E6 : Cybersécurité des services informatiques

ÉTUDE DE CAS

CAS Lascaux IV

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.

Depuis décembre 2016, le Centre international de l'art pariétal de Montignac Lascaux (Ciap) offre au public une réplique de la grotte, appelée Lascaux IV, sur le site de Lascaux. Un des objectifs de ce site est d'utiliser les technologies d'aujourd'hui pour permettre à chaque visiteur de personnaliser sa visite : avant, pendant, après.

Cette grotte comporte une partie musée originale où le visiteur est en interaction permanente avec ce qu'il côtoie, grâce à l'utilisation d'une tablette appelée compagnon de visite (CDV). Le Ciap dispose de mille six cents CDV en dix langues.

Le CDV est connecté à un maillage de dispositifs Wi-Fi et *Bluetooth* du bâtiment composé de cinquante bornes Wi-Fi et de deux cents balises *Bluetooth* qui permettent au CDV de se localiser, de communiquer et aussi de se synchroniser avec les serveurs multimédias du Ciap. Des centaines de tablettes sont prêtées chaque jour pour permettre à chaque visiteur d'avoir une expérience numérique unique et personnalisée.

L'organisation cliente

Le conseil départemental de la Dordogne est le maître d'ouvrage du projet Lascaux IV :

- Lascaux IV dispose de plusieurs systèmes d'information (SI) internes : SI bâtimentaire, SI administratif (ressources humaines, comptabilité, etc.), SI visiteur et 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. Application tablette pour les CDV, site internet, plateforme d'échanges (MyLascaux) 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.

- Le SI administratif est exploité par trois personnes. Outre la gestion avec un progiciel de gestion intégré, il gère les employés (128 en haute saison en 2019 par exemple) avec un annuaire d'entreprise, ce dernier permettant de gérer les différents niveaux d'habilitations dans l'organisation.

- La DSI lance les appels d'offre, expertise les réponses, donne aux prestataires retenus la stratégie numérique, un cadre de standards à respecter (s'entendre sur un vocabulaire et une cible commune) et fait le lien avec les utilisateurs.

Plusieurs appels d'offres ont été lancés. Différentes entreprises de services numériques (ESN) ont obtenu des lots des 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. Soucieuse de s'inscrire dans une vision durable et responsable de son activité, cette entreprise est labellisée entreprise numérique responsable (ENR) chaque année, et ce depuis 2012.

Le premier appel d'offre concerne l'application *Web* de la gestion des réservations de billets et de l'organisation des visites. Le deuxième appel d'offre porte sur l'application mobile permettant une visite interactive du musée avec le compagnon de visite (CDV).

Aquilasc est composée de 30 personnes et dispose d'un budget d'investissement de 1,2 million d'euros et d'un budget de fonctionnement de 1,7 million d'euros. Elle dispose de 4 serveurs pour ses activités ; un poste de travail et un mobile multifonction (*smartphone*) sont mis à disposition de chaque employé.

Depuis quatre ans, Aquilasc utilise la méthode agile *Scrum* pour la gestion de ses projets. Cette méthode permet d'impliquer le client 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 conseil départemental de la Dordogne et Aquilasc. Il définit la nature des interventions de Aquilasc, leurs durées et délais et établit les métriques selon lesquelles la prestation sera jugée finie et délivrée.

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

En mode agile, vous êtes chargé d'analyser les spécifications techniques pour concevoir, développer et maintenir les logiciels. Les bases de données utilisées par les applications sur lesquelles vous allez intervenir sont gérées par le système de gestion de base de données (SGBD) *MySQL*. Vous testez et intégrez en continu les solutions développées.

Au sein de l'équipe technique, vous devrez :

- participer à l'atelier analyse de risques ;
- sécuriser des données ;
- gérer l'identification des utilisateurs ;
- préparer l'environnement de développement et piloter les sous-traitants.

Vous vous appuierez sur les dossiers documentaires mis à votre disposition.

Dossier A – Participation à l'atelier d'analyse des risques sur l'application *Web*

Votre équipe est en charge du développement des nouvelles fonctionnalités de l'application *Web* permettant notamment la réservation et l'achat en ligne de billets de visite par des acheteurs (particulier, agence de voyages, comité d'entreprise, etc.). Une autre fonctionnalité attendue rapidement est la possibilité pour les visiteurs de créer un compte sur l'application Web MyLascaux après leur visite avec leur numéro de billet pour retrouver les données collectées par le compagnon de visite (CDV) lors de leur visite.

Votre équipe est en charge du développement de ces nouvelles fonctionnalités.

Mission A1 – Evaluation des risques à partir des récits utilisateurs

Votre administrateur *Scrum* (*Scrum Master*), dont le rôle est de veiller à ce que l'équipe respecte les règles de la méthode *Scrum*, vous demande de participer à un atelier d'analyse de risque au sein de l'équipe de développement afin d'évaluer les risques sur les récits utilisateurs (*user stories*) qui ont été planifiées pour la première itération de développement (*sprint* 1).

Question A1.1
Indiquer si le tableau contenant les acteurs à l'origine de malveillance est complet. Justifier votre réponse.

Le tableau « Besoins de sécurité pour les récits utilisateurs » propose pour chaque récit de l'itération (*sprint*) une évaluation du besoin de disponibilité, d'intégrité et de confidentialité des données manipulées et la nécessité d'éléments de traçabilité faisant office de preuve. L'évaluation des récits utilisateurs n'a pas été finalisée.

Question A1.2
Proposer une évaluation des récits utilisateurs 1 et 25 pour chacun des 4 critères (disponibilité, intégrité, confidentialité et preuve).

Mission A2 – Gestion des évènements redoutés

Le tableau « Impacts des événements redoutés » permet de définir l'impact et la gravité en terme de sécurité des événements liés à des actes de malveillance pour l'entreprise.

Question A.2.1
Proposer, pour les événements 1 et 3 fournis dans le tableau, les impacts pour l'entreprise et une estimation de leur gravité.

Pour l'instant, le tableau « Scénarios de risque et mesures à prévoir » du dossier documentaire contient les mesures à appliquer pour contrer l'évènement redouté numéro 2.

Le responsable de produit (*Product Owner*) a identifié un événement redouté numéro 4 concernant le récit utilisateur (*user story*) numéro 2 qui porte sur l'impression des billets au format PDF (Portable Document Format).

Question A.2.2
Proposer des mesures à prévoir lors du développement pour contrer l'évènement redouté numéro 4.

Question A.2.3
Proposer un scénario de risque (*abuser story*) et des mesures à prévoir pour l'événement redouté numéro 3.

Question A.2.4
Proposer deux évènements redoutés en lien avec les besoins de sécurité du récit utilisateur (*user story*) 15.

Mission A3 – Prise en compte du règlement général sur la protection des données (RGPD) dans les récits utilisateurs

Question A3.1
Identifier, pour chacun des récits utilisateurs (*user stories*) numérotés 22 et 25, les actions techniques et réglementaires à mettre en œuvre pour respecter le RGPD.

Le responsable de produit (*Product Owner*) vous demande d'ajouter un nouveau récit utilisateur (*user story*) : "En tant que visiteur, je veux être déconnecté du site lorsque je ferme mon navigateur ou à l'issue d'une période d'inactivité fixée".

Question A3.2
Expliquer quel risque de sécurité vient contrer cet ajout.

Dossier B – Sécurisation des données

La réservation des billets est possible sur internet : un acheteur (particulier, agence de voyages, comité d'entreprise, etc.) peut réserver en indiquant à quel moment il souhaite venir et les personnes qui participeront à cette visite.

Le jour de la visite, un guide prend en charge tout le groupe et réalise la visite. Á la fin de la visite, chaque personne en possession d'un billet peut déposer des commentaires sur le déroulé de la visite.

Le module réservation des billets en ligne est en cours de développement. L'équipe que vous avez rejointe est en phase de test des récits utilisateurs (*user stories*) pris en charge au cours de la première itération (*sprint* 1).

Mission B1 – Vérification de la confidentialité des données

Denise Bradord intervient à la fin de chaque itération (*sprint*) pour valider la qualité du produit avant la livraison. Les tests qu'elle vient de réaliser ont mis en évidence des problèmes de confidentialité au niveau :

- de la table qui contient les caractéristiques des acheteurs ayant réalisé une réservation ;
- de l'implémentation du récit utilisateur (*user story*) n° 5 : celui-ci permet aux acheteurs de consulter leurs réservations réalisées pour des seniors durant un mois précis.

Elle vous a adressé un courriel pour vous informer des problèmes rencontrés sur le récit utilisateur (*user story*) n° 5.

Denise Bradord vous confie la résolution de ces problèmes.

Question B.1.1
a) Identifier les données personnelles présentes sur la représentation conceptuelle de la base de données.
b) Identifier, parmi ces données personnelles, celles qui sont sensibles.

Question B.1.2
Lister les données devant être chiffrées et celles devant être hachées, pour assurer la confidentialité des données de la table Acheteur.

Mission B2 – Sécurisation de l'accès à une base de données

Suite à votre intervention, les fonctionnalités développées lors de la première itération (s*print*) viennent d'être intégrées en production.

La fonctionnalité permettant au service commercial de visualiser les commentaires laissés suite aux visites doit être développée lors la deuxième itération (*sprint 2*).

Chaque visiteur peut, suite à sa visite, déposer plusieurs commentaires à partir de l'application *Web*. Le service commercial a un rôle de modérateur de ces commentaires : il peut choisir de les publier ou non. Par ailleurs, il peut éventuellement répondre aux clients.

Question B.2.4
Expliquer en quoi la modération des commentaires est un enjeu important pour Lascaux.

Les commerciaux viennent de constater que le nombre de commentaires à traiter a très fortement augmenté, de manière inexplicable.

Roger Zanches, responsable du service commercial, ne voudrait pas que certains commentaires proviennent d'une personne malveillante.

Il vous demande de vérifier s'il existe des commentaires qui ne seraient pas reliés à un billet.

Question B.2.5
Proposer à Roger Zanches une requête pour répondre à son besoin.

Mission B3 – Adaptation de la représentation conceptuelle de la base de données

IMPORTANT : la candidate ou le candidat peut retenir le formalisme de son choix (schéma entité-association, diagramme de classes) pour représenter les évolutions conceptuelles demandées

La deuxième itération (*sprint 2*) nécessite de faire évoluer la représentation conceptuelle de la base de données.

Le responsable du service des ressources humaines a constaté que les commentaires sur le niveau de langue des guides étaient parfois inappropriés, voire insultants, pour certains guides. Étant donné que l'article 12 du RGPD indique que toute personne physique peut exercer son droit

d'accès aux données qui la concernent, il vous demande d'adapter la base de données pour prendre en compte l'article 6 de la loi «Informatique et Libertés» qui prévoit que les informations collectées doivent être adéquates, pertinentes et non excessives au regard de la finalité poursuivie.

Par ailleurs, le délégué à la protection des données (DPO) vous demande :

- d'appliquer certaines règles de sécurité pour la gestion des mots de passe des acheteurs ;
- de proposer une mise en conformité de la structure de la base de données d'après la fiche de registre relative au traitement de réservation des billets qu'il a complétée.

En respectant les consignes du responsable du service des ressources humaines et du DPO, détaillées dans le dossier documentaire, vous devez faire évoluer la représentation conceptuelle de la base de données.

Question B.3.1
Identifier et justifier les données devant être supprimées pour une mise en conformité vis à vis de la fiche de registre établie par le DPO.

Question B.3.2
Proposer les modifications à réaliser pour répondre aux nouvelles exigences. Seuls les éléments du schéma existant qui sont concernés par l'évolution seront repris dans le schéma proposé.

Dossier C – Gestion de l'identification des utilisateurs

IMPORTANT : la candidate ou le candidat peut choisir de présenter les éléments de code à l'aide du langage de programmation de son choix ou de pseudo-code algorithmique.

Une nouvelle itération (*sprint*) commence ; vous êtes maintenant en charge du développement de l'authentification des utilisateurs sur les compagnons de visite (CDV). Ces tablettes peuvent être utilisées par n'importe quel visiteur ou guide accompagnateur. Vous reprenez le travail effectué par votre prédécesseur lors de l'itération précédente et pouvez prendre connaissance de l'architecture logicielle utilisée en consultant le document C2.

Mission C1 – Identification des visiteurs

Dès le départ de la visite, le guide remet un CDV et invite le visiteur à scanner le code QR (*QrCode)* placé sur son billet. Le visiteur sera ainsi identifié. L'objectif de cette identification est de proposer au visiteur des fonctionnalités différentes selon sa catégorie d'âge (par exemple : un contexte ludique pour les enfants). L'application d'identification a été partiellement réalisée en *Java*.

Principe de fonctionnement :

La tablette établit, en Wi-Fi, des connexions avec un serveur par le biais de requêtes suivant le protocole de transfert hypertexte (*Hypertext Transfer Protocol - HTTP*). Ces dernières déclenchent l'exécution de scripts (en langage *PHP*) qui interrogent le serveur de base de données. Les résultats récupérés seront convertis au format de notation des objets en *JavaScript* (*JavaScript Object Notation - Json*) et renvoyés à la tablette.

Évolutions envisagées **:**

- Vérification de la logique du code et les possibilités d'attaques par injection de code lors de l'utilisation des variables $_POST, $_GET, $_COOKIE ;
- Évolution du script *getLeVisiteurById.php* par l'utilisation d'une requête préparée ;

Question C.1.1
Identifier deux faiblesses du script *getVisiteur.php* du point de vue de la cyber sécurité, en expliquant leurs conséquences possibles sur le système.

Question C.1.2
Le manque de robustesse de la méthode ***jsonStringToVisiteur*** de ***UtilVisiteur*** a été identifié par un collaborateur et le prouve par l'intermédiaire d'un test unitaire (document Cx)
- Expliquer ce que le test unitaire cherche à démontrer
- Proposer une solution au problème signalé (modification à apporter au corps de la fonction)

Question C.1.3
Décrire les mesures à mettre en place pour éviter qu'un visiteur peu scrupuleux puisse scanner un billet trouvé par terre ou dans une poubelle et ainsi effectuer une visite avec un billet déjà utilisé.

Mission C2 – Identification des guides

Les guides utilisent également un le compagnon de visite (CDV) pour gérer leurs visites. Ils ont un code QR (*QrCode)* spécifique qui leur donne un accès à d'autres fonctionnalités sensibles interdites aux simples visiteurs. L'identification du guide est complétée par la saisie d'un mot de passe.

Principe actuel de fonctionnement du changement de mot de passe :

Lors de sa première connexion, un guide doit obligatoirement modifier le mot de passe attribué par défaut. Par la suite, il peut changer son mot de passe lorsqu'il le désire.

La date de création d'un mot de passe est conservée conjointement au mot de passe (classe MotDePasse). À chaque changement de mot de passe, on vérifie que le nouveau mot de passe n'est pas identique à l'ancien et, si ce n'est pas le cas, le changement est validé et l'ancien mot de passe est ajouté à la liste des anciens mots de passe du guide.

La fonctionnalité de changement du mot de passe doit être améliorée lors de cette nouvelle itération (*sprint*).

Évolutions envisagées :

Lors de la connexion sur le CDV, le guide sera invité à changer son mot de passe si celui-ci date de plus de trois mois. Une méthode *doitChangerMdp()* vérifiera la date de validité du mot de passe et retournera un booléen contenant vrai si le mot de passe doit être changé, faux sinon.

Par ailleurs, la méthode s*etMotDePasse()*, utilisée pour changer le mot de passe, doit être modifiée : le nouveau mot de passe ne doit pas être identique à l'ancien, ni même à l'un des mots de passe utilisés par le guide durant les douze derniers mois.

Question C.2.1
En vous basant sur la documentation technique du document C8, rédiger la méthode *doitChangerMdP*() de la classe Guide.

Question C.2.2
Modifier le corps de la méthode *setMotDePasse*() de la classe Guide afin de prendre en compte la nouvelle contrainte de sécurité demandée lors de cette nouvelle itération (*sprint*).

Le délégué à la protection des données (DPO) s'interroge sur la pertinence de la durée de recherche de l'historicité du mot passe à 12 mois.

Question C.2.3
Argumenter en faveur ou non de cette durée.

Dossier D – Préparation du développement et pilotage de la sous-traitance

L'itération zéro (*sprint 0*) est un moment privilégié pour l'équipe qui va apprendre à se connaître et à travailler ensemble. Cette itération n'apporte pas de valeur immédiate ; elle ne se termine pas forcément par une livraison. Cependant, c'est un investissement essentiel dans l'avenir du projet qui permet à toute l'équipe de partager une vision claire du projet sur 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 – Rejet des mauvaises pratiques de développement

Des propositions pour le mode de travail sur le projet ont été émises par les différents participants lors de cette itération zéro (*sprint 0*). Elles sont dans le dossier documentaire.

Question D.1.1
Relever les numéros des propositions qu'il faut rejeter à tout prix et justifier votre position pour chacune des propositions rejetées.

Documents associés au dossier A

Document A1 : Besoins de sécurité pour les récits utilisateurs (user stories) de la première itération de l'application gestion des billets et des visites sur le site Web (extraits)

1	Intitulé de la *user story*	Disponibilité	Intégrité	Confidentialité	Preuve
1	En tant qu'acheteur, je veux acheter en ligne les billets pour plusieurs personnes afin de pouvoir participer à une visite.				
2	En tant qu'acheteur, je veux télécharger les billets d'une de mes réservations au format *PDF* afin de les transmettre aux personnes pour lesquelles j'ai réalisé la réservation.	**	**	*	-
5	En tant qu'acheteur, je veux consulter les caractéristiques des seniors pour lesquels j'ai acheté un billet sur un mois donné afin de réaliser une campagne publicitaire.	-	*	**	-
15	En tant que visiteur, je veux poster un commentaire afin de donner mon avis sur la qualité de la visite organisée.	*	**	-	*
22	En tant que visiteur, je peux créer, à l'issue de ma visite, un compte afin de retrouver sur le site Web les photos, vidéos et expériences effectuées sur le CDV.	*	**	**	*
25	En tant que responsable commercial, je veux consulter les statistiques de temps passé par zone de visite et les activités réalisées par les visiteurs afin de proposer un meilleur service aux visiteurs.				
27	En tant que visiteur, je veux être déconnecté du site lorsque je clique sur le bouton déconnexion afin de ne plus être identifiée.	**	*	-	-
30	En tant que guide je veux modifier mon mot de passe sur l'application mobile en toute sécurité afin de sécuriser mon compte.	*	**	**	*

- : pas de besoin ** : besoin important* *** : besoin très important*

Disponibilité : la fonctionnalité doit être utilisée au moment voulu.
Intégrité : les données doivent être exactes et complètes.
Confidentialité : les informations ne doivent pas être divulguées.
Preuve : les traces de l'activité du système sont opposables en cas de contestation.

Document A2 : Extrait de l'analyse des risques et menaces de l'environnement

Acteurs à l'origine de la malveillance

Acteurs malveillants	Modes opératoires	Probabilité
Attaquant externe (*hacker*)	L'attaquant externe accède à la base de données.	*
	L'attaquant externe surcharge le système.	***
Acheteur	L'acheteur envoie de fausses informations.	**
	L'acheteur surcharge le système.	*
Visiteur	Le visiteur envoie de fausses informations.	*
	Le visiteur surcharge le système.	*

** : faible probabilité* *** : forte* **** : très forte probabilité*

Impacts des événements redoutés

Numéro de l'événement	Événement	Impact pour l'entreprise	Gravité
1	Le système ne répond pas.		
2	Un attaquant parvient à créer des billets sans payer.	Perte financière pour l'entreprise.	**
3	Un attaquant parvient à modifier les affectations des guides aux visites.		
4	Un acheteur accède aux billets au format *PDF* d'un autre acheteur en modifiant le numéro de réservation dans la barre d'adresse.	Problème de confiance. Désorganisation des visites	*

** : modérée* *** : très élevée*

Scénarios de risques (*abuser story, evil user story*) et mesures à prévoir

Numéro de l'événement	scénario de risque (*abuser story, EUS*)	Mesures à prévoir
2	En tant qu'attaquant externe, je souhaite créer des billets sans payer ni réserver.	2.1 Les fonctions de création/réservation de billet (visiteur, acheteur) sont soumises à habilitation.
		2.2 Tout billet doit être obligatoirement associé à une réservation.
		2.3 Les opérations en écriture font l'objet d'une journalisation.
4	En tant qu'acheteur, je peux imprimer les billets d'un autre acheteur.	

Documents associés au dossier B

Document B1 : 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 de pouvoir participer à une visite guidée.

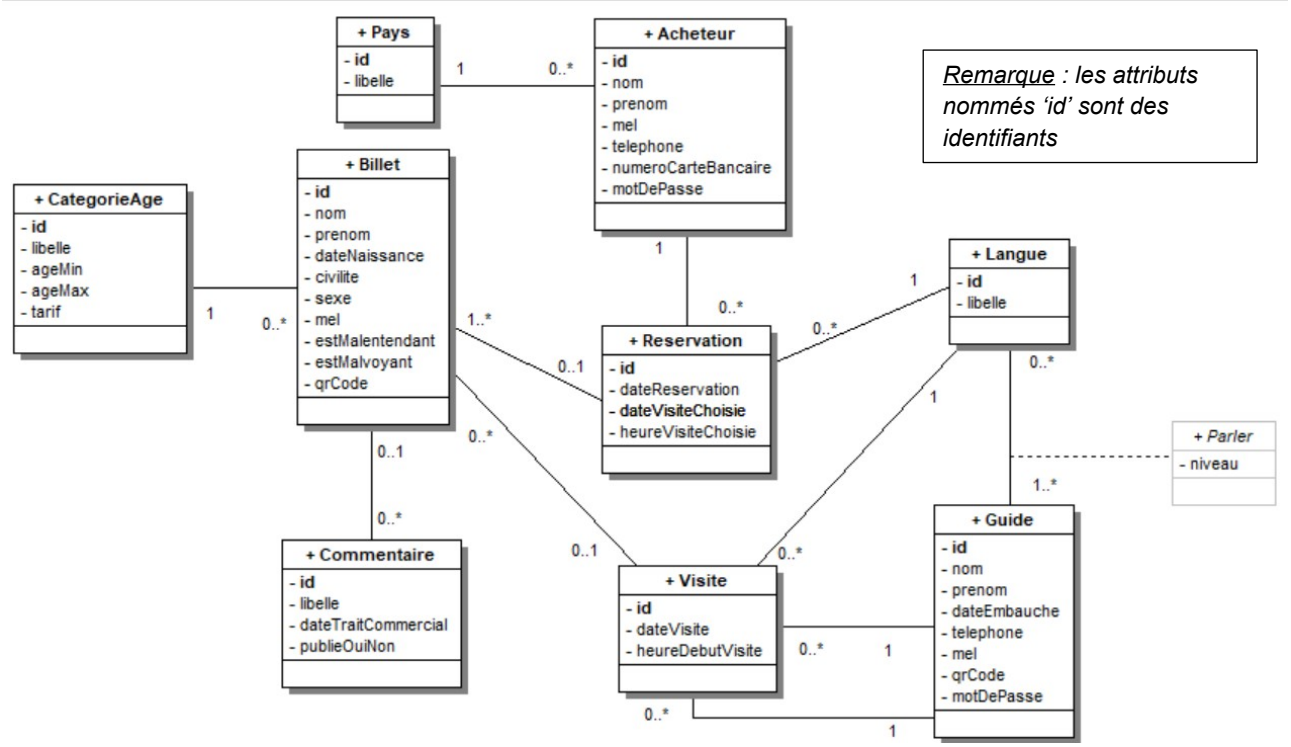
Rattachement : itération (*sprint*) n°1

Critères d'acceptation :

- Pour réserver des billets sur internet, une personne (appelée acheteur) devra créer un compte en fournissant son adresse mel qui sera utilisée comme identifiant, et un mot de passe.
- 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 mel, 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 destinatrice d'un billet : son nom, son prénom, sa civilité, son sexe, son adresse mel, si elle est ou non malentendante, si elle est ou non malvoyante et sa date de naissance. La date de naissance est utilisée uniquement pour déterminer la tranche d'âge de la personne 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.

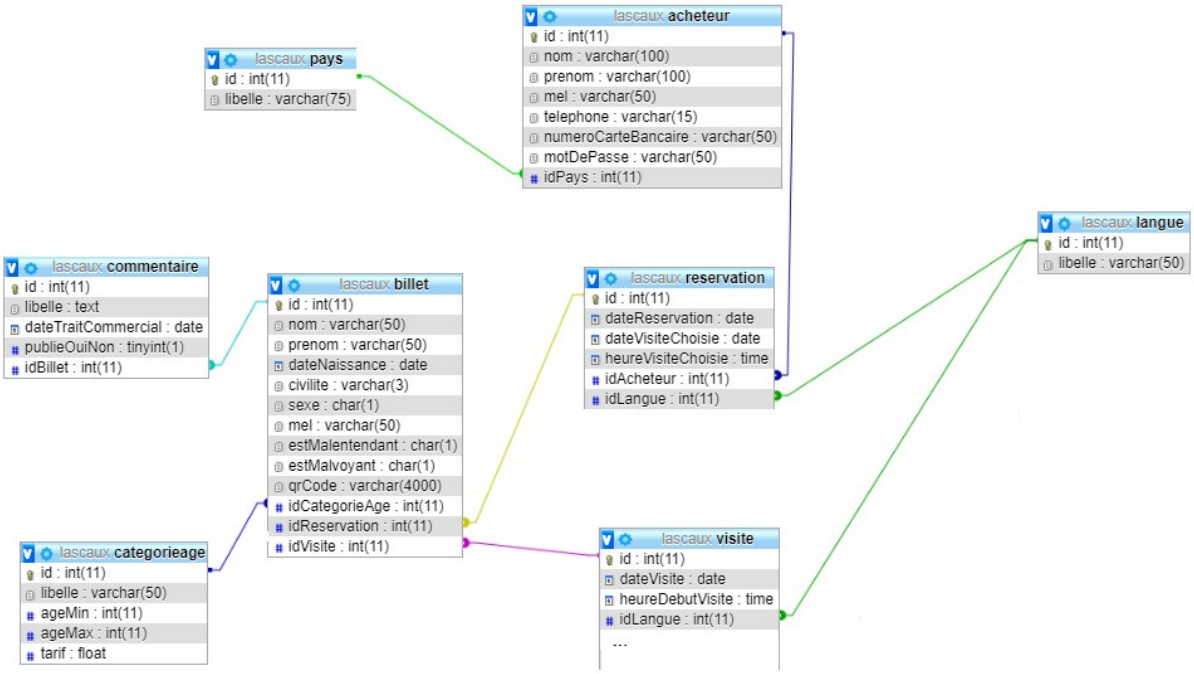
Document B2 : Représentation conceptuelle de la base de données

Représentation conceptuelle de la base de données avec un diagramme UML :



Document B3 : Extrait du schéma relationnel de la base de données utile aux missions 1 et 2

Extrait du schéma relationnel sous forme graphique :



Extrait du schéma relationnel sous forme textuelle :

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, motDePasse, 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

Document B4 : Contenu de la table Acheteur

id	nom	prenom	mel	telephone	numeroCarteBancaire	motDePasse	idPays
1	Tesure	Alice	alice.tesure@gmail.com	0600255636	5485889536529889-0625-999	alice60TESURE	1
2	Leduc	Jean	jean.leduc@gmail.com	0617736690	1235882231129889-0322-888	DUDU@458796	1

Document B7 : Évolutions demandées par le responsable de produit (PO) et le délégué à la protection des données (DPO)

Appréciation du niveau linguistique des guides

Lors de la première itération (*sprint*), une employée du service administratif a porté une appréciation sur le niveau de chaque langue parlée par un guide. Voici ci-contre un extrait des appréciations enregistrées dans la table LangueParlee.

idGuide	idLangue	niveau
1	1	Niveau satisfaisant
1	3	Niveau déplorable.Aucun effort pour comprendre !
2	2	Mauvais accent anglais. Difficile à comprendre.
2	3	Niveau satisfaisant
2	4	Doit impérativement se former car il très mauvais.

Au regard des valeurs saisies, le responsable de produit (PO) souhaite que le niveau de langue ne soit plus un texte libre, mais qu'il soit sélectionné dans une liste déroulante dont il gérera les valeurs via une fonctionnalité de l'application qui sera développée dans une prochaine itération (*sprint*).

Authentification et renouvellement des mots de passe

Pour des raisons de sécurité, le compte d'un acheteur doit être verrouillé après quatre échecs de connexion. Par ailleurs, les acheteurs devront changer leur mot de passe tous les quinze jours : le nouveau mot de passe choisi ne devra pas avoir déjà été utilisé par l'acheteur.

Les données sensibles

La connaissance du/des handicaps du visiteur (malentendance, malvoyance) permet de lui préparer un compagnon de visite (CDV) adapté. Lors de la première itération (*sprint*), ces indications ont été enregistrées à tort dans la base de données. Le délégué à la protection des données (DPO) vous demande de retirer ces informations sensibles et de proposer un moyen d'enregistrer pour le visiteur les caractéristiques du CDV à préparer. Plusieurs caractéristiques sont possibles :

- 1 : taille de la police de caractères moyenne ;
- 2 : taille de la police de caractères grande ;
- 3 : volume sonore élevé ;
- 4 : CDV clavier braille.

Document B8 : Fiche de registre établie par le délégué à la protection des données pour le traitement réservation de billet

Description du traitement	
Nom du traitement	Gestion des réservations
N° / RÉF	Ref-1005
Date de création du traitement	02/01/20
Mise à jour du traitement	

Acteurs	Nom	Adresse	Code Postal	Ville	Pays	Téléphone	Adresse mél
Responsable du traitement	Louise DUPONT	1, rue du musée	24290	Montignac	France	05 53 50 99 10	dupont.louise@lascaux.fr
Délégué à la protection des données	Martine DEPEO	1, rue du musée	24290	Montignac	France	05 53 50 99 12	depeo.martine@lascaux.fr
Société du DPO (si celui-ci est externe)							

Finalité(s) du traitement effectué	
Finalité principale	Réservation de billets
Sous-finalité 1	Édition du billet d'entrée
Sous-finalité 2	Planification des visites

Catégories de données personnelles concernées	Description	Durée de conservation
État civil, identité, données d'identification, images...	Pour l'acheteur : nom, prénom, téléphone,adresse mail, mot de passe Pour le visiteur : civilité, nom, prénom, adresse mail	Un an après la date de la dernière réservation
Vie personnelle (habitudes de vie, situation familiale, etc.)		
Informations d'ordre économique et financier (revenus, situation financière, situation fiscale, etc.)		
Données de connexion (adresse IP, logs, etc.)	Pour l'acheteur : logs de connexion	
Données de localisation (déplacements, données GPS, GSM, etc.)		
Numéro de Sécurité Sociale (ou NIR)		

Données sensibles	Description	Durée de conservation
Données révélant l'origine raciale ou ethnique	Aucune	
Données révélant les opinions politiques		
Données révélant les convictions religieuses ou philosophiques		
Données révélant l'appartenance syndicale		
Données génétiques		
Données biométriques aux fins d'identifier une personne physique de manière unique		
Données concernant la santé		
Données concernant la vie sexuelle ou l'orientation sexuelle		
Données relatives à des condamnations pénales ou infractions		

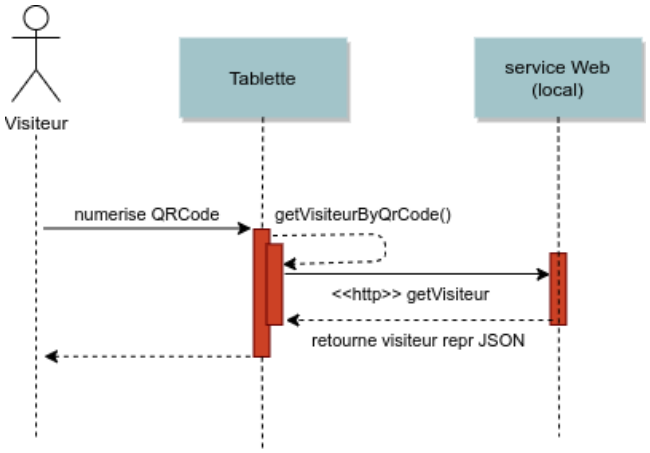
Catégories de personnes concernées	Description	Précisions

Documents associés au dossier C

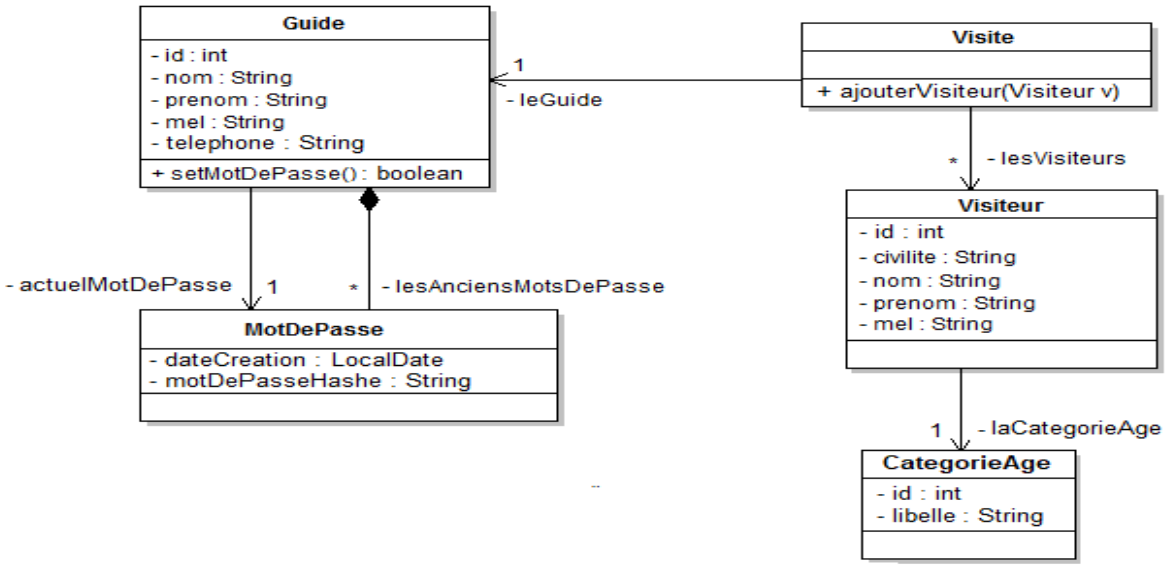
Document C1 : Exemple de billet produit après une réservation.

Ce billet a été attribué pour un adulte identifiable via le code QR (*QrCode)* figurant sur le billet. Le flashage de ce dernier permettra de déterminer l'identité du visiteur.

Document C2 : Diagramme de séquence de l'identification d'un visiteur par le code QR (QRCode) sur le compagnon de visite (CDV)



Document C3 : Diagramme de classes partiel de la partie identification



Remarque: les accesseurs des attributs ne sont pas représentés sur le diagramme.

Document C4a : Codes partiels d'identification des visiteurs : appel au service Web.

```
public class  UtilVisiteur {
    * Obtenir les caractéristiques du visiteur à partir du QrCode scanné
    * @param unQrCode chaîne contenant le code QR du billet
    * @return une référence à une instance de Visiteur
    * @throw  BadQrCodeException si unQrCode n'est pas reconnu
    */
    static public Visiteur getVisiteurByQrCode(String unQrCode) throws BadQrCodeException {
      // declenche la requete HTTP http://www.Lascaux.artparietal.fr/lascaux/getVisiteur.php
      SendHTTP requeteHttp = new SendHTTP();
      requeteHttp.addParam("qrcode",unQrCode);
      String json = requeteHttp.execute("http://" + serveur + chemin + "getVisiteur.php");
      // json contient la réponse du web service
      Visiteur visiteur =  UtilVisiteur.jsonStringToVisiteur(json);
      if (visiteur == null) {
          throw new BadQrCodeException(unQrCode) ;
      }
      return visiteur;
    }

    /**
    * Crée un objet visiteur à partir du flux json reçu en paramètre
    * @param jsonString chaîne Json en provenance du serveur
    * @return un visiteur correctement initialisé ou null si échec à identifier correctement un utilisateur
    */
    static private Visiteur jsonStringToVisiteur(String jsonString){
        Visiteur unVisiteur = null;
        String nomV,civiliteV, prenomV,melV, libCategorieAgeV;
        int idV, idCategorieAgeV;
        try {
            unVisiteur = new Visiteur();
            JSONObject objJson = new JSONObject(jsonString);
            unVisiteur.setId( Integer.parseInt(objJson.getString("id")) );
            unVisiteur.setCiviliteV( objJson.getString("civilite") );
            unVisiteur.setNomV( objJson.getString("nom") );
            unVisiteur.setPrenomV( objJson.getString("prenom") );
            unVisiteur.setMelV( objJson.getString("mel") );
            idCategorieAgeV = Integer.parseInt(objJson.getString("idCategorieAge"));
            libCategorieAgeV = objJson.getString("libCategorieAge");
            CategorieAge  laCategorieAge= new CategorieAge(idCategorieAgeV, libCategorieAgeV);
            unVisiteur.setLaCategorieAge( laCategorieAge ) ;
        }catch (JSONException e){
            // journalisation de l'événement douteux
            Log.d("log","pb decodage JSON");
        }
        return unVisiteur;
    }
} // class
```

L'appel de la méthode getString(String cle) de JSONObject déclenche une exception JSONException si aucune valeur n'est associée à la clé.

Document C4b: Preuve de concept (proof of concept) sous la forme d'un test unitaire qui détecte un manque de robustesse de la méthode jsonStringToVisiteur.

```
@Test
public void testVisiteurNonIdentifie() {
  String json =  "{ }";
  assertNull( UtilVisiteur.jsonStringToVisiteur(json) );
}
```

Document C5 : Codes partiels de l'application d'authentification : gestion des mots de passe du guide.

Guide.java

```
/** * Classe Guide*/
public class Guide {
        private int id;
        private String nom;
        private String prenom;
        private String mel;
        private String telephone;
        private MotDePasse actuelMotDePasse;
        private ArrayList <MotDePasse> lesAnciensMotsDePasse;
    /**
    * Ce constructeur valorise l'ensemble des attributs.
    */
        public Guide(int id, String nom, String prenom, String mel, String telephone,
              MotDePasse  actuelMotDePasse, ArrayList <MotDePasse> lesAnciensMotsDePasse ) {
                 ...
               this.lesAnciensMotsDePasse = lesAnciensMotDePasse;
               this.actuelMotDePasse = actuelMotDePasse;
        }
// Méthode à compléter pour être conforme à l'évolution demandée
     /**
    *  Vérifie si le nouveau mot de passe reçu en paramètre est conforme aux spécifications.
    *  Si le nouveau mot de passe est différent du mot de passe actuel, on mémorise l'instance de
    *  l'ancien mot de passe dans l'historique des anciens mot de passe, puis on crée une instance pour
    *  le nouveau mot de passe qui devient l'actuel mot de passe du guide (déjà hashé).
    * @param unMotDePasse le mot de passe hashé
    * @return true (vrai) si la modification a été effectuée ou false (faux) sinon
    */
        public boolean setMotDePasse(String unMotDePasse) {
                boolean modifARealiser = true;
                if(unMotDePasse.equals(this.actuelMotDePasse.getMotDePasseHashe()){
                        modifARealiser = false;
                } else{
                        this.lesAnciensMotsDePasse.add(this.actuelMotDePasse);
                        this.actuelMotDePasse = new MotDePasse( LocalDate.now(), unMotDePasse);
                }
                return modifARealiser;
        }
    // Méthode à écrire
     /**
     * Vérifie si le mot de passe est périmé, c'est-à-dire s'il a plus de trois mois
     * @return true (vrai) si le mot de passe a plus de trois mois, false (faux) sinon
     */
     public boolean doitChangerMdP() {...}
}
```

MotDePasse.java

```
/**
* Classe MotDePasse
* mémorise un mot de passe avec sa date de création. Le mot de passe est réputé hashé.
* @version 2.0
*/
public class MotDePasse {
	private LocalDate  dateCreation;
	private String motDePasseHashe;

	/**
	* constructeur de la classe
	* @param dateCrea date de début d'utilisation
	* @param motDePasse le mot de passe hashé
	*/
	public MotDePasse(LocalDate dateCrea, String motDePasse) {
		this.dateCreation = dateCrea;
		this.motDePasseHashe = motDePasse;
	}

	/**
	* accesseur de la date de création
	* @return la date de la création du mot de passe
	*/
	public LocalDate getDateCreation() {
		return dateCreation;
	}

	/**
	* accesseur du mot de passe
	* @return le mot de passe
	*/
	public String getMotDePasseHashe() {
		return motDePasseHashe;
	}
}
```

Document C6 : Scripts PHP de la phase d'identification du visiteur getVisiteur.php

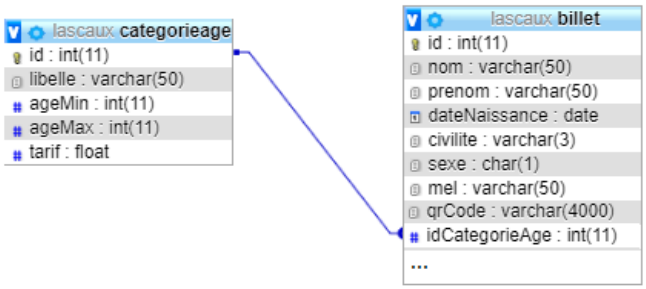
```php
<?php
// Le code du script de connexion a été testé et approuvé au niveau de la sécurité
// il initialise la variable globale $pdo
require_once 'connect-db.php';
/**
* Obtenir les informations du visiteur, en représentation JSON, à partir d'une valeur de qrCode d'un billet
* @param string $qrCode
* @global PDO $pdo une référence valide PDO de connexion à la base de données
* @return string JSON représentation du visiteur ou la cause de l'erreur
*/
function getVisiteurByQrCode($qrCode) {
  global $pdo;
  try {
    $sql = "
      SELECT billet.id as id, civilite, nom, prenom, mel,
                categorieAge.id as idCategorieAge, libelle AS libCategorieAge
      FROM Billet
      JOIN CategorieAge ON idCategorieAge = CategorieAge.id
      WHERE qrCode = '" . $qrCode . "'";
    $req = $pdo->prepare($sql);
    $req->execute();
    // demande la ligne résultante de l'exécution de la requête SQLen tant
    // qu'un tableau indexé par le nom des colonnes
    // ou retourne false si aucune ligne n'est retournée
    $ligne = $req->fetch(PDO::FETCH_ASSOC);
    return json_encode($ligne);

  } catch (PDOException $e) {
    $erreur = array();
    // obtient la raison technique de l'erreur
    $erreur['erreur'] =  $e->getMessage();
    return  json_encode($erreur);
  }
}
header("content-type: application/json; charset=utf-8");
echo getVisiteurByQrCode($_POST['qrcode']);
```

Exemple de réponse d'erreur en JSON

```
{
 "erreur" : "SQL Err..."
}
```

Remarque : La balise fermante php (?>) est volontairement absente en fin de script.

Document C7 : Extrait du schéma relationnel exploité par le script getVisiteur.php



Document C8 : Documentation partielle de la classe LocalDate en Java

```
//pour obtenir l'année en cours
LocalDate aujourdhui =  LocalDate.now();
int anneeEnCours = aujourdhui.getYear();
System.out.println(anneeEnCours);

// pour soustraire un mois à une date :
LocalDate uneDate = LocalDate.now();
LocalDate uneAutreDate ;
uneAutreDate = uneDate.minusMonths(1);

// pour soustraire une année à une date :
uneAutreDate = uneAutre.minusYears(1);

// pour ajouter un mois à une date :
uneAutreDate = uneAutreDate.plusMonths(1);

// Exemple de comparaison de dates :
if (aujourdhui.isAfter(uneAutreDate)) {
   System.out.println("aujourdhui se situe après l'autre date");
 } else {
    System.out.println("aujourdhui se situe avant l'autre date");
}
1
```

Document C9 : Filtrage des données externes (extrait d'une formation)

Les variables super globales $_POST, $_GET, $_COOKIE, et $_REQUEST détiennent des informations transmises via une requête *HTTP* déclenchée par un utilisateur.

Ces informations peuvent potentiellement provenir d'un utilisateur malintentionné. Il existe deux manières de filtrer : soit on valide le contenu de ces informations car on en connaît le type (URL, mél, tél, date par exemple), soit on nettoie le contenu en supprimant ou transformant ce qui pourrait être malicieux comme des apostrophes ou la présence de balises par exemple.

On utilise donc deux types de filtres :

- les filtres de validation (VALIDATE), qui ne modifient pas les données transmises, mais qui cherchent à vérifier qu'ils correspondent à un format. Si ce n'est pas le cas, ils renvoient FALSE.
- Les filtres de conversion (SANITIZE) qui suppriment ou transforment les caractères ou expressions qui ne sont pas conformes à un format sans forcément s'assurer que le résultat soit effectivement valide (par exemple suppression de balises et encodage de caractères spéciaux).

Quelques exemples :

- *pour valider un champ 'mel' contenant une adresse courriel via la méthode POST :*

```
$mail =  filter_input(INPUT_POST, 'mel', FILTER_VALIDATE_EMAIL);
if ($mail !== false) { /*ok on peut utiliser $mail*/ }
```

- *pour nettoyer un champ 'mdp' contenant un mot de passe via la méthode POST*

```
$mdpSaisi = filter_input(INPUT_POST, 'mdp', FILTER_SANITIZE_STRING);
if ($mdpSaisi) { /*ok on peut utiliser $mdpSaisi*/ }
```

Documents associés au dossier D

Document D1 : Liste de propositions émises lors de l'itération zéro (sprint 0)

1 Le compte administrateur de *MySQL* (*root*) sera utilisé par tous les scripts *PHP* de l'application ayant besoin de se connecter à la base de données *MySQL*. Il faudra cependant veiller à changer le mot de passe par défaut de ce compte.

2 L'administrateur *Scrum* (*Scrum Master*) définira l'identifiant et le mot de passe de l'administrateur de l'application en production. Ce dernier aura tous les droits sur le contenu des tables utilisées par l'application. Un fichier contenant l'identifiant et le mot de passe de l'administrateur sera remonté dans le référentiel (*repository*) utilisé par l'équipe *Scrum*.

3 Un référentiel (*repository*) privé sera utilisé pour la gestion de versions (*versioning*) des fichiers contenant le code source des applications.

4 Les développeurs pourront tester l'application en utilisant directement les bases de production.

5 Il est indispensable de vérifier qu'il n'y a pas de vulnérabilités connues sur les dernières versions des bibliothèques logicielles (*librairies*) et des *frameworks* libres (*open source*) avant de les utiliser.

6 Les caractéristiques de chaque connexion et déconnexion (user, IP, date et heure, etc.) seront conservées dans un fichier des événements (fichier *log*). L'écriture dans ce fichier ne sera effective que du lundi au vendredi.