UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Licence 2 Informatique
INFO0303 – 2022/2023
***Resp. :* C. Rabat**

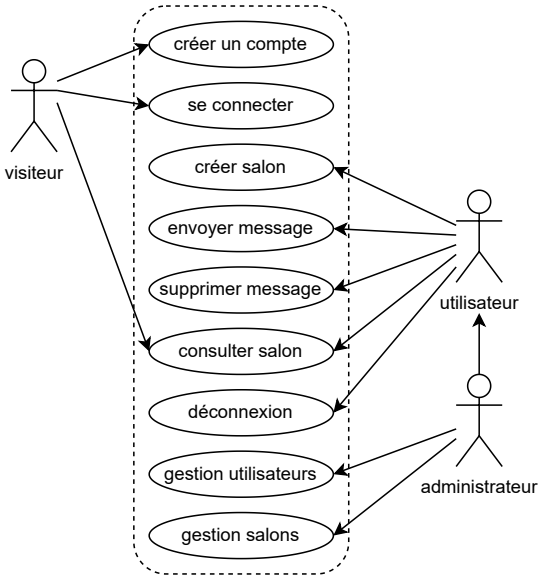
Devoir surveillé terminal

Durée 1h30
(Corrections)

Nous souhaitons développer une application de forums de discussion en PHP. Les utilisateurs peuvent créer des salons, publiques ou privés, et les autres utilisateurs peuvent y écrire des messages. Le créateur d'un salon en est le modérateur : il peut supprimer les messages des autres utilisateurs. Si un salon est privé, le créateur choisit les utilisateurs qui peuvent y accéder. L'administrateur du site a la possibilité de supprimer des salons et gère les utilisateurs. Lorsqu'un visiteur arrive sur le site, il peut uniquement consulter les salons publiques (sans pouvoir écrire de message).

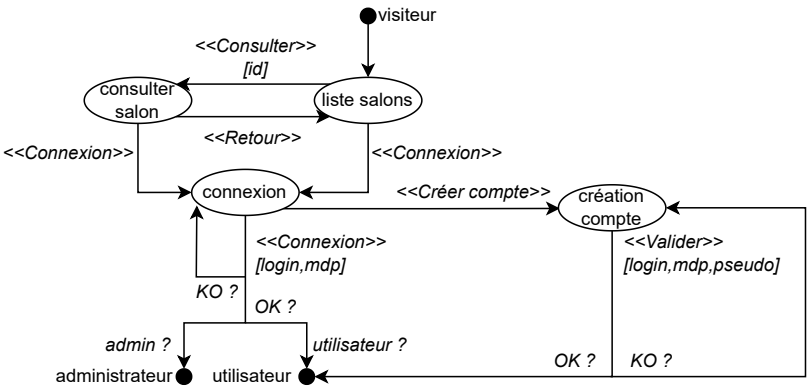
Exercice 1 (Modélisation - *6 pts*)
1°) Identifiez les acteurs du site et donnez les cas d'utilisation de l'application. *(1 point)*

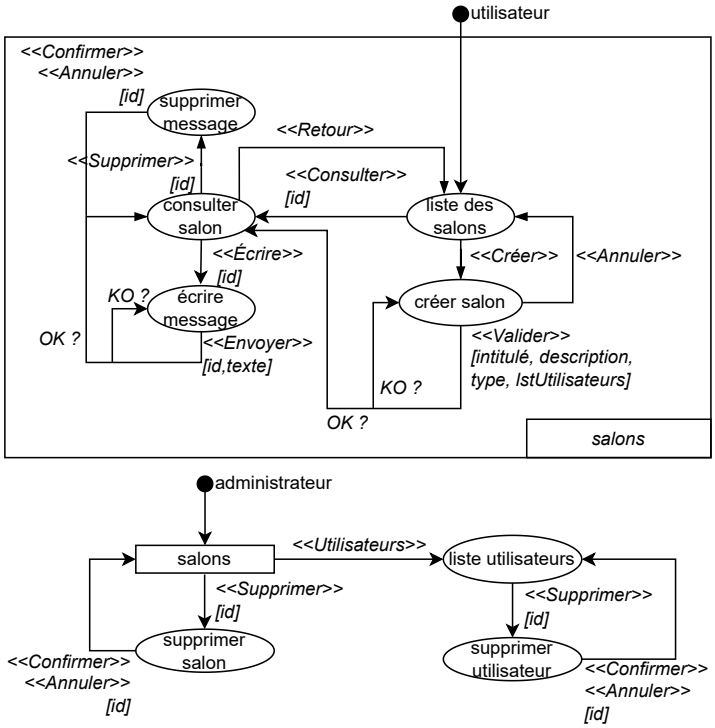
Solution : *Il y a 3 acteurs : le visiteur, l'utilisateur (un visiteur enregistré) et l'administrateur.*



2°) Donnez les diagrammes de navigation de l'application. *(3 points)*

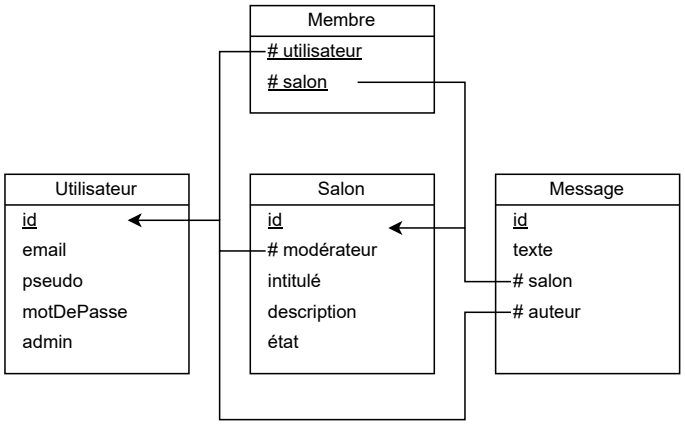
Solution :





3°) Quelles seraient les données nécessaires dans la base de données ? Proposez un MLD et expliquez vos choix. *(2 points)*

Solution : *Les utilisateurs sont caractérisés par leur adresse email, leur pseudo et leur mot de passe, ainsi qu'un type (administrateur ou non). Un salon est caractérisé par un intitulé, une description (un texte d'explication), un état (privée ou non) et par l'utilisateur qui l'a créé (pour connaître le modérateur de la discussion). Les utilisateurs peuvent également être associés à un salon s'il est privé. Les messages des utilisateurs sont caractérisés par un texte, l'identifiant de la discussion et l'identifiant de l'utilisateur qui l'a écrit. On peut y ajouter une date pour les trier.*



Exercice 2 (Programmation PHP - *6 pts*)

Dans un premier temps, nous faisons le choix de développer l'application en PHP sans *framework*.

1°) Proposez une classe `Salon` permettant de représenter un salon avec les méthodes classiques (constructeur, getters, conversion en chaîne de caractères). *(2 points)*

Solution :

```
class Salon {
  private int $id;
  private int $moderateur;
  private string $intitule;
```

```
  private string $description;
  private bool $etat;

  public function __construct(int $id, int $moderateur, string $intitule, string
      $description, bool $etat) {
    $this->id = $id;
    $this->moderateur = $moderateur;
    $this->intitule = $intitule;
    $this->description = $description;
    $this->etat = $etat;
  }

  public function getId() : int { return $this->id; }
  public function getModerateur() : int { return $this->moderateur; }
  public function getIntitule() : string { return $this->intitule; }
  public function getDescription() : string { return $this->description; }
  public function getEtat() : bool { return $this->etat; }

  public function __toString() : string {
    return $this->intitule;
  }
}
```

2°) Proposez un formulaire permettant à l'utilisateur de saisir les données d'un salon (pour le moment, sans spécifier les utilisateurs dans le cas d'un salon privé). *(1 point)*

Solution :

```
...
<form method="post" action="dest.php">
  <label for="intitule">Intitulé</label><input type="text" name="intitule"/>
  <label for="description">Description</label>
  <textarea name="description"></textarea>
  <label for="privee">privé</label>
  <input type="radio" name="etat" value="prive" id="prive">
  <label for="publique">publique</label>
  <input type="radio" name="etat" value="publique" id="publique">
  <button type="submit">Valider</button>
</form>
...
```

3°) Comment faire pour associer le salon en cours de création et son créateur (l'utilisateur courant) ? *(1 point)*

Solution : *Lors de la connexion, on enregistre l'identifiant de l'utilisateur en session, par exemple dans la variable `$_SESSION['current_user']`. Lors de la création du salon, on peut alors lui associer cet identifiant.*

4°) Donnez le contenu du script PHP permettant de récupérer les données depuis le formulaire (en ajoutant les vérifications nécessaires) et de créer un objet `Salon`. *(1 point)*

Solution :

```
<?php
session_start();
include("Salon.php");

if(isset($_POST['intitule'])) $intitule = $_POST['intitule'];
if(isset($_POST['description'])) $description = $_POST['description'];
$type = isset($_POST['type']) && ($_POST['type'] == "prive");
$salon = new Salon(-1, $_SESSION['current_user'], $intitule, $description, $type);
...
```

5°) Comment permettre au créateur de spécifier, dans la page de création, les utilisateurs qui ont accès à la discussion (dans le cas où elle est privée) ? Comment récupérer les utilisateurs sélectionnés dans le script PHP ? *(1 point)*

Solution : *On peut utiliser une liste déroulante contenant les utilisateurs. Dès qu'un utilisateur le sélectionne, on l'ajoute dans le formulaire, dans un champ caché, par exemple. Le nom du champ peut être un tableau, ce qui permet de lui ajouter les utilisateurs. En PHP, il suffit ensuite de récupérer le tableau avec les identifiants qu'il contient. En JavaScript, on peut afficher/masquer la liste en fonction du bouton radio sélectionné (privé ou non).*

Exercice 3 (*Laravel - 8 pts*)
1°) Qu'est-ce qu'une table pivot ? Donnez un exemple. *(2 points)*

Solution : *Dans le cas d'une relation n-n entre deux entités (tables) de la base de données, il est nécessaire de créer une table intermédiaire qui permet de faire le lien entre les deux entités. Par exemple, si on prend une actualité qui possède des mot-clés, la table pivot possède deux champs news_id et keyword_id. Le premier est une clé étrangère vers l'identifiant de l'actualité et le second vers l'identifiant du mot clé.*

2°) Comment faut-il procéder avec *Laravel* pour créer une table pivot ? *(1 point)*

Solution : *Il faut d'abord créer une migration pour créer la table. Celle-ci crée la table dont le nom est composé du nom des deux tables liées séparés par un _, avec les deux champs et les clés étrangères vers les deux autres tables.*

3°) Qu'est-ce qu'un modèle dans *Laravel* ? *(1,5 point)*

Solution : *Un modèle permet de manipuler les enregistrements d'une table et d'assurer la persistance. Il est possible de spécifier les champs de modifiables dans un formulaire. Des méthodes peuvent être ajoutées lorsque l'on a des relations avec d'autres éléments (comme dans le cas d'une table pivot).*

4°) Lors de la création d'une table pivot, est-il nécessaire de créer un nouveau modèle ? Est-ce qu'il faut modifier les modèles existants ? *(1,5 point)*

Solution : *Le modèle est nécessaire seulement si des données supplémentaires sont associées. Par exemple, si on souhaite ajouter une date de création associée à la relation entre une actualité et un mot-clé. Les modèles existants peuvent être modifiés si on souhaite ajouter des méthodes pour récupérer les informations liées. Par exemple, si on souhaite récupérer les mots-clés d'une actualité ou bien les actualités associées à un mot-clé.*

Finalement, nous choisissons d'utiliser *Laravel* pour développer l'application précédente.

5°) Quels éléments doivent être mis en place dans *Laravel* pour réaliser l'application ? Expliquez chacun d'entre eux. *(2 points)*

Solution : *Il faut :*

- *Les migrations : une pour chaque table ; permet de créer les tables*
- *Les contrôleurs : un contrôleur pour gérer chaque élément (salons, messages, utilisateurs) ; permettent de gérer les accès, de récupérer les données depuis la base et d'appeler les vues*
- *Les modèles : salon, messages, utilisateur ; permettent de manipuler les données de la base*
- *Les vues : une pour chaque page et un ou plusieurs templates (pour les données communes) ; mise en forme, affichage des données, formulaires*
- *Les requêtes : pour vérifier automatiquement les données saisies (ajout/modification)*
- *Les seeders/factories : uniquement pour créer l'administrateur (ou pour générer des données de test)*

On peut parler également de la configuration de l'application avec le fichier `.env` *(configuration du nom de l'application, de l'accès à la base de données).*