

BTS SIO
Bloc 3 – SLAM

ENTRAÎNEMENT À L'ÉPREUVE E7

Étude de cas type

Amal Hecker

Les cours du Cned sont strictement réservés à l'usage privé de leurs destinataires et ne sont pas destinés à une utilisation collective. Les personnes qui s'en serviraient pour d'autres usages, qui en feraient une reproduction intégrale ou partielle, une traduction sans le consentement du Cned, s'exposeraient à des poursuites judiciaires et aux sanctions pénales prévues par le Code de la propriété intellectuelle. Les reproductions par reprographie de livres et de périodiques protégés contenues dans cet ouvrage sont effectuées par le Cned avec l'autorisation du Centre français d'exploitation du droit de copie (20, rue des Grands-Augustins, 75006 Paris).

Cned, BP 60200, 86980 Futuroscope Chasseneuil Cedex, France

Le Cned remercie les personnes qui ont contribué à la réalisation de ce document. Qu'elles trouvent ici l'expression de toute sa reconnaissance.

© Cned 2026

87DE6DVWB0125

ÉTUDE DE CAS TYPE



Durée conseillée : 4 heures

» Barème indicatif (notation sur 100, note finale sur 20) :

- **Dossier A** – Conformité au RGPD de l'application de gestion des réservations : **36 points**
- **Dossier B** – Sécurisation de l'application de gestion des guides : **26 points**
- **Dossier C** – Sécurisation de l'authentification des acheteurs : **30 points**
- **Dossier D** – Sécurisation des environnements de développement et de déploiement : **8 points**

DOSSIER DOCUMENTAIRE

DOCUMENTS COMMUNS AU PROJET

DOCUMENT 1	Vocabulaire de la méthode agile Scrum.....	page 12
DOCUMENT 2	Récit utilisateur (user story) n°1 « Achat des billets en ligne »	page 12
DOCUMENT 3	Représentations conceptuelles de la base de données.....	page 13

Documents associés au dossier A page 14

Document A1	Règlement général sur la protection des données – Article 5.1	page 14
Document A2	Règles d'archivage et de nettoyage des données personnelles des acheteurs	page 15
Document A3	Script SQL de nettoyage de la base de données active et d'archivage.....	page 15
Document A4	Extrait du schéma relationnel de la base de données active lascauxprod	page 16

Documents associés au dossier B page 18

Document B1	Scénarios de risque (abuser story ou evil user story)	page 18
Document B2	Syntaxe MySQL (extrait du manuel de référence de MySQL 8.0)	page 18
Document B3	Code partiel de l'application de gestion des visites.....	page 19
Document B4	Exemple de test unitaire	page 21
Document B5	Diagramme de classes partiel de la planification des visites.....	page 21
Document B6	Expressions régulières	page 21

Documents associés au dossier C page 22

Document C1	Codes partiels de l'enregistrement d'un acheteur	page 22
Document C2	Fonction de hachage – Extraits de la documentation PHP	page 23
Document C3	Erreur d'enregistrement d'un nouvel acheteur	page 23
Document C4	Politique de sécurité des mots de passe – Extrait du guide de l'ANSSI	page 23
Document C5	Exemple d'utilisation d'un tableau	page 24
Document C6	Exemple de création d'une requête préparée	page 24
Document C7	Manipulation de chaînes de caractères en PHP	page 24

Documents associés au dossier D page 25

Document D1	Architecture technique des applications de Lascaux IV	page 25
-------------	---	---------



Présentation du contexte

La grotte de Lascaux, située à Montignac en Dordogne, est une grotte ornée du paléolithique, découverte en 1940. Elle fait partie du patrimoine mondial de l'Unesco depuis 1979. Très vite, les scientifiques ont constaté que les visites humaines détérioraient les peintures, ce qui a conduit à fermer la grotte au public en 1963.

Le Ciap, appelé également Lascaux IV, offre au public une réplique de la grotte de Lascaux en permettant aux visiteurs d'utiliser les technologies d'aujourd'hui pour personnaliser leur visite avant, pendant et après. Cette réplique de la grotte comporte une partie musée où le visiteur est en interaction permanente avec ce qu'il côtoie, grâce à l'utilisation d'une tablette appelée compagnon de visite (CDV). Mille six cents CDV en dix langues sont mis à la disposition des visiteurs.

L'organisation cliente

Le Ciap dispose de plusieurs systèmes d'information (SI) internes, à savoir le SI bâtiminaire, le SI administratif (ressources humaines, comptabilité, etc.), le SI visiteur et le SI scénographie (écrans, vidéoprojections, lumières, etc.). L'articulation de ces SI internes avec la billetterie en ligne, le site internet, l'application mégadonnées (*big data*) permet au Ciap la gestion de ce qu'il appelle l'expérience utilisateur.

Il peut y avoir cinq mille visiteurs par jour. Grâce à l'application tablette pour les CDV, le site internet, la plateforme d'échanges (MyLascaux), la solution mégadonnées (*big data*), il s'agit d'assurer un service très complet de gestion de la relation client avant, pendant et après la visite. Sur la partie après visite, plus d'un million de visiteurs peuvent se connecter.

La direction des systèmes d'information (DSI) du conseil départemental a lancé des appels d'offres dont un pour les outils numériques et multimédias et un pour les applications informatiques. Quatre entreprises de services numériques (ESN) ont obtenu des lots de ces deux appels d'offres.

L'entreprise prestataire de services

L'ESN Aquilasc, éditeur de logiciels de gestion sur mesure de type client lourd, *Web* ou mobile et spécialisée dans le secteur du tourisme, a été retenue pour deux lots.

Le premier appel d'offres concerne l'application *Web* de la gestion des réservations de billets et de l'organisation des visites. Le deuxième appel d'offres porte sur la maintenance et la sécurisation de l'application client lourd permettant la gestion des visites.

Depuis quatre ans, Aquilasc utilise la méthode agile *Scrum* pour la gestion de ses projets. Cette méthode permet d'impliquer l'organisation cliente durant la durée totale du projet et de lui livrer de manière régulière de nouvelles versions de l'application, avec leur lot de corrections et de nouvelles fonctionnalités, apportant ainsi de plus en plus de valeur métier à l'application.

Un contrat de prestation de services a été établi entre le Ciap et Aquilasc. Il définit la nature des interventions de l'ESN, leurs durées.

Vous faites partie de l'équipe *Scrum*, votre mission consiste à participer au développement des applications *Web* et client lourd.

En mode agile, vous participerez à l'atelier analyse de risques des applications, la sécurisation des données et à la gestion de l'authentification des utilisateurs internes et externes. Vous vous appuierez sur les dossiers documentaires mis à votre disposition.

DOSSIER A CONFORMITÉ RGPD DE L'APPLICATION DE GESTION DES RÉSERVATIONS EN LIGNE

» Documents communs à utiliser : 1, 2, 3

» Documents propres au dossier A : A1, A2, A3, A4

Soucieuse de son image numérique, Lascaux IV réalise, dans le cadre de ses développements d'applications métier, des revues d'itération (*sprint view*) dédiées à la vérification de la conformité RGPD des données manipulées par les applications. Les développeurs et les concepteurs de bases de données intègrent ainsi, dès la conception, les recommandations du RGPD (Règlement général sur la protection des données) concernant les données à caractère personnel dans le respect de la vie privée des utilisateurs (*privacy by design*).

Votre équipe a la charge de concevoir la base de données pour l'application de gestion des achats de billets en ligne en cours de développement (récit utilisateur n°1).

MISSION A1 Mesures préventives pour assurer la sécurité et la conformité RGPD des données de l'application de réservation de billets en ligne

Vous présentez une 1^{re} ébauche de votre modélisation de la base de données lors du sprint 1. Martine Depeo, déléguée à la protection des données, attire particulièrement l'attention de votre équipe sur le principe de limitation de la durée de conservation des données personnelles des acheteurs, notamment celles liées au paiement. Ce principe impose à l'organisme de mettre en place différentes mesures pour organiser la gestion du cycle de vie des données personnelles traitées, et les durées de conservation associées.



Question A1.1

Expliquer en quoi les obligations de conformité et de sécurité des données personnelles et bancaires des acheteurs sont un enjeu pour Lascaux IV.

Martine Depeo recommande de prévoir un script qui s'exécutera de manière automatique dans la base de données pour la nettoyer selon le règlement général sur la protection des données (RGPD).



Question A1.2

Expliquer l'intérêt d'avoir un script de nettoyage et d'archivage au niveau de la base de données et non au niveau applicatif.

La responsable de traitement et la DPO vous fournissent les règles d'archivage et de nettoyage des données personnelles des acheteurs pour respecter l'article 5.1.e du RGPD.

L'administrateur de la base de données a créé une base de données d'archive intermédiaire. Elle contient la table AcheteurArchive qui va servir à archiver les données des acheteurs inactifs.



Question A1.3

- Justifier la nécessité de l'archivage des données des acheteurs inactifs.
- Compléter le script fourni avec les requêtes SQL permettant de traiter les règles d'archivage et de nettoyage des données personnelles des acheteurs. Vous reporterez sur votre copie le numéro de la ligne concernée par la règle à traiter.

MISSION A2 Adaptation de la représentation conceptuelle de la base de données pour sécuriser les données et se conformer au RGPD



Important

Vous pouvez retenir le formalisme de votre choix (schéma entité-association, diagramme de classes) pour représenter les évolutions conceptuelles demandées.

La 2^e itération consiste à faire évoluer la représentation conceptuelle de la base de données pour une meilleure conformité au RGPD et pour respecter la nouvelle politique de sécurité et de conservation des données personnelles et bancaires manipulées par le récit utilisateur n°1.

Lorsqu'un acheteur effectue une réservation en ligne, les données relatives à la carte ayant servi au paiement seront conservées, de manière chiffrée, dans la base de données du prestataire de paiement certifié PCI-DSS (*Payment Card Industry Data Security Standard*). L'acheteur peut enregistrer ou non ces informations de paiement dans la base de données locale de Lascaux. S'il accepte l'enregistrement de sa carte bancaire pour ses achats ultérieurs, le numéro et la date d'expiration de la carte seront stockés de manière chiffrée. Pour permettre à l'acheteur de reconnaître la carte qui a servi à sa dernière transaction, les quatre derniers chiffres du numéro de la carte seront sauvegardés en clair avant le chiffrement de cette dernière. La preuve du consentement doit être conservée afin de pouvoir la produire à tout moment. Pour répondre à cet objectif de la CNIL, on enregistrera l'URL, la date et l'heure du consentement ainsi que l'adresse IP de l'acheteur.

Les acheteurs peuvent enregistrer plusieurs cartes. Le consentement doit être demandé pour chacune d'elles. Pour garantir la conformité au RGPD, un acheteur peut retirer son consentement à tout moment depuis son onglet Paiements dans son espace. La date et l'heure de retrait du consentement de la carte concernée doivent être sauvegardées.

Un paiement correspond à une réservation. Il est caractérisé par un identifiant unique, le montant de la transaction et le moyen utilisé pour le règlement (CB, crypto-monnaie). En effet, les acheteurs particuliers peuvent payer avec des monnaies électroniques (crypto-monnaies). Ils pourront ainsi effectuer rapidement des transactions sécurisées et réduire le risque de fraude. Dans le cas d'un paiement avec une crypto-monnaie, un code QR de la transaction est généré. Il permet d'enregistrer automatiquement l'adresse correcte d'envoi du montant à régler. Le paiement en crypto-monnaie doit être rapide à cause de la volatilité du cours. Si le délai est dépassé, l'état du paiement apparaîtra comme « expiré ». Lascaux peut alors l'accepter ou le refuser. Dans le cas où le paiement est « refusé », Lascaux devra rembourser le montant réglé à l'acheteur particulier.

La nouvelle base de données doit respecter le principe de minimisation des données (article 5.1.c). Les données à caractère personnel conservées lors de l'itération précédente qui ne sont pas nécessaires au traitement doivent être retirées de la nouvelle base de données. L'acheteur doit pouvoir saisir le nombre de compagnons de visite (CDV) numériques éventuellement nécessaires lors de la visite. Il existe plusieurs types de CDV : CDV avec la taille de la police de caractères paramétrable, CDV avec volume sonore élevé, CDV avec clavier braille.



Question A2.1

Identifier les données personnelles qui ne respectent pas l'article 5.1.c du RGPD. Justifier votre réponse.



Question A2.2

Proposer les modifications de la base de données pour prendre en compte les évolutions de la nouvelle politique de sécurité des données et de paiement tout en respectant le principe de minimisation des données du RGPD.

DOSSIER B SÉCURISATION DE L'APPLICATION CLIENT LOURD DE GESTION DES GUIDES

» Documents propres au dossier B : B1, B2, B3, B4, B5, B6



Important

Vous pouvez choisir de présenter les éléments de code à l'aide du langage de programmation de votre choix.

La planification des visites est une application de bureau développée par un prestataire externe il y a quelques années. Cette application regroupe deux fonctionnalités principales :

- **Gestion des guides** : enregistrement, mise à jour d'informations personnelles et des langues parlées avec leur niveau de maîtrise, visualisation des plannings des visites ;
- **Planification des visites (réservée aux guides gestionnaires)** : création des visites et affectation des guides visiteurs selon leur disponibilité et leur niveau de maîtrise de la langue réservée.

Lascaux a des guides permanents et des guides vacataires sollicités en cas de forte affluence. Les informations des guides permanents sont récupérées depuis l'annuaire de Lascaux. Les guides vacataires sont recrutés après un entretien et des tests d'évaluation de leurs niveaux de langues. Si les tests sont concluants, ils saisissent, via l'application, leurs informations personnelles. La saisie de leurs niveaux de langues est effectuée par les gestionnaires pour éviter les tricheries.

Depuis la mise à jour de l'application, les guides se plaignent d'avoir des informations erronées dans la base de données.

MISSION B1 Participation à l'analyse et à l'évaluation des risques

Vous participez à l'identification et à l'estimation des menaces potentiellement réalisables dans le cadre de l'application en l'état. Une ébauche de cette analyse vous a été transmise.



Question B1.1

- Expliquer l'impact de l'événement redouté 1.
- Préciser, sur votre copie, la probabilité, la gravité et les contre-mesures de l'événement 4.
- Quel est le critère de sécurité DICP (*Disponibilité-Intégrité-Confidentialité-Preuve*) non respecté dans l'événement 3 ?

Des guides arrivent à s'authentifier avec un mot de passe même erroné (événement redouté n°1).



Question B1.2

Mettre en place la contre-mesure n°1 de l'événement redouté n°1. *Reporter uniquement les modifications nécessaires sur votre copie.*

Il y a des erreurs au niveau de la saisie des adresses mails lors de l'enregistrement des guides vacataires. Votre collègue a développé une méthode pour les vérifier. Vous devez la tester.



Question B1.3

Écrire la méthode de test unitaire de la méthode `validerEmail()` avec trois jeux de données de test pertinentes.

Vous participez à la mise en place d'une politique plus granulaire des accès des guides à la base de données. La réduction des privilèges au minimum nécessite la mise en place de deux rôles :

- « `gest_guide` » : regroupera les privilèges CRUD (Create-Read-Update-Delete) des guides gestionnaires sur les tables Visite, Guide, Langue et Parler depuis le réseau 192.168.30.0/24 uniquement. Aucun droit ne leur sera attribué sur les autres tables ;
- « `visit_guide` » : regroupera le privilège de consultation des tables Guide et Visite depuis le réseau 192.168.30.0/24 aux guides chargés des visites.



Question B1.4

- a. Écrire la requête SQL permettant de créer le rôle gest_guide.
- b. Écrire les requêtes SQL permettant d'affecter les privilèges CRUD au rôle gest_guide sur la table Visite.

MISSION B2 Surveillance des accès à la base de données active

Le responsable sécurité souhaite surveiller les connexions et les opérations effectuées sur la base de données depuis l'application de gestion des visites. Il vous demande de développer un système d'horodatage de ces opérations en journalisant : la date et l'heure de la requête, l'hôte depuis lequel elle a été envoyée, la requête qui a été effectuée et l'adresse mail du guide qui l'a initiée.

Une table log a été créée dans la base de données log_guides sur le serveur log.lascaux.org afin de stocker les informations souhaitées.

```
log(id, dateHeureLog, hote, requete, mel)
```

id : clé primaire



Question B2.1

Écrire le code permettant de journaliser les opérations effectuées sur la base de données.

Les contre-mesures des événements redoutés sont mises en place. La nouvelle application est redéployée sur les postes. Le guide gestionnaire Jean Vilar a détecté la présence d'enregistrements de visites créées par des guides vacataires alors qu'ils ne sont pas autorisés à le faire (événement redouté n°2).

L'analyse des logs a confirmé cet événement et a révélé un problème au niveau de l'enregistrement d'un guide vacataire.



Question B2.2

Apporter les modifications nécessaires afin que le guide vacataire soit bien identifié dans la base de données et éviter une confusion avec le guide gestionnaire.

DOSSIER C SÉCURISATION DE L'AUTHENTIFICATION DES ACHETEURS

» Documents propres au dossier C : C1, C2, C3, C4, C5, C6, C7



Important

Vous pouvez choisir de présenter les éléments de code à l'aide du langage de programmation de votre choix.

Le développement en cours de l'application de réservation des billets en ligne est réalisé en adoptant la technique de la programmation pilotée par les tests. Vous êtes tour à tour développeur ou testeur (test de sûreté et de sécurité).

MISSION C1 Confidentialité des éléments d'authentification des acheteurs

Votre collègue a développé le formulaire d'enregistrement des acheteurs que vous devez tester.

Roger Zanches vous fait part de l'étude de Verizon de 2021 selon laquelle 81 % des notifications de violations de données mondiales seraient liés à une problématique de mots de passe et que 60 % des notifications reçues par la CNIL depuis le début de l'année 2021 sont liés à du piratage et un grand nombre aurait pu être évité par le respect de bonnes pratiques en matière de mots de passe.

Il vous demande de tout mettre en œuvre pour vérifier la protection des mots de passe des acheteurs enregistrés.

Le test d'intrusion que vous avez effectué avec l'outil John the Ripper sur le jeu de données enregistré par votre binôme a révélé une faille de sécurité puisque certains mots de passe ont été déchiffrés par l'outil.



Question C1.1

Expliquer à votre collègue les risques encourus par Lascaux à cause de cette faille et identifier les besoins de sécurité impactés.

Afin de garantir la sécurité des données des acheteurs et des visiteurs contre les attaques par **force brute**, vous utilisez une fonction de hachage réputée être la plus robuste à l'état de l'art pour chiffrer les mots de passe des acheteurs.



Question C1.2

Réécrire la méthode `hacherMotDePasse()` de la classe `Securite`.

Suite à la réécriture de la méthode `hacherMotDePasse()`, l'inscription d'un nouvel acheteur provoque une erreur au niveau de la base de données.



Question C1.3

Écrire la requête SQL permettant de corriger l'incident constaté.

Afin d'homogénéiser les pratiques, Roger Zanches souhaite que l'algorithme de chiffrement utilisé pour les mots de passe des acheteurs puisse être utilisé pour d'autres mots de passe.

Il vous demande de développer une méthode qui permettra de le faire.



Question C1.4

Écrire la méthode `hacherListeMotsDePasse($listeMdp)` dans la classe `Securite`.

MISSION C2 Confidentialité des éléments d'authentification des acheteurs

Christine Berton, la responsable de produit, a testé l'enregistrement d'un acheteur. Lors de la revue de sprint, elle vous fait part des problèmes détectés.

- **Problème 1** : un acheteur peut s'enregistrer en définissant un mot de passe sur 1 caractère.
- **Problème 2** : un acheteur ne peut pas saisir un mot de passe supérieur à 15 caractères.
- **Problème 3** : un acheteur peut s'enregistrer plusieurs fois avec la même adresse mail.
- **Problème 4** : un acheteur peut saisir des adresses de courrier non valides ou non existantes.

La DSI de Lascaux définit le niveau de sensibilité du mot de passe de « fort à très fort » de l'échelle de référence du guide des recommandations de l'ANSSI relatives à la gestion des mots de passe (R21 et R22). La longueur minimale doit être respectée. En outre, le mot de passe doit contenir au moins une majuscule et un chiffre. Les mots de passe des acheteurs ne doivent pas déroger à cette règle.



Question C2.1

- Identifier la fonction concernée par les problèmes numéro 1 et 2 dans la classe *ValidationForm*.
- Modifier la méthode pour respecter les contraintes sur les mots de passe définies par la DSI.

Les problèmes 3 et 4 peuvent avoir des conséquences fâcheuses en terme de communication si la vulnérabilité est détectée par un attaquant externe ou un acheteur peu scrupuleux. Vous êtes en charge de la mise en œuvre des contre-mesures pour empêcher le problème 3 de se produire lors de l'enregistrement d'un acheteur. L'adresse mail fournie sera vérifiée et si elle existe, l'enregistrement n'aboutit pas.



Question C2.2

- Présenter les conséquences des problèmes 3 et 4 si les adresses mails non valides ou en doublon sont utilisées dans les lettres de diffusion de Lascaux.
- Écrire la méthode qui empêcherait le problème 3 de se produire. Justifier l'emplacement (classe) de cette méthode.

DOSSIER D CHOIX ET PROTECTION DES ENVIRONNEMENTS DE DÉVELOPPEMENT

» Document propre au dossier D : D1

L'itération zéro (sprint 0) prépare le développement en précisant, pour toute l'équipe, les points suivants : les objectifs du client, le périmètre de l'application, les contraintes, les intervenants dans le projet, les utilisateurs finaux, la modélisation du domaine métier, l'architecture technique, le mode de travail sur le projet, le budget, le planning global, la gestion des sous-traitants, etc.

MISSION D1 Besoins de sécurité dans l'environnement de développement

Michaël Vaulthon, intégrateur des applications métiers de Lascaux, vous présente l'architecture technique des environnements de développement, de tests et de production.



Question D1.1

- Justifier les raisons d'isoler les environnements de développement et de production par des VLAN (*Virtual Local Area Network*).
- Pour chaque critère fondamental de la sécurité (confidentialité, intégrité, disponibilité et traçabilité), désigner les éléments de l'architecture qui le garantissent.

La disponibilité des données et la traçabilité des accès sont des éléments qui contribuent à la sécurité des applications de réservation de billets en ligne et de l'organisation des visites.



Question D1.2

Expliquer pourquoi l'architecture technique actuelle présente des faiblesses au regard de ces deux éléments.



DOCUMENT 1 Vocabulaire de la méthode agile Scrum

Le développement du projet est réalisé en adoptant la méthode agile Scrum. Chaque récit utilisateur¹ fait l'objet de plusieurs itérations². La fonctionnalité développée durant une itération est testée à la fois sur le plan de la qualité logicielle, de la sûreté et de la sécurité du code pour minimiser les vulnérabilités et contrer les attaques. Les récits utilisateurs peuvent être accompagnés par des scénarios de risque³. À la fin de chaque itération, la fonctionnalité est testée par les utilisateurs pour valider la fonctionnalité et/ou détecter des dysfonctionnements au plus tôt du déroulement du projet.

Sources : wikipédia et guide ANSSI cité ci-dessous.

DOCUMENT 2 Récit utilisateur (user story) n° 1 « Achat des billets en ligne »

Titre : achat des billets en ligne **Valeur Métier :** 15

Objectif : en tant qu'acheteur, je veux acheter en ligne les billets pour plusieurs personnes afin qu'elles participent à une visite guidée.

Rattachement : itération (*sprint*) n°1

Critères d'acceptation :

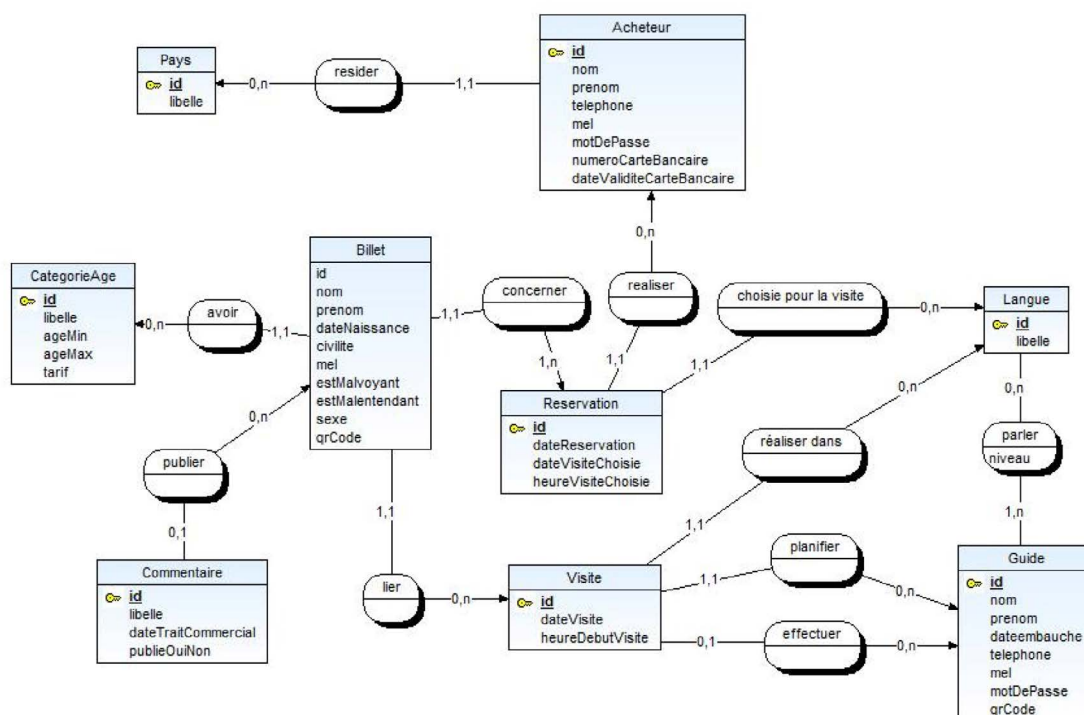
- pour réserver des billets sur internet, une personne (appelée acheteur) devra créer un compte en fournissant son adresse mél qui sera utilisée comme identifiant, et un mot de passe. Elle doit fournir également son consentement pour l'inscription à une newsletter ;
- une fois connecté, l'acheteur obtiendra un premier formulaire lui proposant de renseigner les informations utiles à la réservation de billets : son nom, son prénom, son mél, son numéro de téléphone. Puis, il devra choisir la date et l'heure souhaitées pour la visite, la langue dans laquelle il souhaite que la visite se déroule et le nombre de billets désirés ;
- ensuite, pour chacun des billets qu'il souhaite réserver, un deuxième formulaire de saisie lui demandera de fournir, pour chaque personne destinataire d'un billet : son nom, son prénom, sa civilité, son sexe, son adresse mél, son handicap (malvoyante ou malentendante) et sa date de naissance. La date de naissance est utilisée uniquement pour déterminer la tranche d'âge de la personne ; elle est déterminante pour connaître le tarif du billet. La connaissance du/des handicaps (malentendance, malvoyance) du visiteur permettra de lui préparer un compagnon de visite adapté.

Suite à la validation des données saisies, l'acheteur est invité à saisir ses données bancaires. Un paiement sécurisé est effectué.

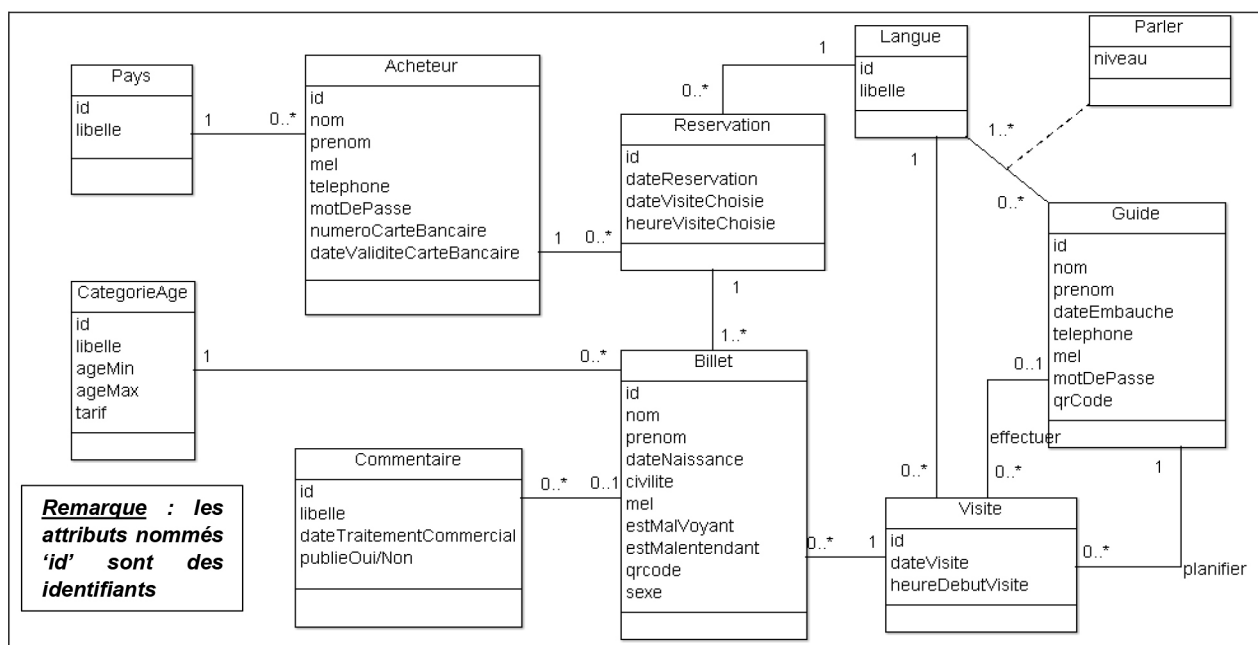
1. Récit utilisateur (*user story*) : fonctionnalité à réaliser, exprimée sous la forme d'un scénario utilisateur.
2. Itération (*sprint*) : intervalle de temps court, généralement de 2 à 4 semaines, pendant lequel on doit concevoir, réaliser et tester un ensemble de fonctionnalités. Conventionnellement, l'itération zéro est l'itération de préparation.
3. Scénario de risque (*abuser story ou evil user story*) : le scénario de risque est une variante utilisée pour intégrer la sécurité dès le début des développements. Ils peuvent être nommés « *abuser stories* » car ils correspondent au revers néfaste d'un récit utilisateur (*user story*) et engendrent une perte de valeur. Ce type de récit présente les intentions d'un utilisateur malveillant que l'on cherchera à tenir en échec.

DOCUMENT 3 Représentations conceptuelles de la base de données

Représentation conceptuelle de la base de données avec un schéma entité-association



Représentation conceptuelle de la base de données avec un diagramme de classes



Documents associés au dossier A

Document A1 Règlement général sur la protection des données – Article 5.1

Les données à caractère personnel doivent être :

- a. Traitées de manière licite, loyale et transparente au regard de la personne concernée ;
- b. Collectées pour des finalités déterminées, explicites et légitimes, et ne pas être traitées ultérieurement d'une manière incompatible avec ces finalités ;
- c. Adéquates, pertinentes et limitées à ce qui est nécessaire au regard des finalités pour lesquelles elles sont traitées (minimisation des données) ;
- d. Exactes et, si nécessaire, tenues à jour ;
- e. Conservées sous une forme permettant l'identification des personnes concernées pendant une durée n'excédant pas celle nécessaire au regard des finalités pour lesquelles elles sont traitées ; (limitation de la conservation) ;
- f. Traitées de façon à garantir une sécurité appropriée des données à caractère personnel, y compris la protection contre le traitement non autorisé ou illicite et contre la perte, la destruction ou les dégâts d'origine accidentelle, à l'aide de mesures techniques ou organisationnelles appropriées.

Gestion du cycle de vie des données personnelles

Le cycle de vie d'une donnée peut se décomposer en trois phases successives :

- l'utilisation courante dite « **base active** » : cette étape concerne l'utilisation courante des données personnelles par les services chargés de la mise en œuvre de leur traitement ;
- l'archivage en **base intermédiaire** : les données personnelles ne sont plus utilisées pour atteindre l'objectif fixé mais présentent encore un intérêt administratif pour l'organisme ou doivent être conservées pour répondre à une obligation légale. Les données peuvent alors être consultées de manière ponctuelle et motivée par des personnes spécifiquement habilitées ;
- l'archivage définitif : il s'agit des données qui sont archivées sans limitation de durée. Cette phase concerne essentiellement les traitements mis en œuvre à des fins archivistiques dans l'intérêt public (secteur public soumis aux dispositions du livre II du code du patrimoine).

Principe de limitation de la durée de conservation des données de paiement

Durées de conservation des données de paiement

Finalité	Durée de conservation
Paiement unique	Jusqu'au paiement complet. Jusqu'à la réception du bien ou à l'exécution de la prestation de service. Augmenté du délai de rétraction prévu pour les ventes de biens et fournitures de prestations de services à distance.
Gestion des réclamations	13 mois suivant la date de débit ou 15 mois en cas de cartes de paiement à débit différé. Les données sont conservées à des fins de preuve en archive intermédiaire et ne sont consultées qu'en cas de contestation de la transaction.
Faciliter les achats ultérieurs	Jusqu'au retrait du consentement et/ou à l'expiration de la validité des données de la carte bancaire.

Les données strictement nécessaires à la réalisation d'une transaction sont : le numéro de la carte, la date d'expiration, le cryptogramme visuel. Dans tous les cas, la conservation du cryptogramme est interdite après la réalisation de la première transaction.

Document A2 Règles d'archivage et de nettoyage des données personnelles des acheteurs

La durée de conservation des données personnelles (nom, prénom, ...) est précisée dans le registre de traitement. L'application web de réservation de billets en ligne ne nécessite pas d'archivage définitif. Deux bases de données sont créées pour répondre au principe de limitation de la conservation des données pour les acheteurs. Elles sont hébergées sur le même serveur.

- Base « active » : nommée lascauxprod ;
- Base « d'archivage » : nommée lascauxarchive.

Règle 1. Acheteur actif : masquer les informations relatives à ses données bancaires si la date de validité de sa carte de paiement a expiré en réalisant les deux traitements suivants :

1. La date d'expiration sera remplacée par la date du 01 janvier 1970 ;
2. Le numéro de la carte bancaire sera remplacé par la chaîne : xxxx-xxxx-xxxx-xxxx.

Règle 2a. Acheteur inactif (n'a pas effectué de nouvelle réservation depuis un an) : ses données sont archivées dans la table lascauxarchive.AcheteurArchive. La date d'archivage et la date de fin de conservation sont également stockées.

Règle 2b. Les données personnelles archivées de l'acheteur inactif sont nettoyées de la table active.

Règle 3. Acheteur dans la base d'archive : supprimer ses données personnelles quand la durée de conservation est atteinte.

Document A3 Script SQL de nettoyage de la base de données active et d'archivage

Le script suivant permet de nettoyer la base de données active et la base d'archivage selon les règles précisées dans le document A2.

```
1. CREATE PROCEDURE purge_acheteur()
2. BEGIN          -- Déclaration de variables
3.
4. DECLARE fin int DEFAULT 0 ; DECLARE c_id INT ;
5. DECLARE c_nom VARCHAR(100) ;      DECLARE c_prenom VARCHAR(100) ;
6. DECLARE c_mel VARCHAR(100) ;      DECLARE c_tel VARCHAR(100) ;
7. DECLARE c_motDePasse VARCHAR(100) ;
8. DECLARE c_numeroCarteBancaire VARCHAR(50) ;
9. DECLARE c_dateValiditeCarteBancaire DATE ;
10. DECLARE c_idPays INT ;
11. DECLARE c_dateDerniereReservation DATE ;      DECLARE dateFinTraitement DATE ;
12. DECLARE dateFinConservation DATE ;            DECLARE dateAujourd'hui DATE ;
13. -- Déclaration d'un curseur avec les enregistrements des acheteurs
14. DECLARE c_acheteur CURSOR FOR
15. -- Ecrire la requête SQL qui liste les données des acheteurs et la date de leur dernière
   -- réservation. Les colonnes devront respecter l'ordre des colonnes à la ligne 21
   -- À compléter sur votre copie => reporter le numéro 15
16. SELECT now() INTO dateAujourd'hui ; -- Initialiser dateAujourd'hui avec la date courante
17. DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin = 1 ; -- Variable de fin de boucle
18. OPEN c_acheteur ; -- Ouvrir le curseur des acheteurs
19. read_loop : LOOP -- Parcours de la liste des acheteurs
20. -- Récupérer les données de l'enregistrement courant dans des variables
21. FETCH c_acheteur INTO c_id, c_nom, c_prenom, c_mel, c_tel, c_motDePasse,
22.      c_numeroCarteBancaire, c_dateValiditeCarteBancaire, c_idPays,
23.      c_dateDerniereReservation ;
```

```

24. IF fin = 1 THEN          -- S'il n'y a plus d'enregistrements
25.     LEAVE read_loop ;    -- Sortir de la boucle
26. END IF ;
27. -- Calcul de la date de fin de traitement
28. SELECT adddate(c_dateDerniereReservation, INTERVAL 1 YEAR) INTO dateFinTraitement ;

29. -- Bloc d'instructions SQL pour traiter la règle 1 - à rédiger sur votre copie
   -- Reporter le n° de ligne 29

30. -- Bloc d'instructions SQL pour traiter la règle n° 2a, à rédiger sur votre copie
   -- Reporter le n° de ligne 30

31. END LOOP read_loop ;
32. -- Fermer le curseur
33. CLOSE c_acheteur ;

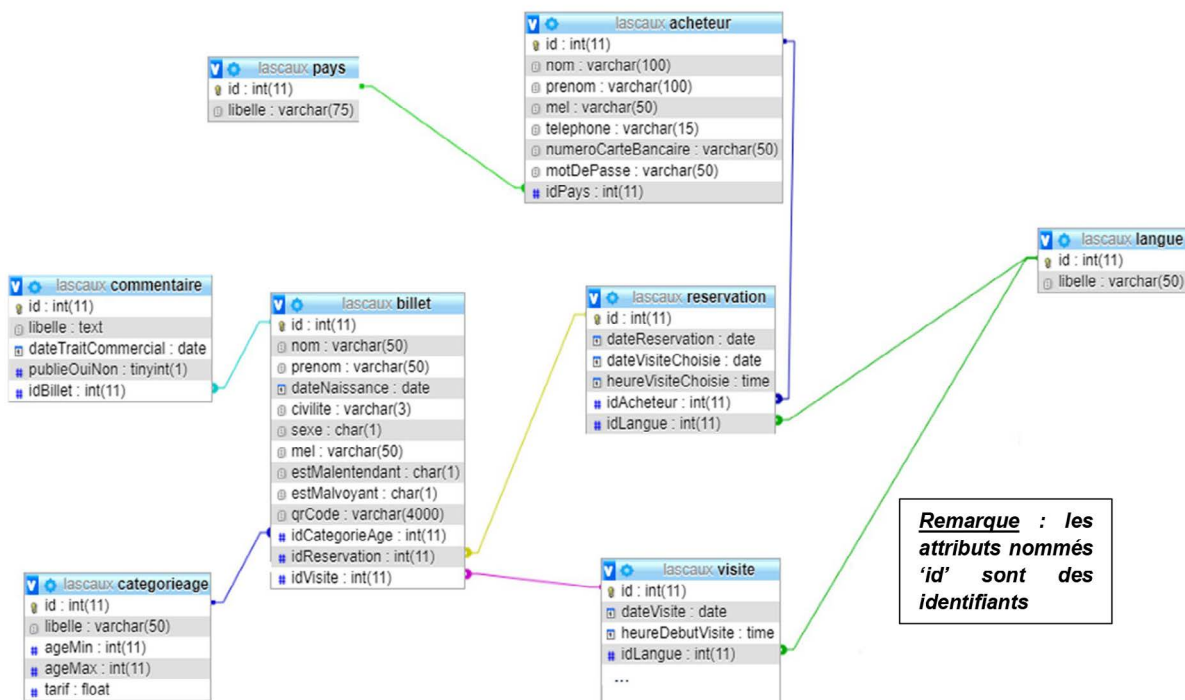
34. -- Règle 3 : à compléter sur votre copie => Reporter le n° de ligne 34

35. END $$ ;
36. DELIMITER ;

```

Document A4 **Extrait du schéma relationnel de la base de données active lascauxprod**

Extrait du schéma relationnel sous forme graphique



Extrait du schéma relationnel sous forme textuelle de la base de données active

Pays (id, libelle)

Clé primaire : id

CategorieAge (id, libelle, ageMin , ageMax, tarif)

Clé primaire : id

Langue (id, libelle)

Clé primaire : id

Acheteur (id, nom, prenom, mel, telephone, numeroCarteBancaire, dateValiditeCarteBancaire, mot_de_passe, idPays)

Clé primaire : id

Clé étrangère : idPays en référence à id de Pays

Reservation (id, dateReservation, dateVisiteChoisie, heureVisiteChoisie, idAcheteur, idLangue)

Clé primaire : id

Clés étrangères : idAcheteur en référence à id de Acheteur
idLangue en référence à id de Langue

Visite (id, dateVisite, heureDebutVisite, idLangue, ...)

Clé primaire : id

Clés étrangères : idLangue en référence à id de Langue

...

Billet (id, nom, prenom, dateNaissance, civilite, sexe, mel, estMalentendant, estMalvoyant, qrCode, idCategorieAge, idReservation, idVisite)

Clé primaire : id

Clés étrangères : idCategorieAge en référence à id de CategorieAge
idReservation en référence à id de Reservation
idVisite en référence à id de Visite

Commentaire (id, libelle, dateTraitCommercial, publieOuiNon, idBillet)

Clé primaire : id

Clé étrangère : idBillet en référence à id de Billet

Base de données d'archivage : lascauxarchive

Extrait du schéma relationnel de la sous forme textuelle :

AcheteurArchive (id, nom, prenom, mel, telephone, numeroCarteBancaire, dateValiditeCarteBancaire, motDePasse, idPays, dateArchivage, dateFinConservation)

Documents associés au dossier B

Document B1 Scénarios de risque (abuser story ou evil user story)

Numéro de l'événement	Scénario de risque	Probabilité/Gravité	Mesures à prévoir
1	En tant qu'attaquant externe, je peux altérer les données de la base	(3) (4)	1. Valider les entrées pour éviter des injections SQL. 2. Réduire les privilèges et droits d'accès à la base de données.
2	En tant que guide vacataire, je peux planifier des visites	(2) (3)	2.1 Restreindre les fonctionnalités de planification des visites aux seuls guides gestionnaires au niveau de l'application. 2.2 Restreindre les accès à la table Visite aux seuls guides gestionnaires.
3	En tant que guide vacataire malhonnête, je peux récupérer les tickets de mes visiteurs afin d'ajouter des commentaires en ma faveur	(2) (3)	3.1. Retirer les droits d'accès à la table des commentaires pour les guides. 3.2. Exiger une authentification des visiteurs pour la saisie des commentaires (application web).
4	En tant que guide jaloux, je peux ajouter à un guide une langue qu'il ne parle pas afin de lui nuire	À compléter sur votre copie	À compléter sur votre copie

Échelle de probabilité : (1) : Minime, (2) : Significative, (3) : Forte, (4) : Maximale

Échelle de gravité : (1) : Négligeable, (2) : Limitée, (3) : Importante, (4) : Critique

Document B2 Syntaxe MySQL (extrait du manuel de référence de MySQL 8.0)

L'instruction CREATE ROLE permet la création d'un nouveau rôle MySQL qui peut être appliqué à plusieurs utilisateurs. La syntaxe générale est : `CREATE ROLE 'nom_du_role' ;`

`nom_du_rôle` respecte la syntaxe suivante `'role_name'[@'host_name']` où `'role_name'` est le nom du rôle et le `'host_name'` (facultatif) peut être l'adresse IP d'une machine ou d'un réseau. Par exemple : `'role_name'@'10.%.%.%'` ou `'role_name'@'10.0.0.0/255.255.0.0'` ou `'role_name'@'10.0.0.0/16'`.

Pour attribuer des privilèges ou des rôles, il faudra utiliser l'instruction GRANT.

```
GRANT priv_type [(column_list)][, priv_type [(column_list)]] ...  
ON [object_type] priv_level  
TO user_or_role [, user_or_role] ...
```

`priv_type` contient les privilèges attribués. Les valeurs autorisées sont : ALTER, CREATE VIEW, CREATE, DELETE, DROP, GRANT OPTION, INSERT, SELECT, UPDATE et ALL.

`ON` contient le nom de la base de données suivi du nom de la table : `db_name.tbl_name`.

`TO` contient le nom du compte (utilisateur ou rôle).

Pour supprimer un privilège : `REVOKE [priv_type | ALL] ON object_type FROM user_or_role ;`

Document B3 Code partiel de l'application de gestion des visites

Classe Guide : gestion des guides

```
public class Guide {
    private int id ;
    private String nom ; private String prenom ;           private Date dateEmbauche;
    private String telephone;    private String mel;           private String motDePasse;
    private Boolean isVacataire = false;    // Guide non vacataire

    public Guide() { super(); }
    public Guide(String nom, String prenom, String tel, String mel, String mdp) {
        super();
        this.nom = nom;           this.prenom = prenom; this.telephone = tel;
        this.mel = mel;           this.motDePasse = mdp;
    }
    public Guide(String nom, String prenom, String tel, String mel, String mdp,
        Boolean isVacataire)
    {
        super();
        this.nom = nom;           this.prenom = prenom; this.telephone = tel;
        this.mel = mel;           this.motDePasse = mdp;    this.isVacataire =
isVacataire;
    }
}
```

Classe FormulaireInscriptionVacataire : Formulaire d'inscription d'un guide vacataire

```
public class FormulaireInscriptionVacataire {
    // Déclaration des composants graphiques et Construction du formulaire (code non présenté)
    public void formInscriptionGuideVacataire() { // Traitement du formulaire
        jbtnEnregistrer.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                if (e.getSource() == jbtnEnregistrer) { // On a soumis le formulaire
                    String nom = jtfNom.getText();
                    String prenom = jtfPrenom.getText();
                    String tel = jtfTel.getText();
                    String mel = jtfMel.getText();
                    String mdp = jtfMdp.getPassword().toString() ;
                    Guide guideVacataire = new Guide(nom, prenom, tel, mel, mdp) ;
                    ModeleLascaux.enregistrerGuide(guideVacataire);
                }
            }
        });
    }
}
```

Classe ValidationFormulaire – Extrait

```
public class ValidationFormulaire {
    // retourne vrai si le mail vérifie l'expression régulière définie dans la variable pattern
    public static boolean validerEmail(String mail) {
        String pattern = "^([a-zA-Z0-9_-]+)(\\.[a-zA-Z0-9_-]+)*@([a-zA-Z0-9_-]+\\.){2,7}$";
        Pattern pat = Pattern.compile(pattern);
        Matcher matcher = pat.matcher(mail);
        return matcher.matches();
    }
}
```

Classe ModeleLascaux : Accès à la base de données

```
public class ModeleLascaux {
    private static Connection connexion ;
    /* Connexion à la base de données, paramètres de connexion dans un fichier externe */
    public static Connection getConnexion() {
        // Pattern Singleton
    }

    public static Guide authentifier(String mail, String motDePasse)
    {
        connexion = getConnexion();
        Statement stmt = null; ResultSet rs = null; Guide guide = null;

        String sql = "select guide.* from guide "
            + " where mel = ' "+mail+" ' and mot_de_passe = ' "+motDePasse+" ' ";
        try {
            stmt = connexion.createStatement();
            rs = stmt.executeQuery(sql);
            if (rs.next()) {
                guide = new Guide(    rs.getInt("id"), rs.getString("nom"),
                                    rs.getString("prenom"), rs.getString("telephone"),
                                    rs.getString("mel"), rs.getString("motDePasse"));
            }
        } catch (SQLException e) { e.printStackTrace(); }
        return guide ;
    }

    public static void setLog(String uneDate, String requete, String melGuide) {
        connexion = getConnexion();
        String adresseIP = (String) InetAddress.getLocalHost().getHostAddress();
        String evtSql ;          // Requête SQL qui a été lancée
        int nb = 0 ;

        // Question B2.1 – Code à compléter sur votre copie

    }

    public static int enregistrerGuide(Guide guide) {
        connexion = ModeleLascaux.getConnexion();
        int nb = 0;
        PreparedStatement pstmt = null;

        String sql = "insert into guide(nom, prenom, telephone, mel, mot_de_passe) "
            + "values(?,?,?,?,?)";

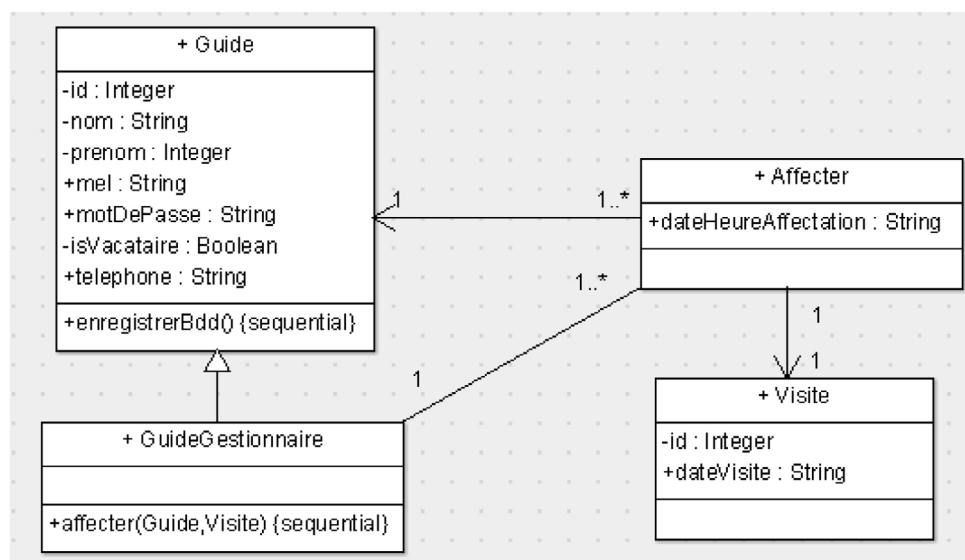
        try {
            pstmt = connexion.prepareStatement(sql);
            pstmt.setString(1, guide.getNom());    pstmt.setString(2, guide.getPrenom());
            pstmt.setString(3, guide.getTelephone());
            pstmt.setString(4, guide.getMel());
            pstmt.setString(5, guide.getMotDePasse());
            nb = pstmt.executeUpdate();
        } catch (SQLException e) { e.printStackTrace(); }
        return nb ;
    }
}
```

Document B4 Exemple de test unitaire

```
class Compteur {  
    private int niveau;  
    public int getNiveau() {  
        return niveau;  
    }  
    public void incrementer() {  
        niveau++;  
    }  
}
```

```
class CompteurTest  
{  
    @Test  
    void testIncrementer() {  
        Compteur compteur = new Compteur();  
        compteur.incrementer();  
        assertEquals(1,compteur.getNiveau());  
    }  
}
```

Document B5 Diagramme de classes partiel de la planification des visites



Document B6 Expressions régulières

Une expression régulière est une chaîne de caractères qui décrit un motif dans une séquence de caractères. Elle peut être utilisée pour valider une séquence de caractères ou rechercher des motifs.

Méta-caractère	Description	Méta-caractère	Description
[]	Ensemble de caractères Exemple [a-z] : minuscules	{min,max}	Quantificateur de l'expression qui précède
^	Début de ligne	\$	Fin de ligne
\	Le caractère qui suit n'est pas considéré comme méta-caractère	*	0 ou plusieurs fois l'expression qui précède
.	Remplace n'importe quel caractère	?	0 ou plusieurs fois l'expression qui précède

Documents associés au dossier C

Document C1 Codes partiels de l'enregistrement d'un acheteur

Classe Securite

```
< ?php
Class Securite {
    // Méthode vulnérable
    /**
     * Utilise la méthode sha1() pour hacher un mot de passe
     * @param string $motDePasse mot de passe à hacher
     * @return string $motDePasseHache mot de passe haché et stocké sur 32 caractères
     */
    public static function hacherMotDePasse($motDePasse) {
        $salt = 'unPetitSelPourRenforcer!' ; // Non recommandé : valeur non aléatoire
        return sha1($motDePasse.$salt) ;
    }

    // Réécriture de la méthode hacherMotDePasse($motDePasse)
    // Générer le sel avec la fonction sha256(random_bytes(int $longueur)).
    // $longueur représente la taille de l'argument $motDePasse.
    // La méthode password_hash() utilisera l'algorithme qui garantit le plus de sûreté
    public static function hacherMotDePasse($motDePasse) {

        À compléter sur votre copie

    }

    /**
     * Hacher une liste des mots fournis en paramètre
     * @param string[ ] $listeMdp tableau de mots de passe à chiffrer
     * @return string[ ] $listeMdpHaches tableau des mots de passe chiffrés
     */
    public static function hacherListeMotsDePasse (array $listeMdp) {

        À compléter sur votre copie

    }
}
} ?>
```

Classe ValidationForm

```
<?php
use Modele\PdoLasc ;
class ValidationForm
{
    const LG_MAX_MDP = 15 ; // Longueur maximale d'un mot de passe
    private getConnexion() { // Récupérer la référence de connexion PDO à la base de données
        $pdo = PdoLasc::getPdo () ;
        return $pdo ;
    }
    static function nettoyer($data) {
        return htmlspecialchars(trim($data)) ; // supprimer les espaces et échapper certains caractères
    }
    static function validerMdp($mdp) {
        return (strlen($mdp) <= self::$LG_MAX_MDP) ;
    }
}
?>
```

Document C2 Fonction de hachage – Extraits de la documentation PHP

Fonction password_hash() : la fonction PHP password_hash() crée un nouveau hachage en utilisant un algorithme de hachage fort et irréversible.

Syntaxe : password_hash(string **\$password**, string | int | null **\$algo**, array **\$options** = []) : string

Arguments :

\$password : chaîne de caractères qu'on veut hacher.

\$algo : algorithme utilisé pour le hachage. Deux constantes sont possibles :

PASSWORD_DEFAULT : utilisation de l'algorithme bcrypt. Cette constante est conçue pour évoluer dans le temps, au fur et à mesure que des algorithmes plus récents et plus forts sont ajoutés à PHP. Pour cette raison, la longueur du résultat peut changer, il est donc recommandé de stocker le résultat dans une colonne de la base de données qui peut contenir au moins 60 caractères (255 caractères peuvent être un très bon choix, mais cela peut évoluer).

PASSWORD_BCRYPT : utilise l'algorithme CRYPT_BLOWFISH pour créer la clé de hachage. Le résultat sera toujours une chaîne de 60 caractères, ou false en cas d'erreur.

\$options : paramètre optionnel attend un tableau associatif contenant des options avancées qui dépendent de l'algorithme utilisé. Les options supportées pour PASSWORD_BCRYPT sont :

cost : détermine le coût algorithmique qui sera appliqué. Plus ce coût est grand, plus le temps de traitement augmente. S'il est omis, la valeur 10 sera utilisée par défaut.

salt (sel) : valeur aléatoire unique qui permet de se prémunir des attaquants utilisant des tables pré-calculées.

Valeur de retour : le mot de passe haché ou FALSE en cas d'échec.

Document C3 Erreur d'enregistrement d'un nouvel acheteur

Insertion d'un nouvel acheteur avec hacherMotDePasse() utilisant la méthode password_hash()

```
$2y$10$a6f16ab483da9847d431au87Fj/W4wA5KXPUGRK0UMVsySY7peSGy
```

```
Array ( [0] => 22001 [1] => 1406 [2] => Data too long for column 'mot_de_passe' at row 1)
```

Document C4 Politique de sécurité des mots de passe – Extrait du guide de l'ANSSI

La règle R21 : il est recommandé de définir une longueur minimale pour les mots de passe lors de leur création. Les recommandations minimales en fonction du niveau de sensibilité sont :

Niveau de sensibilité	Longueur minimale en nombre de caractères	Taille de clé équivalente en bits (5)
Faible à moyen	Entre 9 et 11	≈ 65
Moyen à fort	Entre 12 et 14	≈ 85
Fort à très fort	Au moins 15	≥100

La règle R22 : selon les systèmes, il est recommandé de ne pas fixer de limite à la longueur maximale d'un mot de passe afin de permettre aux utilisateurs d'avoir recours à des phrases de passe ou longs mots de passe.

Document C5 Exemple d'utilisation d'un tableau

```
$lesChiffres = array() ; // tableau de nombres
$lesChiffres[ ] = 10 ; // ajoute le nombre 10 au tableau
// Parcours du tableau
foreach($lesChiffres as $unChiffre) {
    echo $unChiffre ;
}
```

Document C6 Exemple de création d'une requête préparée

```
<?php
$stmt = $pdo->prepare('SELECT car_name, car_type FROM car WHERE car_id = :id');
$stmt->bindValue(':id', 3, PDO::PARAM_INT);
$stmt->execute();

print_r($stmt->fetchObject());
?>
```

Pour une valeur de type chaîne, le 3ème paramètre de la méthode bindValue est **PDO::PARAM_STR**

Dans notre exemple, l'appel de la méthode bindValue permet d'associer la valeur 3 au paramètre :id de la requête préparée.

Document C7 Manipulation de chaînes de caractères en PHP

Fonction	Signification
<code>preg_match(string \$pattern, string \$chaine) : boolean</code> <u>Exemple</u> : <code>\$pattern = '/[0-5]/'</code> ; <code>preg_match(\$pattern, "Ceci est un exemple")</code> retourne false car la chaîne ne contient pas de chiffres	Cherche le motif fourni par \$pattern dans la chaîne de caractères \$chaine.
<code>strlen(string \$chaine) : int</code>	Retourne la longueur d'une chaîne de caractères

Documents associés au dossier D

Document D1 Architecture technique des applications de Lascaux IV

