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

U6 – CYBERSÉCURITÉ DES SERVICES INFORMATIQUES

SESSION 2023

_____

Durée : 4 heures
Coefficient : 4

_____

Matériel autorisé :
Aucun matériel ni document est autorisé.

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

Le sujet comporte 18 pages, numérotées de 1/18 à 18/18.

CAS Yak

BARÈME

DOSSIER A	Authentification et habilitations de l'application *Holy*	30 points
DOSSIER B	Fusion des bases de données	35 points
DOSSIER C	Amélioration de la sécurité des applications *Web*	15 points
	TOTAL	80 points

Présentation du contexte

Yak-à-Partir[1] est une agence de voyages créée en 2017 et spécialisée en prestations de voyage sur mesure. Ses mots d'ordre sont qualité de service et flexibilité pour des propositions de voyages personnalisés à travers le monde. Elle vise un public ne souhaitant pas suivre des programmes préétablis ou des circuits classiques de voyages organisés.

Société à responsabilité limitée située à A. en Bretagne, elle coordonne et organise les voyages à la demande de ses clients, qu'il s'agisse d'une nuit de découverte à proximité ou de plusieurs semaines au bout du monde. Sa clientèle peut formuler complètement ses souhaits de lieux, d'activités, de repas ou autre. L'agence se chargera de tout organiser.

À l'origine du projet, développeuse et passionnée de voyage, Mme Lenvy gère cette société avec deux salariés qu'elle emploie afin de l'assister dans la personnalisation qualitative exigée par ses clients.

En 2020, afin de compléter son activité, Mme Lenvy met en place une offre de coffret cadeau EchapBox. Cette offre, ciblant spécifiquement la région du Grand-Ouest, propose un moment de découverte en région, incluant un ensemble d'activités de la restauration à l'évasion sportive.

Yak-à-Partir possède deux applications *Web*, possédant chacune son nom de domaine :

- *Désir d'Ailleurs* (*Desir-dAilleurs.com*) permet aux intéressés de créer un compte puis de programmer un voyage personnalisé, bénéficier des conseils et du réseau de Mme Lenvy et de son équipe.
- *EchapBox* (*EchapBox.com*) permet d'acheter des coffrets de découverte, de s'inscrire via une création de compte, d'activer un coffret et de réserver une activité.

Chaque application *Web* dispose d'une partie administrateur permettant à la dirigeante et à son équipe le suivi et l'interaction avec les clients.

Holy, une application client lourd développée en langage *Java*, permet la gestion administrative et comptable de l'entreprise, notamment via la génération de contrats de vente et de prestation, de suivi des intervenants, d'édition comptable, etc.

Pour le développement, Mme Lenvy a développé elle-même les différentes applications, mais suite à l'augmentation d'actes de malveillance numérique affectant les PME d'une part, et d'autre part son implication grandissante dans la gestion de l'entreprise, elle décide de missionner un prestataire externe, Breizh-SN, pour s'occuper de l'aspect cybersécurité de sa société et la maintenance des applications. Après avoir fait un audit de sécurité, cette entreprise de services numériques accepte de reprendre et de maintenir le code source des applications précédemment développées.

Présentation de l'organisation prestataire Breizh-SN

Breizh-SN, labelisée « ExpertCyber[2] », réalise des applications *Web,* des applications en langage *Java*, majoritairement en architecture *Modèle-Vue-Contrôleur (MVC)* et supervise les systèmes informatiques de ses clients.

Membre de l'équipe cybersécurité de Breizh-SN, vous aurez en charge de vérifier et sécuriser les différentes applications de Yak-à-Partir.

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

[1] Pour des raisons de confidentialité, le nom des entreprises et les données afférentes ont été modifiés.

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

Dossier A – Authentification et habilitations de l'application *Holy*

Mission A1 – Mobilisation des bonnes pratiques pour la classe Utilisateur

Un des éléments clés de la cybersécurité étant une bonne hygiène de code, vous reprenez les conseils de bonnes pratiques pour améliorer l'application *Holy*. Vous constatez que la méthode *ancienMdp* de la classe Utilisateur n'est pas conforme aux règles de programmation *Java*.

Question A1.1

Corriger les erreurs de nommage présentes dans cette méthode.

Question A1.2

Proposer une documentation au format *Javadoc* pour cette méthode.

Mission A2 – Authentification : validation des mots de passe

Pour accéder à l'application *Holy*, une authentification des utilisateurs est en cours de développement. La classe Utilisateur contient des méthodes qui permettent de gérer le mot de passe actuel de l'utilisateur, mais aussi les précédents, avec des règles de sécurité pour ceux-ci.

Question A2.1

Identifier la complexité des mots de passe attendue lors de l'authentification des utilisateurs.

Question A2.2

Écrire le code de la méthode *modifierMdp* de la classe Utilisateur.

Mission A3 – Validation de l'authentification

La classe de tests unitaires UtilisateurTest n'est pas complète. Il manque une méthode de test permettant de valider le fonctionnement de la méthode *modifierMdp* de la classe Utilisateur.

Question A3.1

Ajouter une méthode de test *verifModifierMdp* pour compléter vos tests unitaires.

Face à l'augmentation constante des risques liés à la cybersécurité, Breizh-SN a proposé à sa cliente de réaliser une gestion des habilitations des utilisateurs restreignant l'accès aux éléments du menu de l'application.

Votre responsable vous charge de décrire à Mme Lenvy l'utilité de cette restriction, sous forme de scénarios de risques.

Question A3.2

Décrire un scénario de risque exploitant l'absence de restriction d'accès aux éléments du menu de l'application.

Mme Lenvy vous donne son accord pour réaliser cette restriction.

Question A3.3

a. Écrire le code de la méthode getNiveauHabilitation de la classe Utilisateur.

b. Compléter le code du constructeur de la classe AppliHoly.

Dossier B – Sécurisation de la fusion des bases de données

Les deux applications *Web* (*EchapBox* et *Désir d'Ailleurs*) sont implémentées séparément y compris au niveau des données gérées par le système de gestion de base de données (SGBD) *MySql*. Le choix de cette séparation a été fait lors du développement de l'activité EchapBox afin d'éviter de perturber l'activité Désir d'Ailleurs.

Mme Lenvy demande d'unifier les deux bases de données pour simplifier l'administration et la sécurisation des données des deux clientèles.

Mission B1 – Sécuriser les données personnelles

Pour bien comprendre les enjeux de cette fusion, vous avez mené un entretien avec Mme Lenvy.

Question B1.1

Réaliser le tableau demandé par Mme Lenvy durant l'entretien.

Suite à cet entretien avec Mme Lenvy, il est nécessaire de collecter le consentement des clients et de l'enregistrer dans la base de données *EchapBox* pour le démarchage commercial.

Pour cela, la table Client doit être modifiée pour accueillir un champ de type booléen appelé accordPubli ayant une valeur par défaut à FAUX.

Après la modification, la description de la table sera celle-ci :

EchapBox.Client(id, civilité, nom, prénom, dateNaiss, pseudo, mdp, adresse, codePostal, ville, pays, tél, mél, accordPubli)

La clé primaire est id.

Question B1.2

Écrire la requête permettant de modifier la table Client de la base de données *EchapBox*.

Vous avez développé un formulaire *Web* sur le site EchapBox.com disponible à l'adresse

https://www.EchapBox.com/accordPubli

Ce formulaire permet à un utilisateur de l'application *Web* EchapBox de donner son consentement pour le publipostage.

Si les utilisateurs ne désirent pas le donner, ils n'ont rien à faire, l'accord n'étant pas validé par défaut.

Ils ont un mois pour répondre au formulaire.

Vous devez maintenant rédiger un courriel aux utilisateurs pour les avertir de cette démarche en ligne en reprenant toutes les informations nécessaires pour obtenir leur consentement éclairé.

Question B1.3

Rédiger le corps du courriel destiné aux utilisateurs.

Pour être en conformité avec le RGPD, il faut conserver des preuves du recueil du consentement. Ces preuves prennent généralement la forme d'une documentation du processus de recueil du consentement. Pour documenter ce processus, il faut (entre autres) :

- garder une trace du processus ;
- garantir l'intégrité de cette trace.

Suite au recueil du consentement des utilisateurs d'EchapBox pour le démarchage commercial, il faudrait documenter ce processus pour se mettre en conformité.

Question B1.4

Proposer une solution détaillée permettant de conserver une trace du processus ainsi que l'intégrité de cette trace.

Mme Lenvy envisage de faire des statistiques sur les usages de ses clients, comme par exemple connaître les achats par genre, par tranche d'âge ou par département d'origine des clients.

L'ancienne base de données contient plus de données personnelles que nécessaire pour ce traitement. Il faut donc minimiser ces données afin de respecter le RGPD.

Les données retenues après minimisation seront stockées dans une nouvelle table nommée ClientAnonyme.

Question B1.5

Donner la structure de la nouvelle table ClientAnonyme en utilisant le formalisme du document B2.

Mission B2 – Détecter les agissements frauduleux

La fusion des bases est maintenant fonctionnelle, mais certains risques identifiés lors de l'audit de sécurité n'ont pas encore été traités.

Mme Lenvy réalise systématiquement un devis pour les voyages proposés au client. Si le client valide le devis, il paie un acompte et un contrat est ajouté dans la base de données. Le client recevra ensuite une facture du montant restant à payer (soit le montant à payer auquel est soustrait l'acompte versé).

Les risques suivants ont été identifiés :

- R1 : Un utilisateur pourrait profiter d'une faille de type injection SQL pour créer ou modifier un contrat avec un acompte supérieur ou égal au prix du contrat (montant à payer).

- R2 : Un utilisateur pourrait profiter d'une faille de type injection SQL pour créer ou modifier un contrat avec un montant à payer inférieur au montant minimum de 75 euros facturé par l'agence par personne et par jour.

Ces risques ont été placés dans une matrice d'analyse des risques EBIOS fournie dans le dossier documentaire.

Question B2.1

Justifier le niveau de gravité affecté aux risques R1 et R2, en décrivant l'impact que ces risques pourraient avoir sur l'entreprise.

Inquiétée par l'identification de ces risques, Mme Lenvy vous demande s'il y a un moyen de vérifier si certaines de ces failles ont déjà été exploitées.

Question B2.2

Écrire la requête permettant de visualiser la liste des clients (identifiant, nom et prénom) ayant un contrat avec un acompte versé supérieur ou égal au montant à payer.

Pour l'instant, un déclencheur (*trigger*) *before_insert_contrat_voyage* permet de vérifier certaines règles métier lors de l'ajout de contrats dans la base de données. Cependant, la règle imposant un montant minimal de 75 euros par participant et par jour pour un contrat n'est pas encore vérifiée.

Question B2.3

Compléter le code du déclencheur (*trigger*) en ajoutant la vérification de cette règle métier.

Dossier C – Amélioration de la sécurité des applications *Web*

Pour mieux sécuriser les applications *Web* de Yak-à-Partir, vous décidez de mener une analyse en profondeur pour identifier les failles les plus courantes, vérifier la protection existante, et mettre en place des contre-mesures.

Mission C1 – Vérifier la conformité de la protection contre une attaque CSRF

La documentation technique contient des extraits du code source permettant la modification du mot de passe utilisateur, utilisée dans les applications *Web*, ainsi qu'une description de la faille de contrefaçon de requête intersite (*cross-site request forgery* ou *CSRF*).

Question C1.1

Décrire le fonctionnement de la protection mise en place contre une attaque de type CSRF.

Mission C2 – Analyse des fichiers de journalisation

La sécurisation de l'application étant validée et mise en production, il vous appartient désormais de veiller au bon fonctionnement de l'application et d'intervenir lors d'évènements non souhaités. Des fichiers de journalisation sont mis en place afin d'aider à l'identification de bogues (*bugs*), incidents divers et menaces potentielles. Ces fichiers contiennent tous les événements notables générés par une application.

Question C2.1

a. Identifier tous les événements présents dans l'extrait du fichier de journalisation de l'application *Désir d'Ailleurs* du 10/05/2023.

b. Émettre une hypothèse sur l'origine de l'événement qui attire votre attention dans une courte note à destination de votre responsable.

Actuellement la table Client peut être décrite ainsi :

Client(idCli, civilité, nom, prénom, dateNaiss, pseudo, mdp, adresse, codePostal, ville, pays, tél, mél, numPièceIdentité, typePièceIdentité, nationalité, estAMobilitéRéduite)
La clé primaire est idCli.

Lorsqu'un compte est compromis, il faut le désactiver temporairement. Ceci laissera à l'administrateur le temps de mener des vérifications et, éventuellement, de réactiver le compte ou de prendre d'autres mesures.

Dans ce but, il faudrait référencer dans la base de données les tentatives de connexion des différents utilisateurs, chacun pouvant tenter plusieurs connexions. Une tentative n'est attribuée qu'à un seul utilisateur et possède un résultat (réussite ou échec).

Question C2.2

a. Proposer, dans le formalisme de votre choix, une évolution de la base de données permettant de répondre à cette demande.

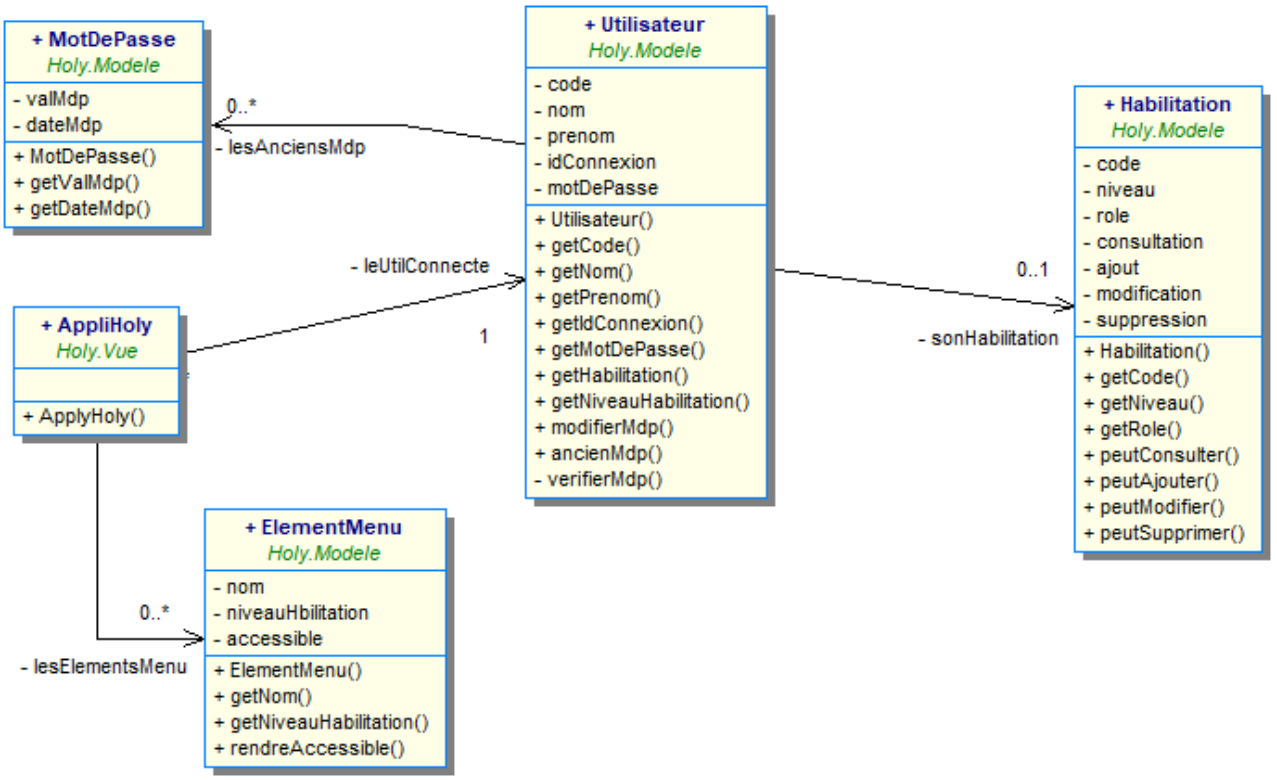
b. Donner deux enregistrements illustrant une tentative de connexion par un même utilisateur.

Question C2.3

Proposer et décrire, sans l'implémenter, une solution technique permettant de désactiver le compte.

Documents associés au dossier A

Document A1 : Extrait du diagramme de classes de l'application Holy



Document A2 : Extrait de la description de la classe technique Character

Character.isUpperCase(c)	Retourne *true* si le caractère c passé en paramètre est une majuscule, *false* sinon. Les caractères accentués sont pris en compte.
Character.isLowerCase(c)	Retourne *true* si le caractère c passé en paramètre est une minuscule, *false* sinon. Les caractères accentués sont pris en compte.
Character.isDigit(c)	Retourne *true* si le caractère c passé en paramètre est un chiffre, *false* sinon.

Document A3 : Extrait du code des classes métier

```
public class MotDePasse {
        private String valMdp;
        private LocalDate dateMdp;

        /** @param valMdp valeur du mot de passe
         * @param dateMdp date de création du mot de passe */
        public MotDePasse(String valMdp, LocalDate dateMdp) {
                this.valMdp = valMdp;
                this.dateMdp = dateMdp;
        }
        …
}
```

```
public class Habilitation {
	private String code, role;
	private int niveau;
	private boolean consultation, ajout, modification, suppression;
	/** Constructeur de la classe Habilitation
	 * @param code code de l'habilitation
	 * @param niveau niveau de l'habilitation
	 * @param role nom de l'habilitation
	 * @param consultation vrai si l'utilisateur peut consulter
	 * @param ajout vrai si l'utilisateur peut ajouter
	 * @param modification vrai si l'utilisateur peut modifier
	 * @param suppression vrai si l'utilisateur peut supprimer */
	public Habilitation(String code, int niveau, String role, boolean consultation, boolean ajout, boolean modification, boolean suppression) { … }

	public int getNiveau() { return niveau; }
	…
}
```

```
public class Utilisateur {
	private String code, nom, prenom, idConnexion, motDePasse;
	private Habilitation sonHabilitation;
	private ArrayList<MotDePasse> lesAnciensMdp = new ArrayList<MotDePasse>();

	/** Constructeur de la classe Utilisateur
	 * @param code code de l'utilisateur
	 * @param nom nom de l'utilisateur
	 * @param prenom prénom de l'utilisateur
	 * @param idConnexion identifiant de connexion à l'application Holy
	 * @param motDePasse mot de passe de connexion à l'application Holy
	 * @param habil habilitation de l'utilisateur pour l'application Holy */
	public Utilisateur(String code, String nom, String prenom, String idConnexion, String motDePasse, Habilitation habil) { … }

	public boolean ancienMdp(String m) {
		boolean existe = false;
		int i = 0;
		while (i < this.lesAnciensMdp.size() && existe == false) {
			if (this.lesAnciensMdp.get(i).getValMdp().equals(m)) {
				existe = true;
			}
			else {
				i = i + 1;
			}
		}
		return existe;
	}
```

```
        /** @return l'habilitation de l'utilisateur */
        public Habilitation getHabilitation() { return sonHabilitation; }

        /** Vérifie qu'un mot de passe est suffisamment complexe
         * @param mdp Le mot de passe à vérifier
         * @return vrai si le mot de passe respecte les règles de complexité */
        private boolean verifierMdp(String mdp) {
                boolean verif = false ;
                int nb1 = 0;
                int nb2 = 0;
                int nb3 = 0;
                int nb4 = 0;
                for (int i=0; i < mdp.length(); i = i + 1 ) {
                        char c = mdp.charAt(i);                 //Récupère le caractère situé à l'indice i
                        if (Character.isUpperCase(c)) { nb1 = nb1 + 1; }
                        else if (Character.isLowerCase(c)) { nb2 = nb2 + 1; }
                        else if (Character.isDigit(c)) { nb3 = nb3 + 1; }
                        else if (c >= 33 && c <= 46 || c == 64) { //plages de caractères spéciaux
                                nb4 = nb4 + 1;
                        }
                }
                if (mdp.length() >= 12 && nb1 >= 1 && nb2 >= 3 && nb3 >= 4 && nb4 >= 1) {
                        verif=true;
                }
                return verif;
        }

        /** @return le niveau de l'habilitation de l'utilisateur */
        public int getNiveauHabilitation() {
                /* A COMPLÉTER SUR VOTRE COPIE */
        }
        /** Vérifie que le nouveau mot de passe passé en paramètre
         * répond aux règles de complexité et qu'il ne fait pas partie des anciens mots de passe.
         * Si les vérifications sont correctes le mot de passe actuel est enregistré comme
         * ancien mot de passe avec la date du jour obtenue par LocalDate.Now(),
         * puis le mot de passe actuel est modifié.
         * @param valMdp nouveau mot de passe de l'utilisateur
         * @return vrai si la modification du mot de passe s'est bien passée, faux sinon */
        public boolean modifierMdp(String valMdp) {
                /* A COMPLÉTER SUR VOTRE COPIE */
        }
…
}
```

```
public class ElementMenu {
        private String nom;
        private int niveauHabilitation;
        private boolean accessible = false;

        /** Constructeur de la classe ElementMenu
         * @param nom le libellé du menu
         * @param niveauHabilitation Niveau d'habilitation minimum requis
         *                                  pour accéder à cet élément du menu        */
        public ElementMenu(String nom, int niveauHabilitation) { … }

        public int getNiveauHabilitation() { return niveauHabilitation; }

        public void rendreAccessible() { accessible = true; }

        …
}
```

```
/** {@summary Formulaire principal de l'application suite à une authentification réussie} */
public class AppliHoly extends JFrame {
    // déclaration de tous les composants graphiques du formulaire
    // et des éléments du menu
    private ArrayList<ElementMenu> lesElementsMenu;
    //...

    // objet permettant de conserver l'utilisateur connecté
    private Utilisateur leUtilConnecte;

    /** Constructeur du formulaire **/
    public AppliHoly(Utilisateur unUtil) throws HeadlessException {

        // instanciation de tous les composants graphiques du formulaire
        // y compris tous les éléments du menu de l'application (code non fourni)
        …

        leUtilConnecte = unUtil;

        // seuls les éléments du menu ayant un niveau d'habilitation inférieur ou égal à celui de l'utilisateur connecté doivent être rendus accessibles
        /* A COMPLÉTER SUR VOTRE COPIE */
    }
}
```

Document A4 : Code d'une classe de test unitaire de la classe Utilisateur

```
class UtilisateurTest {
    Utilisateur unUtilisateur;

    @BeforeEach // initialisation avant chaque test
    void init() {
        Habilitation uneHabilitation = new Habilitation("ma01", 1, "master", true, false, true, false);
        unUtilisateur = new Utilisateur("U001", "Durand", "Louis", "lodurand", "Coe8@MatH279",
uneHabilitation);
        unUtilisateur.modifierMdp("Lae99_Mat00!");
        unUtilisateur.modifierMdp("M1ue@uiT455n");
    }

    @Test // définit une méthode de test unitaire
    void verifHabilitation() {
        assertTrue("Erreur sur le droit de lecture",unUtilisateur.getHabilitation().peutConsulter());
        assertFalse("Erreur sur le droit d'ajout",unUtilisateur.getHabilitation().peutAjouter());
        assertTrue("Erreur sur le droit de modification",unUtilisateur.getHabilitation().peutModifier());
        assertFalse("Erreur sur le droit de suppression",
                                                    unUtilisateur.getHabilitation().peutSupprimer());
    }
    /* A COMPLÉTER SUR VOTRE COPIE */
}
```

Document A5 : Extrait des règles de programmation Java (bonnes pratiques)

Type	**Règles**	**Exemples**
Classes	La première lettre prend une majuscule. Mélange de minuscules et majuscules avec la première lettre de chaque mot en majuscule. Donner des noms simples et descriptifs. Éviter les acronymes : hormis ceux commun (XML, URL, HTML, etc.) N'utiliser que les caractères [a-z] et [A-Z] et [0-9] : ne pas utiliser de caractères spéciaux (-, _, $, *, accents, etc.).	`class Image` `class ClientPrivilegie`
Interfaces	Mêmes règles que pour les classes.	`interface ImgAccesDirect` `interface Stockage`

Méthodes	La première lettre est en minuscule.
	Les noms de méthodes doivent refléter une action. Choisir de préférence des verbes.
	Une méthode doit posséder une documentation JavaDoc : rôle de la méthode, nature des paramètres et de la valeur de retour.
		`afficher()`
		`getValue()`
		`setValue()`

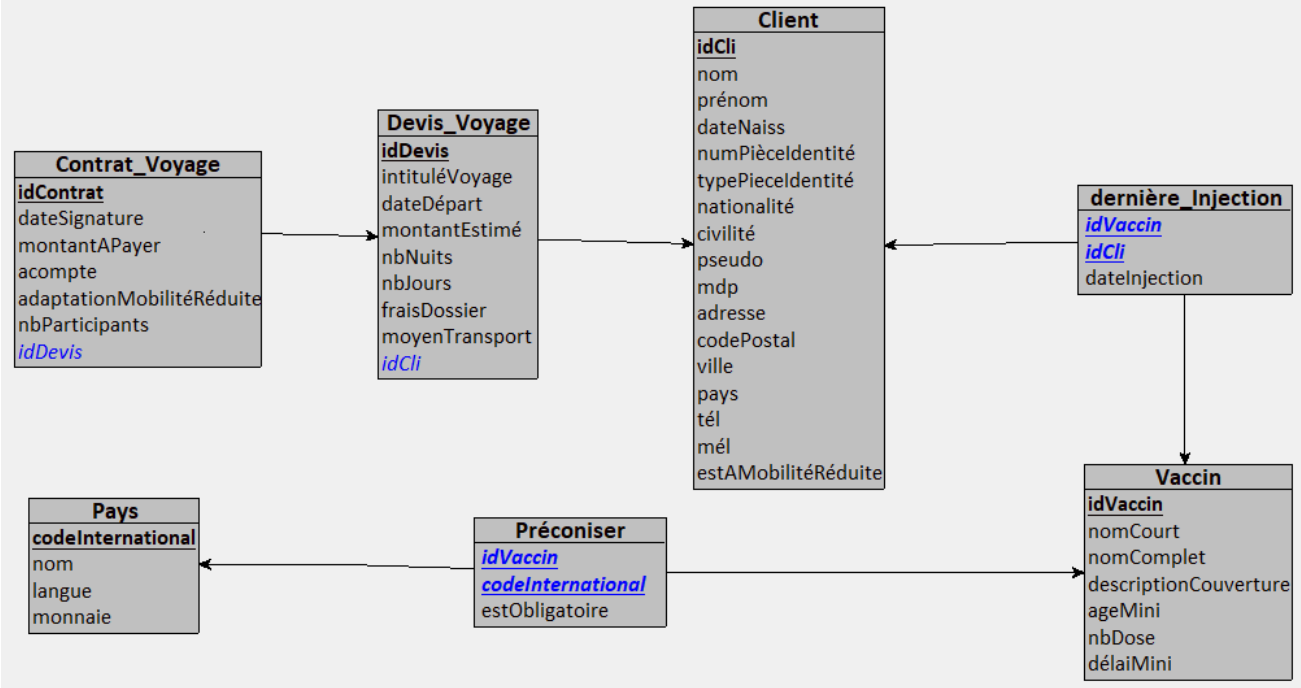
Variables	La première lettre est en minuscule.
	Mélange de minuscules et majuscules avec la première lettre de chaque mot en majuscule.
	Donner des noms simples et descriptifs.
	Ne pas commencer les noms avec '$' ou '_' (bien que ce soit possible).
	
	On n'utilisera une variable dont le nom est composé d'une seule lettre que pour un usage local :
	int : i, j et k
	char : c, d, et e
	boolean : b
	
	De plus, d'une manière générale :
	N'utiliser que les caractères [a-z] et [A-Z] et [0-9] : ne pas utiliser de caractères spéciaux (-, _, $, *, accents, ...).
		`int i;`
		`float largeur;`
		`String nomDuCapitaine;`

Constantes	En majuscules. Le séparateur devient donc forcément un souligné (_).	`final static int VAL_MAX = 999;`
Commentaires	Bloc de commentaires /* …. */ Commentaire sur une ligne //	

Source : extrait loribel.com

Documents associés au dossier B

Document B1 : Extrait du schéma relationnel de la base de données de l'application Désir d'Ailleurs



Document B2 : Description de la table Client de la base de données EchapBox

EchapBox.Client(id, civilité, nom, prénom, dateNaiss, pseudo, mdp, adresse, codePostal, ville, pays, tél, mél)
Clef primaire : id

Document B3 : Extrait du compte-rendu de l'entretien avec Mme Lenvy

Mme Lenvy (ML) : Les deux applications *Désir d'Ailleurs* et *EchapBox* utilisent la même technologie avec de la programmation spécifique à chaque projet et des bases de données différentes. Nous voulons donc fusionner les bases de données des deux applications pour simplifier la gestion des serveurs, notamment leur sécurisation, et optimiser nos coûts.

Vous : Je vois. Du point de vue du RGPD, cela implique un nouveau traitement, celui de la fusion des bases. La finalité de ce traitement est compatible avec les finalités précédentes et vous êtes dans votre intérêt légitime. Allez-vous réaliser d'autres traitements sur ces données ?

ML : Oh, non, nous n'allons pas réaliser de nouveaux traitements, les applications fonctionneront toujours sur les mêmes données, rien ne changera. D'ailleurs, la nouvelle base ne sera pas très différente des anciennes, c'est surtout la table Client qui sera impactée.

Vous : D'accord. Il faudra bien penser à compléter le registre des traitements, en identifiant les données personnelles et les données sensibles collectées.

ML : Oui, mais c'est à vous de faire cela non ? J'ai demandé à votre entreprise d'assumer le rôle de délégué à la protection des données de Yak-à-Partir. Vous pouvez commencer par réaliser un tableau qui présente les données personnelles et les données sensibles existantes dans chacune des deux bases de données.

Vous : Très bien ! Et vous êtes sûre de ne pas utiliser ces données pour autre chose, en dehors des applications ?

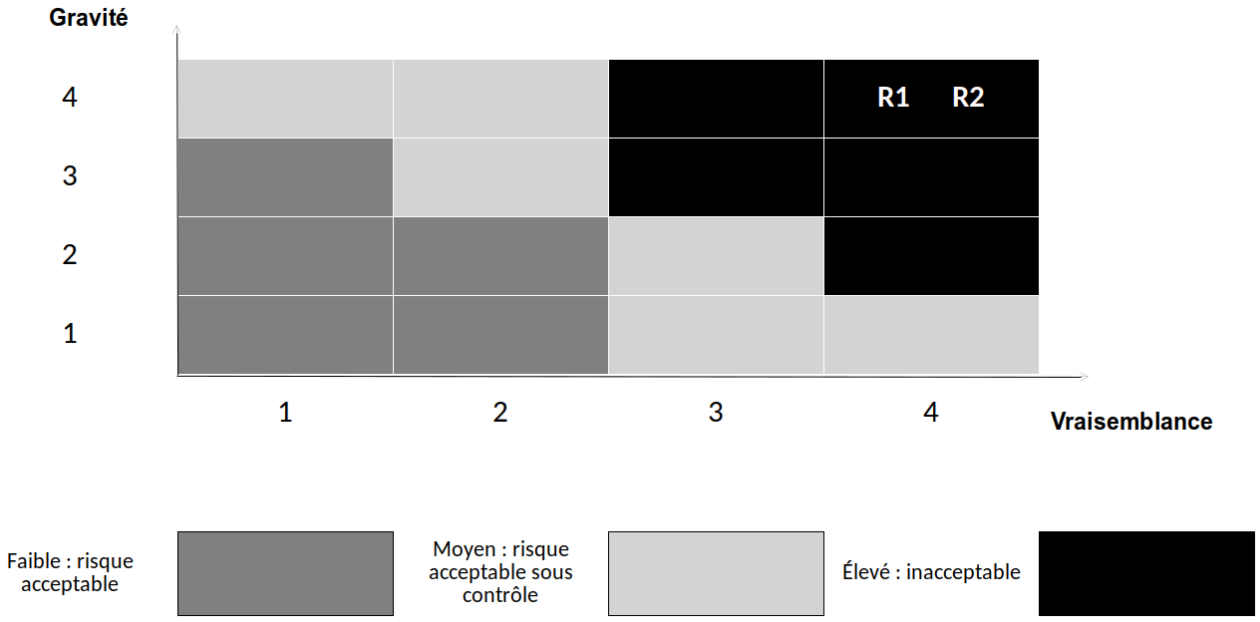
ML : Ah si ! Nous faisons aussi du démarchage commercial par publipostage auprès des clients de Désir d'Ailleurs.Et vu comme notre outil est configuré, le publipostage s'appliquerait aussi aux clients d'EchapBox après la fusion des bases.

Vous : Ah, ce serait un problème : vous seriez dans un cas de détournement de finalités, ce qui n'est pas autorisé. Il faudrait obligatoirement obtenir le consentement des clients d'EchapBox avant de commencer votre démarchage commercial auprès d'eux. Pour les clients de Désir d'Ailleurs, rien ne change. Il faudra tout de même que vous informiez tous les utilisateurs d'EchapBox de la fusion des bases, tout en leur rappelant leurs droits.

Document B4 : Extrait de mémento SQL

ALTER TABLE <*nom_table*>
{ **ADD** <*définition attribut*>
| **ALTER** <*nom_attribut*> { **DEFAULT** <*valeur*> | **DROP DEFAULT** }
| **DROP** <*nom_attribut*> [**CASCADE** | **RESTRICT**]
| **ADD** <*définition contrainte*>
| **DROP CONSTRAINT** <*nom_contrainte*> [**CASCADE** | **RESTRICT**] };

Document B5 : Matrice d'analyse de risques de la méthode EBIOS



Document B6 : Code du déclencheur (trigger) à compléter

```
-- Création d'un trigger agissant avant l'insertion dans Contrat_Voyage
-- définition du caractère indiquant la fin du déclencheur
DELIMITER |
CREATE TRIGGER before_insert_contrat_voyage BEFORE INSERT
ON Contrat_Voyage FOR EACH ROW
BEGIN
-- SET permet d'assigner une valeur à une variable
-- Le mot-clé NEW fait référence à l'enregistrement en cours d'insertion.
-- NEW.idDevis est la valeur d'idDevis qu'on insére dans Contrat_Voyage.
SET @nb_jours = ( SELECT nbJours FROM Devis_Voyage
                        WHERE idDevis = NEW.idDevis);
IF @nb_jours < 3 THEN
    -- On déclenche une erreur car la règle métier n'est pas respectée
    SIGNAL SQLSTATE '10001' ;
    -- On précise le message d'erreur
    SET MESSAGE_TEXT = 'Le devis validé par le contrat est d\'une durée
inférieure à la durée minimale autorisée' ;
END IF;
--------- A COMPLÉTER SUR VOTRE COPIE ----------
END |
```

Documents associés au dossier C

Document C1 : Extrait du fichier de journalisation de l'application Désir d'Ailleurs du 10/05/2023

```
...
[21:41:18] NOTICE: utilisateur AllanG : erreur de connexion
[21:41:20] WARNING: utilisateur : essai de connexion avec champs vides
[21:41:24] NOTICE: utilisateur RichardP : erreur de connexion
[21:41:24] NOTICE: utilisateur RichardP : erreur de connexion
[21:41:24] NOTICE: utilisateur RichardP : erreur de connexion
[21:41:24] NOTICE: utilisateur RichardP : erreur de connexion
[21:41:24] NOTICE: utilisateur RichardP : erreur de connexion
[21:41:26] NOTICE: utilisateur AllanG : s'est connecté correctement
[21:41:26] NOTICE: utilisateur RichardP : erreur de connexion
[21:41:26] NOTICE: utilisateur RichardP : erreur de connexion
[21:41:26] NOTICE: utilisateur RichardP : erreur de connexion
...
```

Document C2 : Description de la faille CSRF (cross site request forgery)

CSRF (*cross site request forgery*) [...] est un mode d'escroquerie courant sur internet. Les criminels prennent le contrôle d'une session d'un utilisateur [...] et peuvent ainsi exécuter des actions malveillantes. Celles-ci passent par le biais de requêtes HTTP.

Exemple : un utilisateur est légitimement authentifié à une plateforme en ligne. Si, à la fin de sa connexion, l'utilisateur oublie de se déconnecter, il reste ainsi connecté jusqu'à la fin de la période prévue par le site (*timeout*), sans devoir saisir à nouveau son mot de passe. [...]

Il ne reste plus qu'à piéger cet utilisateur sur un site contrefait ou un lien alléchant transmis par courriel (*phishing*) pour pousser l'utilisateur à cliquer sur une action qu'il pense inoffensive sur le site contrefait ou via le courriel.

L'action ainsi déclenchée envoie alors une requête HTTP à la plateforme utilisée précédemment par l'utilisateur et usurpe ainsi son identité pour exécuter une action malveillante pendant que sa session est encore active. [...]

Le serveur de la plateforme ciblée reconnaît la formulation d'une requête HTTP d'origine légitime (l'utilisateur authentifié) et utilise les témoins de connexion (*cookies*) correspondants pour confirmer que l'utilisateur (c'est-à-dire son navigateur) est encore connecté. Le serveur exécute l'action et il se peut que l'utilisateur ne remarque pas qu'une action a été exécutée en son nom.

Source : d'après ionos.fr

Document C3 : Extrait du code source permettant la modification du mot de passe utilisateur

Extrait du fichier modifMdp.php

```
<?php
// Démarrage de la session en début de chaque page
session_start();
$token = $_SESSION['token'];
?>
<!DOCTYPE html>
<html lang="fr">
<head><meta charset="UTF-8"><title>Modification mot de passe</title></head>
<body>
  <form method="POST" action="traitement.php?action=modifMdp">
     <div class="input-div">
       <input type=" password" name="pwd" id="pwd" required>
       <label for="pwd" class="label-name">
         <img src="img/lock.svg">Nouveau mot de passe</label>
     </div>
     <div class="input-div">
       <input type="password" name="confirmPwd" id="confirmPwd" required>
       <label for="confirmPwd" class="label-name">
         <img src="img/lock.svg">Confirmation mot de passe</label>
     </div>
<input type="hidden" name="token" id="token" value="<?php echo $token; ?>" />
     <button type="submit" name="connect-btn">Se connecter</button>
  </form>
</body>
</html>
```

Extrait du fichier traitement.php

```
<?php
session_start();
if (!empty($_SESSION['token']) AND !empty($_POST['token'])
     AND $_SESSION['token'] == $_POST['token']) {

    // le traitement demandé s'effectue ici
}
else {
    echo "Erreur de vérification";
}
?>
```

La méthode ***empty*** détermine si une variable est considérée comme vide. Une variable est considérée comme vide si elle n'existe pas, ou si sa valeur équivaut à false.