalisées | Dernier résultat retourné |
| 10 000 requêtes GET api.pucesin.com/livres en l'espace d'une minute | 500 - Server Error |

## ***Document A6* : Composition d'un jeton (token) JWT**

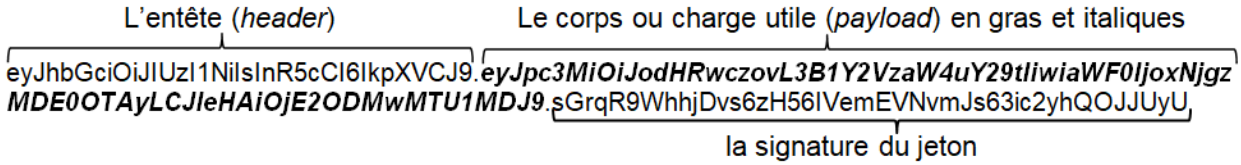
JWT signifie *JSON Web Token* : c'est un standard ouvert (RFC 7519) qui décrit une méthode pour transmettre une information de manière sécurisée. Les jetons JWT sont utilisés comme jetons d'authentification pour consommer un service. Ils sont généralement placés dans l'entête d'un échange HTTP du type autorisation de porteur (*Bearer)*.

Un jeton JWT est composé de trois parties séparées par un point « . » :

- L'**entête** (*header*) contient des informations sur le jeton (type de jeton et algorithme utilisé pour la signature) au format JSON (*JavaScript Objet Notation*) encodées en base 64.
- Le corps appelé également **charge utile** (*payload*) au format JSON encodé en base 64 contient des informations utiles au serveur utilisant le jeton. Les informations sont fournies sous la forme "nomDePropriété" : valeur. Le standard RFC 7519 définit des noms de propriété (*claims*) mais il est possible de définir ses propres noms de propriété pour répondre aux besoins de l'application. Voici des exemples de propriétés :
  - **iss** (*Issuer* = émetteur du jeton) : permet d'identifier le serveur qui a émis le jeton ;
  - **iat** (*Issued At* = date de génération du jeton) : date au format *timestamp* (exprimée en nombre de secondes écoulées depuis le 01 janvier 1970) ;
  - **exp** (*Expiration Time* = date d'expiration du jeton) : date au format *timestamp.*
- La signature du jeton est un condensé (*hash*) obtenu en appliquant une fonction de hachage sur l'entête (*header*) concaténé à la charge utile (*payload*). Cette fonction de hachage utilise une clé secrète. La signature d'un jeton permet de s'assurer de la validité de celui-ci. De ce fait, si un utilisateur malveillant modifie le contenu du jeton, la signature ne sera plus correcte et le jeton sera ainsi rejeté.

Rappel : ces 3 parties sont séparées par un point.

Voici un exemple de jeton JWT transmis lors d'un échange HTTP :




- L'entête (*header*) est **eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9**. Si nous le décodons, nous obtenons :

```
{
    "typ":"JWT",
    "alg":"HS256"
 }
```

- La charge utile (*payload*) est la partie en gras et en italique commençant par ***eyJpc3Mi*** et finissant par ***MTU1MDJ9***. Si nous la décodons nous obtenons :

```
{
    "iss":"https://pucesin.com",
    "iat":1521309600,
    "exp":1521313200
}
```

- La signature est obtenue en utilisant l'algorithme HS256 (HMAC-SHA-256), la signature sera créée de cette manière :

  Signature = HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload), clé)
  clé contient la clé secrète utilisée par le serveur pour générer la signature du jeton.

***Document A7 : Tests de l'interface de programmation d'application (API) réalisés suite à l'intégration des jetons (tokens) JWT***

Test n°1 réalisé le 02/05/2024 à 10:08:20 :

1) Réalisation d'une requête POST api.pucesin.com/auth permettant de se connecter pour obtenir un jeton d'authentification JWT.
2) Réalisation d'une requête GET api.pucesin.com/colis/user1 en transmettant le jeton obtenu.

Contenu du jeton JWT encodé

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
Jpc3MiOiJodHRwczovL3B1Y2VzaW4uY29tIiwia
WF0IjoiMTcxNDYzNzMwMiIsImV4cCI6IjE3MTQ2
Mzc5MDIifQ.b7k6CZODzYew3e9VRxsvET6Hrj_j
GQgi6Auu3d9RMA0
```

Contenu du jeton JWT décodé

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "iss": "https://pucesin.com",
  "iat": "1714637302",
  "exp": "1714637902"
}
```

| Date au format *timestamp* | Date au format jj/mm/aaaa hh:mm:ss |
|---|---|
| 1714637302 | 02/05/2024 10:08:22 |
| 1714637902 | 02/05/2024 10:18:22 |
| 1714641502 | 02/05/2024 11:18:22 |

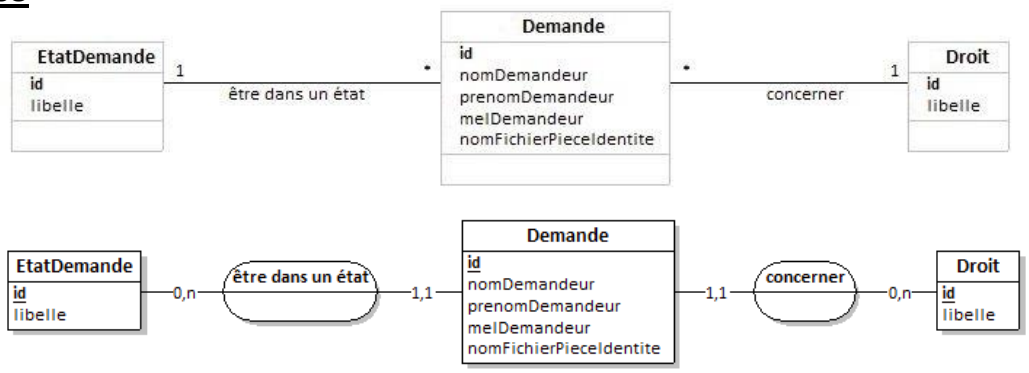
= 1714637302 + 600 secondes (10min)

Test n°2 réalisé le 02/05/2024 à 10:30:00 :

1) Réalisation d'une requête GET api.pucesin.com/colis/user1 en transmettant le jeton obtenu lors du test n°1. L'erreur *ExpiredJwtException* est retournée.

**Document associé au dossier B**

***Document B1 : Diagramme de classes et schéma entité-association de la base de données***




Remarque : actuellement, seul l'état courant de la demande est géré, l'identifiant peut prendre les valeurs : 1 pour en attente de traitement, 2 pour en cours de traitement, 3 pour traitée.

**Documents associés au dossier C**

### *Document C1 : Ticket S-C-1092*

**S-C-1092 : Bug après redirection mail**

Joindre | Ajouter un ticket enfant | Associer un ticket | •••

Bonjour,

Ce matin j'ai reçu un mail d'un client m'indiquant qu'il avait un problème sur une commande de livres. Dans son courriel il m'invitait à cliquer sur un lien pour visualiser son problème, ce que j'ai fait. Je n'ai pas réussi à visualiser quoi que ce soit concernant son problème, j'ai simplement obtenu un message indiquant «demande de suppression prise en compte» ce qui m'a un peu surpris.

Depuis je n'arrive plus à me connecter. Pouvez-vous m'aider car je suis très inquiet étant donné que je suis habilité à gérer les utilisateurs de l'application ?

Je ne sais pas si cela peut vous aider mais je viens de consulter l'URL associée au lien sur lequel j'ai cliqué et celle-ci contient https://www.pucesin.com/admin/deleteCompte.php?id=123

Merci de bien vouloir prendre en compte ma demande au plus vite.

Cordialement
Etienne Klein

### *Document C2 : Ticket S-A-0095*

**S-A-0095 : Sites similaires**

Joindre | Ajouter un ticket enfant | Associer un ticket | •••

Bonjour,

En saisissant le nom de notre site dans un moteur de recherche, j'ai obtenu d'autres sites que le nôtre dans les premiers résultats : pucesin.fr, puceslin.org, puces.com.

Notre site apparaît mais bien après ces sites dans ce moteur de recherche.

Ces sites copient notre interface graphique et même notre logo et sont bien mieux classés que nous. C'est une perte financière potentielle très importante pour nous, sans parler de l'image renvoyée à nos clients.

Est-il possible d'intervenir ?

Cordialement.
Jade CHEN

## ***Document C3 : Programme utilisant les données de la base Whois pour prévenir l'administrateur système avant l'expiration de la propriété du nom de domaine pucesin.com***

```
1.  // renseigne le nom de domaine
2.  $domain = 'pucesin.com';
3.  // exécution de la commande whois pour obtenir les informations du nom de domaine
4.  $info_nom_domaine = shell_exec('whois ' . escapeshellarg($domain));
5.  // Récupération de la date d'expiration du nom de domaine
6.  preg_match('/Expiry Date:(.*?)\n/', $info_nom_domaine, $matches);
7.  $date_expiration_complete = trim($matches[1]);
8.  $date_expiration= date('Y-m-d', strtotime($date_expiration_complete));
9.  // Récupération de la date du jour
10. $date_jour = date('Y-m-d');
11. // contrôle de la date d'expiration
12. $date_dans_3_mois = date('Y-m-d', strtotime('+3 month', strtotime($date_jour)));
13. if ($date_expiration == $date_dans_3_mois) {
14.     $message = "Le nom de domaine ".$domain." expirera dans 3 mois";
15. } else {
16.     $date_dans_1_mois = date('Y-m-d', strtotime('+1 month', strtotime($date_jour)));
17.     if ($date_expiration == $date_dans_1_mois) {
18.         $message = "Le nom de domaine ".$domain." expirera dans 1 mois";
19.     }
20. }
21. if (isset($message)) {
22.     envoyer_mail("admin@pucesin.com","Nom de domaine ".$domain, $message);
23. }
```

La fonction *envoyer_mail* permet l'envoi d'un mail. L'entête de cette fonction est :

*envoyer_mail(string $destinataire, string $sujet, string $corps)*

Remarque : pour ajouter une année à une date on indiquera '+1 year', pour enlever 2 jours à une date on écrira '- 2 days'.

## ***Document C4 : Extrait du fichier auth.log des traces (logs) d'authentification réussies ou non du mois de mars 2024***

Site-pucesin sshd[7683] : Invalid user livre from 51.87.35.150 port 44142
Site-pucesin sshd[7683] : Failed password for invalid user livre from 51.87.35.150 port 44142 ssh2
Site-pucesin sshd[7687] : Failed password for admsys from 51.87.35.150 port 44146 ssh2
Site-pucesin sshd[7687] : Failed password for admsys from 51.87.35.150 port 44146 ssh2
Site-pucesin sshd[7687] : Failed password for admsys from 51.87.35.150 port 44146 ssh2
Site-pucesin sshd[7687] : Failed password for admsys from 51.87.35.150 port 44146 ssh2
Site-pucesin sshd[7687] : Failed password for admsys from 51.87.35.150 port 44146 ssh2
Site-pucesin sshd[7687] : Failed password for admsys from 51.87.35.150 port 44146 ssh2
Site-pucesin sshd[7687] : Failed password for admsys from 51.87.35.150 port 44146 ssh2
Site-pucesin sshd[7687] : Accepted password for admsys from 51.87.35.150 port 44146 ssh2
Site-pucesin systemd-logind[679]: New session 30 of user admsys

**admsys** est le compte de l'administrateur du serveur *pucesin.com* et **dispose de tous les droits**.

## ***Document C5 : Extrait du fichier des traces (logs) d'audit de MySQL du mois d'avril 2024***

| event_time | user_host | command_type | convert(argument using utf8) |
|---|---|---|---|
| 2024-04-16 14:55:41.114633 | admsys [root] @localhost [ ] | Query | CREATE USER 'lecteur'@'dwln-app.to' identified by <secret> |
| 2024-04-16 14:55:47.502300 | admsys [root] @localhost [ ] | Query | GRANT ALL ON *.* TO 'lecteur'@'dwln-app.to' |