
CCP**Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité****Modalités d'évaluation des compétences et organisation de l'épreuve**

Modalités	Compétences évaluées	Durée	Détail de l'organisation de l'épreuve
Présentation d'un projet réalisé en amont de la session	Développer la partie back-end d'une interface utilisateur web Développer des composants d'accès aux données Développer une interface utilisateur de type desktop Développer la partie front-end d'une interface utilisateur web Maquetter une application	00 h 30 min	En amont de la session, le candidat a réalisé un projet en entreprise concernant le développement d'une interface utilisateur et a préparé un dossier de projet ainsi qu'un support de présentation. La structure et le contenu de ces documents figurent au paragraphe « Informations complémentaires concernant la présentation d'un projet réalisé en amont de la session ». Au moins dix jours ouvrés avant la session d'examen, le candidat a remis à l'organisateur un résumé en français de son projet. Au moins un jour ouvré avant la session d'examen, le candidat a remis à l'organisateur son dossier de projet. Lors de la session d'examen, le candidat commence sa présentation au jury par un résumé en anglais de son projet. Il présente ensuite son projet à l'aide d'un support de présentation.
Autres modalités d'évaluation le cas échéant :			
▪ Entretien technique	Développer la partie back-end d'une interface utilisateur web Développer des composants d'accès aux données Développer une interface utilisateur de type desktop Développer la partie front-end d'une interface utilisateur web Maquetter une application	00 h 20 min	Le jury questionne le candidat sur la base de son dossier de projet et de sa présentation orale. Un questionnement complémentaire lui permet d'évaluer les compétences qui ne sont pas couvertes par le projet.
▪ Questionnaire professionnel	Sans objet		Sans objet

SIGLE	Type de document	Code titre	Millésime	Date dernier JO	Date de mise à jour	Page
CDA	RC	TP-01281	03	17/04/2018	19/04/2018	19/38

Modalités	Compétences évaluées	Durée	Détail de l'organisation de l'épreuve
▪ Questionnement à partir de production(s)	Sans objet		Sans objet
Durée totale de l'épreuve pour le candidat :			00 h 50 min

PLATEFORME **TICKETING**

STEVE HOAREAU - CDA6

SOMMAIRE

- 
- 
- 01** PRÉSENTATION DU PROJET EN ANGLAIS
 - 02** EXPRESSION DES BESOINS
 - 03** CONCEPTION DU PROJET
 - 04** RÉALISATIONS AVEC EXTRAITS DE CODES ET DÉMONSTRATIONS
 - 05** PRÉSENTATION DU JEU D'ESSAI
 - 06** VEILLE DE VULNÉRABILITÉ
 - 07** CONCLUSION

IN ENGLISH



A company requires an online ticket management platform that enables various departments to create tickets for tasks and monitor their progress.



The platform is secured through an authentication system and developed in PHP using the Symfony framework for the back-end, along with daisyUI and Tailwind CSS utilities for the front-end.

EXPRESSION DES BESOINS

Objectif n° 1

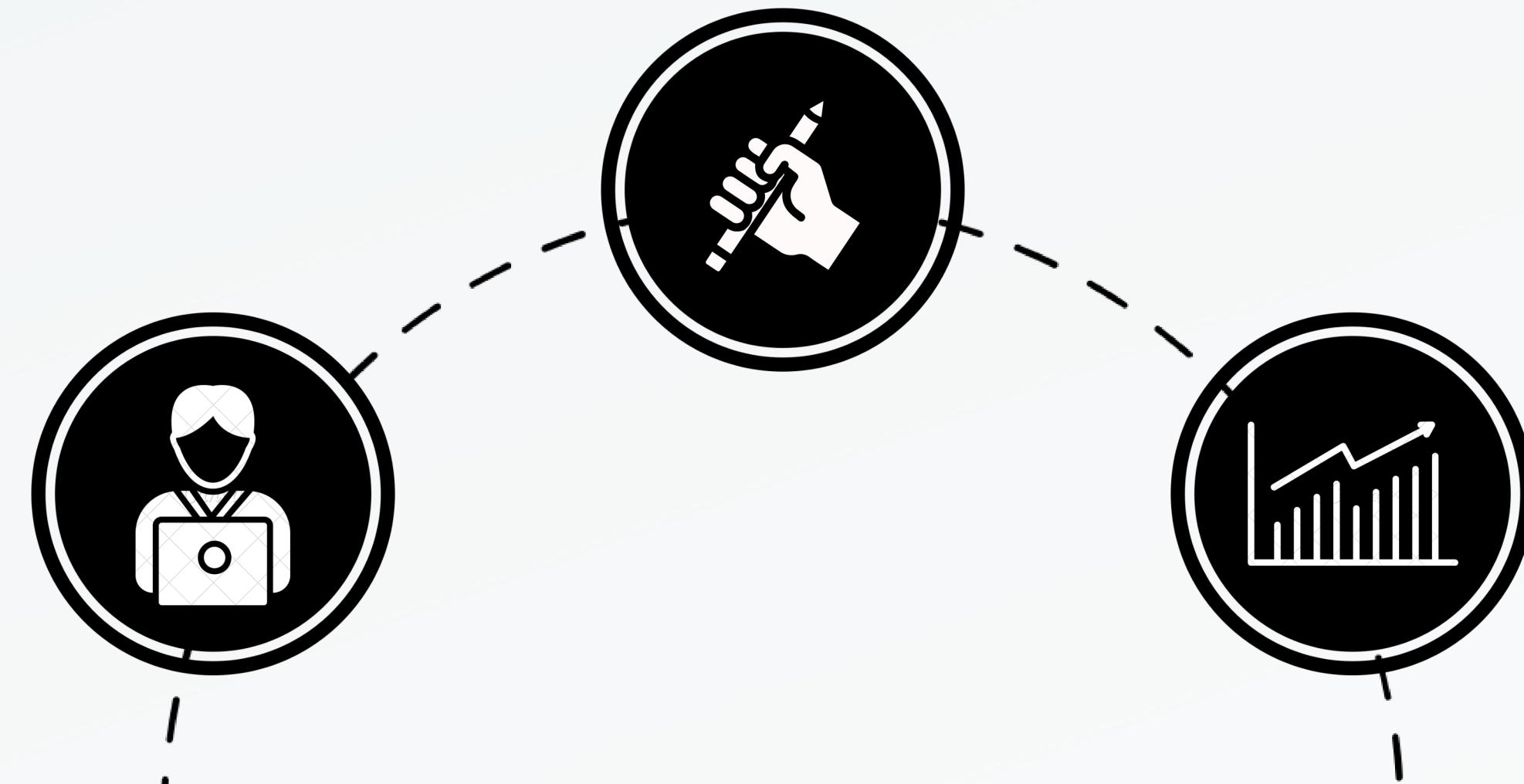
La plateforme a une page de connexion

Objectif n° 2

Création, Suivi et Prise en charge d'un ticket par service

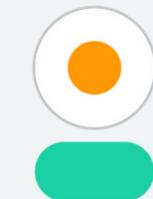
Objectif n° 3

Tableau de bord avec chiffres clés pour les administrateurs et gestion des utilisateurs



SPÉCIFICATIONS TECHNIQUES DU PROJET

Développement



Apport hors cahier des charges

- Fonctionnalités administrateur = Fonctionnalités utilisateur
 - Statistiques par utilisateur pour les admins
 - Réinitialisation de mot de passe par mail et le mot de passe est généré automatiquement

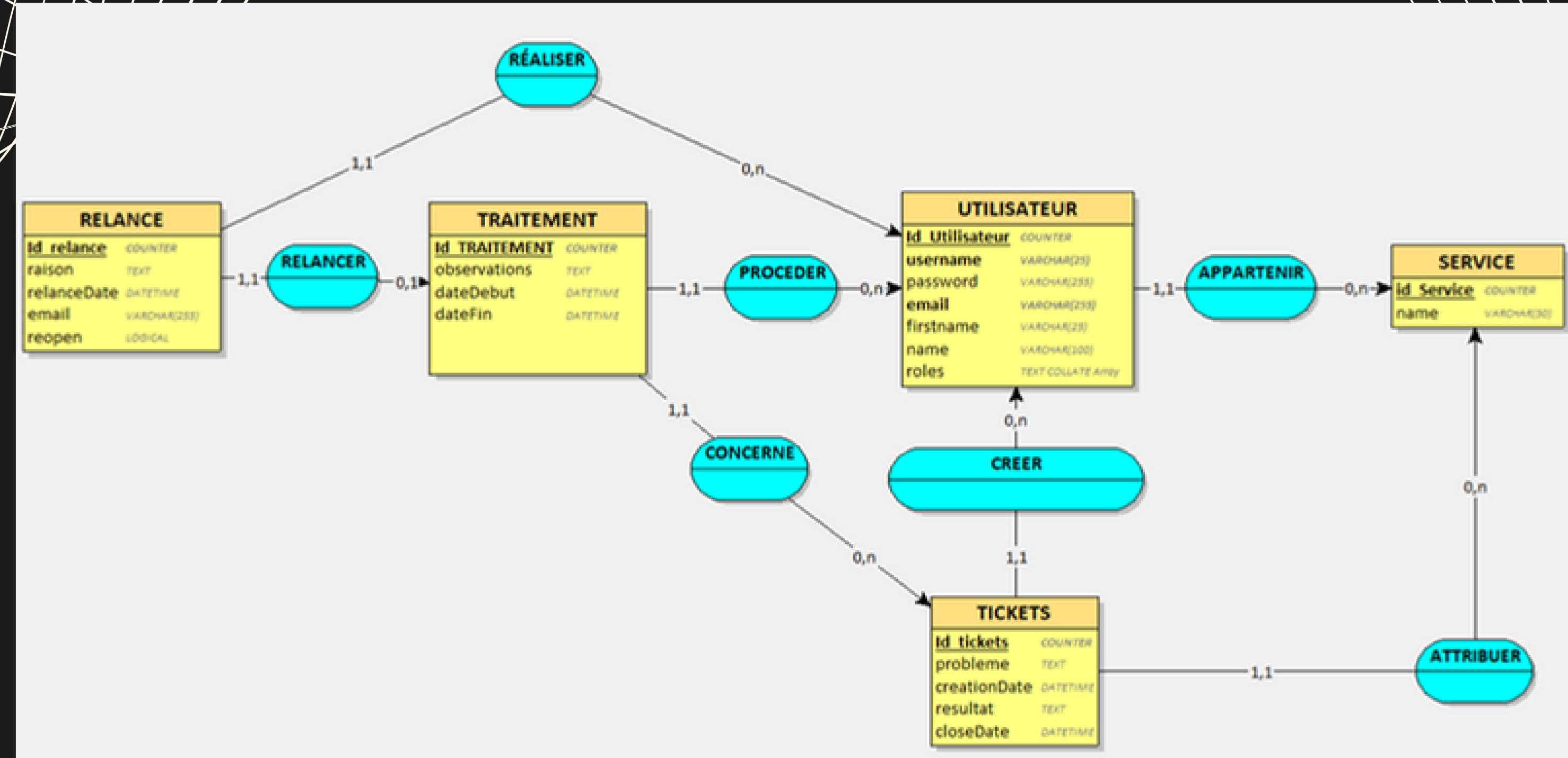
Sécurité

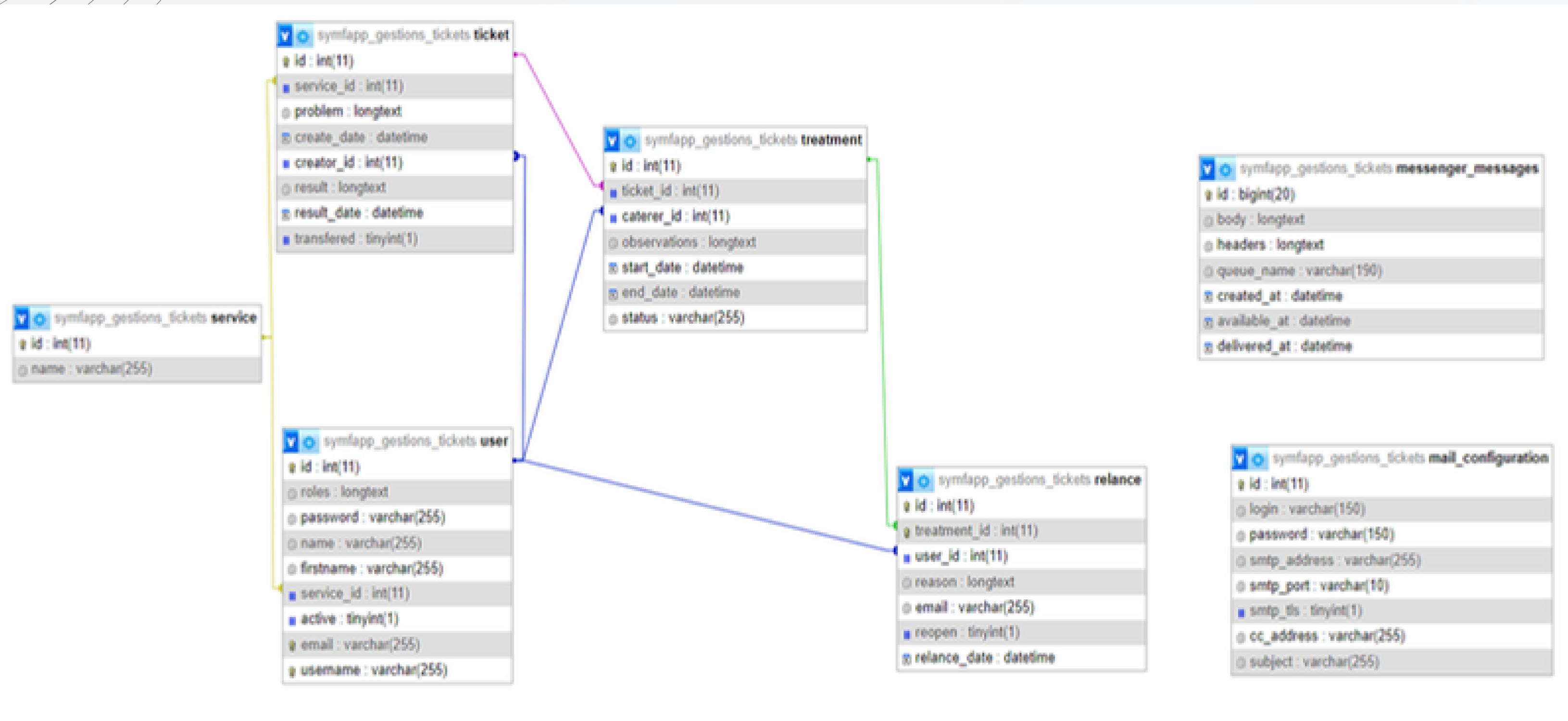
- Les mots de passe sont forts et cryptés avec bCrypt
- Utilisation de CSRF Token

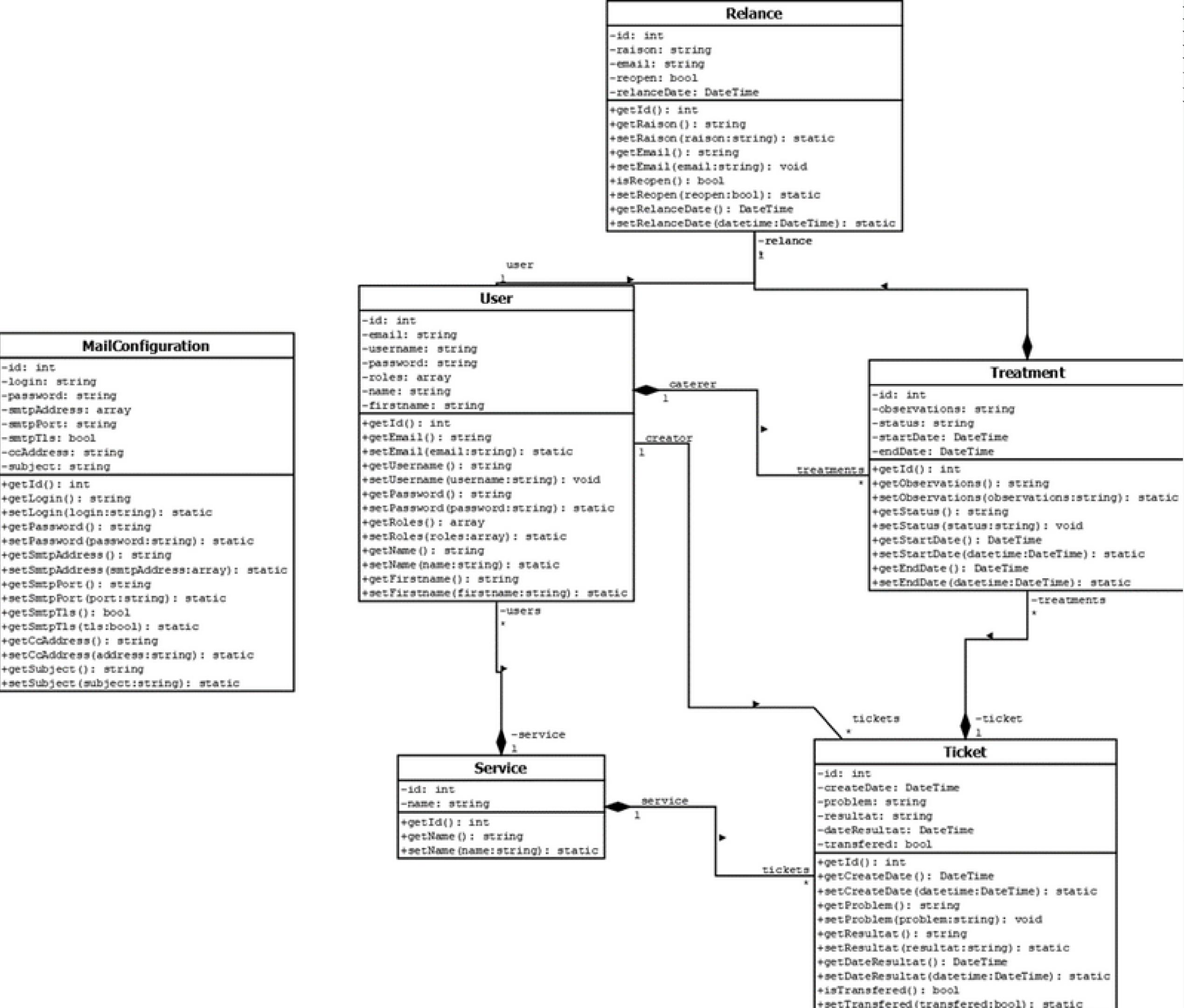


CONCEPTION DU PROJET

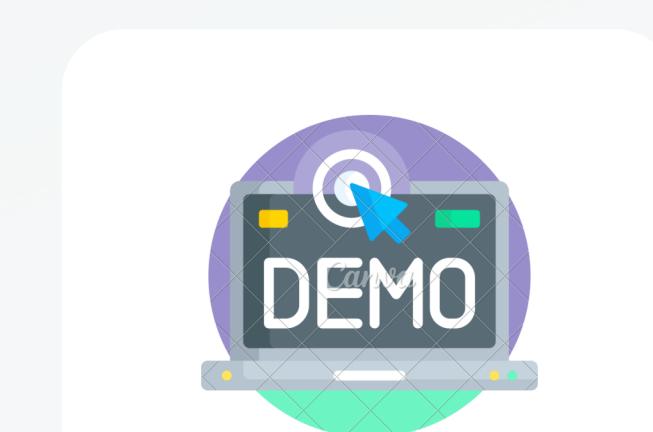
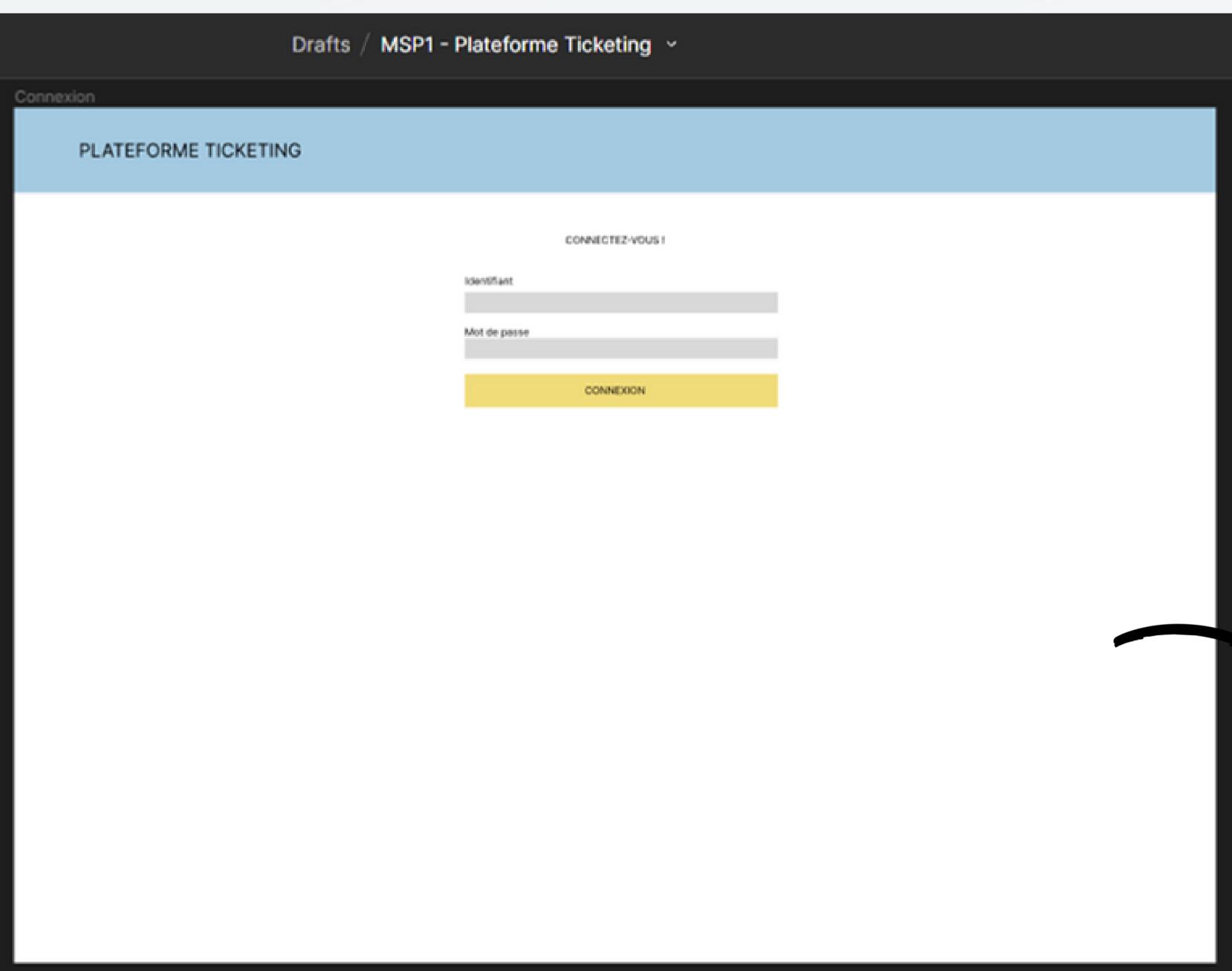








RÉALISATIONS AVEC EXTRAITS DE CODES ET DÉMONSTRATIONS



PAGE DE CONNEXION

```
class AppAuthenticator extends AbstractLoginFormAuthenticator
{
    use TargetPathTrait;

    public const LOGIN_ROUTE = 'app_login';

    public function __construct(private UrlGeneratorInterface $urlGenerator)
    {
    }

    /**
     * Fonction dans l'interface qui permet de créer un passport avec l'utilisateur, son mot de passe et un jeton de sécurité
     * @see AuthenticatorInterface
     */
    public function authenticate(Request $request): Passport
    {
        $username = $request->request->get('username', '');

        $request->getSession()->set(Security::LAST_USERNAME, $username);

        return new Passport(
            new UserBadge($username),
            new PasswordCredentials($request->request->get('password', '')),
            [
                new CsrfTokenBadge('authenticate', $request->request->get('_csrf_token')),
                new RememberMeBadge(),
            ]
        );
    }
}
```

```
class AppAuthenticator extends AbstractLoginFormAuthenticator
{
    use TargetPathTrait;

    public const LOGIN_ROUTE = 'app_login';

    public function __construct(private UrlGeneratorInterface $urlGenerator)
    {
    }

    /**
     * Fonction dans l'interface qui permet de créer un passport avec l'utilisateur, son mot de passe et un jeton de sécurité
     * @see AuthenticatorInterface
     */
    public function authenticate(Request $request): Passport
    {
        $username = $request->request->get('username', '');

        $request->getSession()->set(Security::LAST_USERNAME, $username);

        return new Passport(
            new UserBadge($username),
            new PasswordCredentials($request->request->get('password', '')),
            [
                new CsrfTokenBadge('authenticate', $request->request->get('_csrf_token')),
                new RememberMeBadge(),
            ]
        );
    }
}
```

DASHBOARD ADMIN

Drafts / MSP1 - Plateforme Ticketing

Connexion administrateur

PLATEFORME TICKETING (admin)

hsteve

Tableau de bord administratif

SERVICE DÉVELOPPEMENT

SERVICE TECHNIQUE

SERVICE COMMERCIAL

hsteve

OUVERT FERME EN ATTENTE

OUVERT FERME EN ATTENTE

OUVERT FERME EN ATTENTE

63 tickets sont fermés
8 tickets sont en cours
5 tickets tickets sont en attente

63 tickets sont fermés
8 tickets sont en cours
5 tickets tickets sont en attente

63 tickets sont fermés
8 tickets sont en cours
5 tickets tickets sont en attente

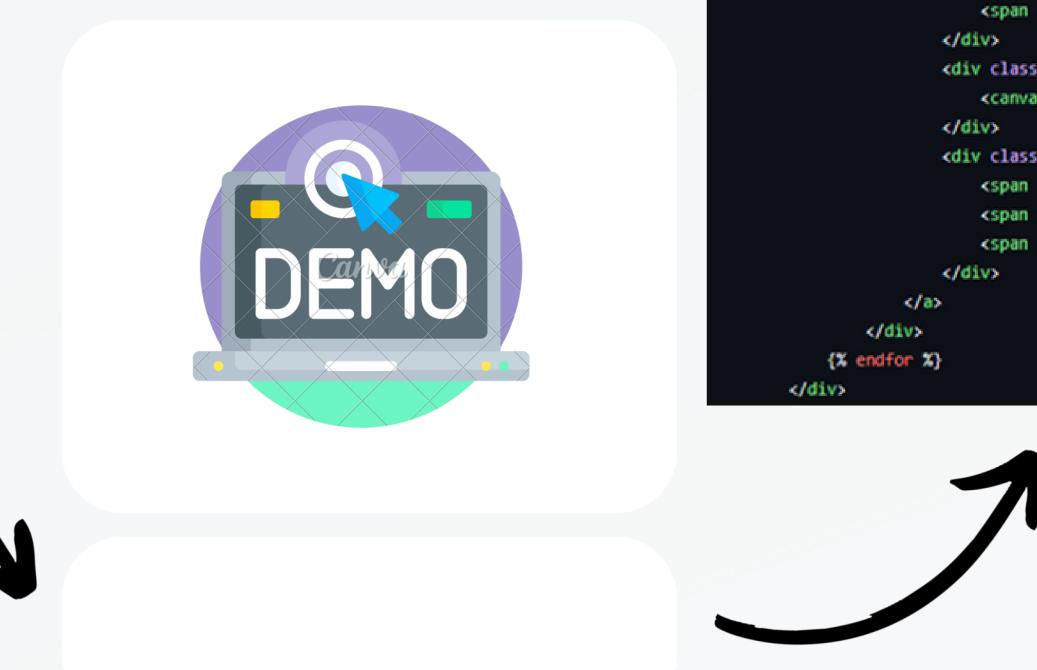
VISER LES TICKETS DE MON SERVICE

GESTION DES UTILISATEURS

PARAMETRE MAIL

The dashboard displays three circular charts representing ticket status for Service Développement, Service Technique, and Service Commercial. Each chart has three segments: yellow (Open), green (Closed), and red (On Hold). Below each chart is a list of ticket counts for each status. Navigation buttons at the bottom include 'Voir les tickets de mon service', 'Gestion des utilisateurs', and 'Paramètre mail'.

```
(% if "ROLE_ADMIN" in app.user.roles %)
  (# Mode admin #)
  (% if dashboard == true %)
    (# Affichage du dashboard #)
    <h1 class="text-center font-extrabold m-3">Tableau de bord administratif</h1>
    <div class="grid grid-cols-3 gap-4">
      (% for service in serviceList %)(# on parcours la liste des service donné en paramètre lors du rendu de la vue depuis le controller)
        <div class="bg-gray-300 rounded-lg p-4">(# Possibilité de cliquer pour avoir + d'informations)
          <a href="{{ path('app_historic_by_service',{'service':service.id}) }}">(◍ Possibilité de cliquer pour avoir + d'informations)
            <div class="text-center">
              <span class="badge badge-secondary">{{ service.name }}</span>
            </div>
            <div class="chart-container" style="position: relative; height:40vh; width:80vw; margin-left: 20px">(◍ canvas qui se génère après le chargement du contenu)
              <canvas id="canvasService{{ service.id }}"></canvas>(◍ canvas qui se génère après le chargement du contenu)
            </div>
            <div class="mt-1 grid grid-cols-3 justify-center justify-items-center">
              <span id="waitingBadgeIdService{{ service.id }}" class="badge badge-error">-</span>(◍ Nombre de tickets en attente)
              <span id="inProgressBadgeIdService{{ service.id }}" class="badge badge-info">-</span>(◍ Nombre de tickets en cours)
              <span id="closeBadgeIdService{{ service.id }}" class="badge badge-success">-</span>(◍ Nombre de tickets fermés)
            </div>
          </a>
        </div>
      (% endfor %)
    </div>
```



```
(# if "ROLE_ADMIN" in app.user.roles #)
    (#      Mode admin #)
    (# if dashboard == true #)
        (#          Affichage du dashboard #)
        <h1 class="text-center font-extrabold m-3">Tableau de bord administratif</h1>
        <div class="grid grid-cols-3 gap-4">
            (# for service in serviceList #){# on parcours la liste des service donné en paramètre lors du rendu de la vue depuis le controller
                <div class="bg-gray-300 rounded-lg p-4">
                    <a href="{{ path('app_historic_by_service',{'service':service.id}) }}" # Possibilité de cliquer pour avoir + d'informations sur ce service
                        <div class="text-center">
                            <span class="badge badge-secondary">{{ service.name }}</span>
                        </div>
                        <div class="chart-container" style="position: relative; height:40vh; width:80vw; margin-left: 20%">
                            <canvas id="canvaService{{ service.id }}"></canvas> (# canvas qui se génère après le chargement du contenu
                        </div>
                        <div class="mt-1 grid grid-cols-3 justify-center justify-items-center">
                            <span id="waitingBadgeIdService{{ service.id }}" class="badge badge-error">-</span>(# Nombre de tickets en attente)
                            <span id="inProgressBadgeIdService{{ service.id }}" class="badge badge-info">-</span>(# Nombre de tickets en cours)
                            <span id="closeBadgeIdService{{ service.id }}" class="badge badge-success">-</span>(# Nombre de tickets fermés)
                        </div>
                    </a>
                </div>
            (# endfor #)
        </div>
```

PAGE UTILISATEUR

Drafts / MSP1 - Plateforme Ticketing

Connexion utilisateur - Dashboard

PLATEFORME TICKETING

user

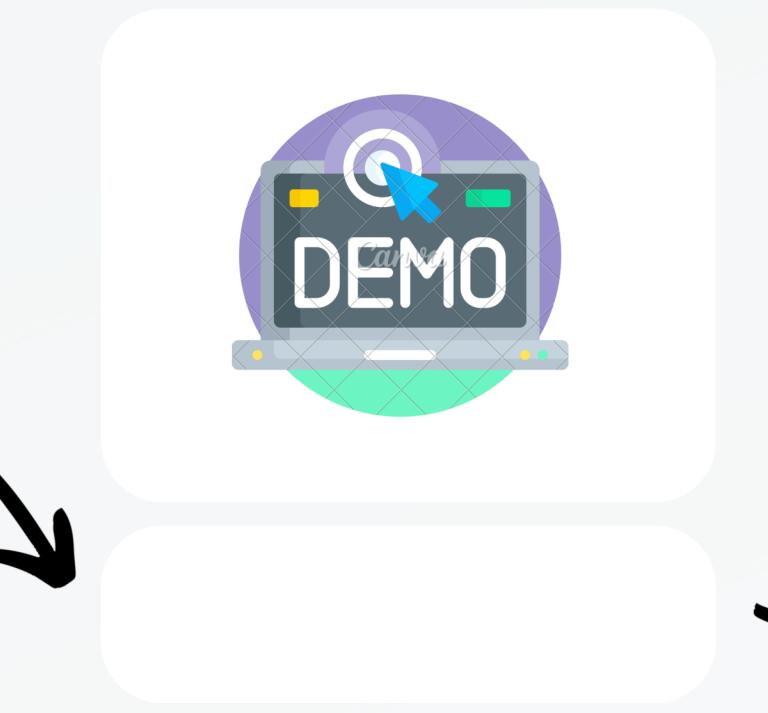
CREER UN TICKET VOIR LES TICKETS DE MON SERVICE VOIR LES TICKETS QUE J'AI CREER

SERVICE COMMERCIAL

TICKET T#4
EN ATTENTE OUVERT LE 01/09/2023 12h23 TICKET T#4
Créer par Doe John
Traitements: 0
VOIR

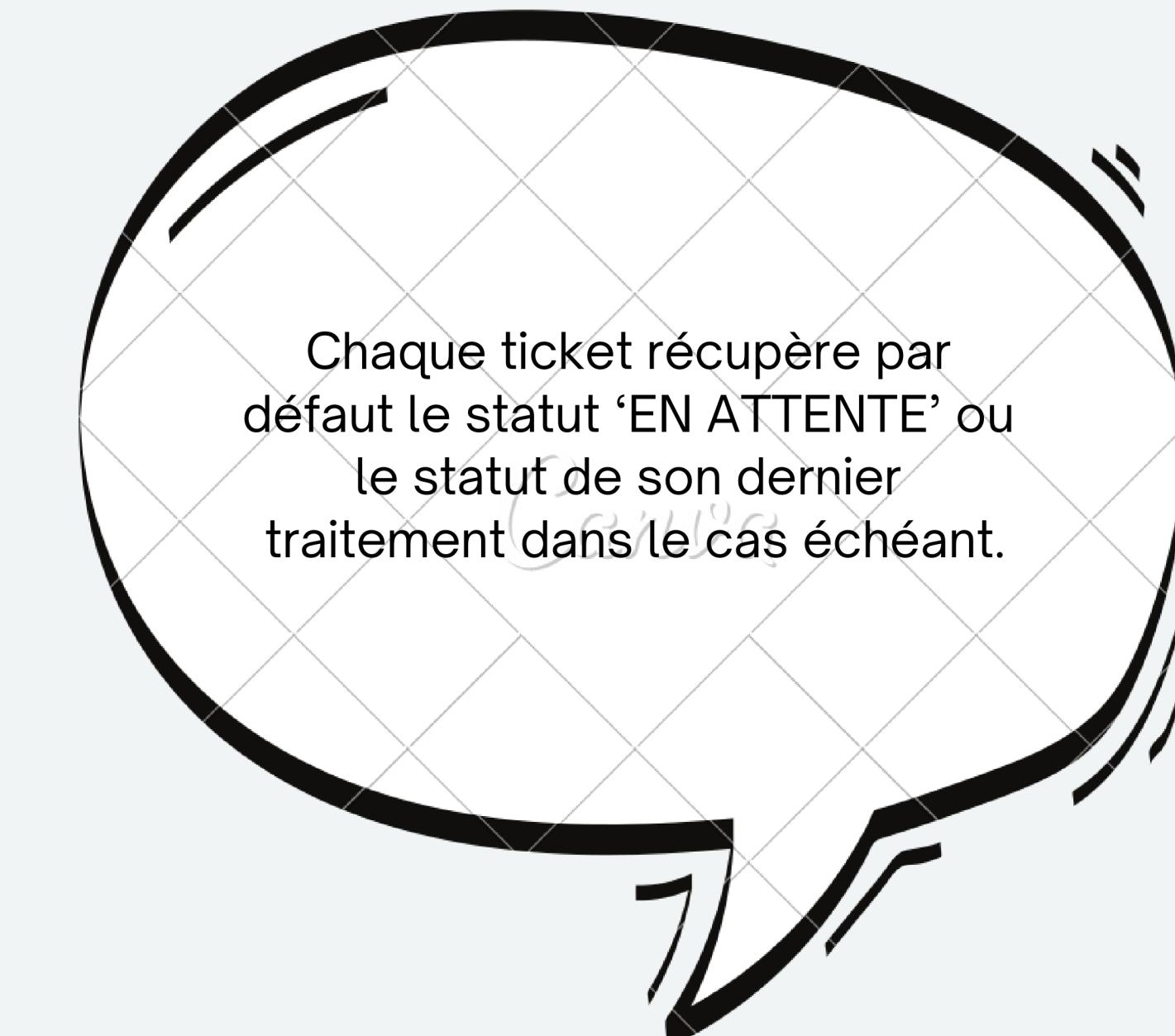
TICKET T#8
EN ATTENTE OUVERT LE 01/09/2023 12h23 TICKET T#8
Créer par Patrice Kenzo
Traitements: 1
VOIR

TICKET T#9
EN COURS OUVERT LE 01/09/2023 12h23 TICKET T#9
Créer par Martin Martin
Traitements: 2
VOIR



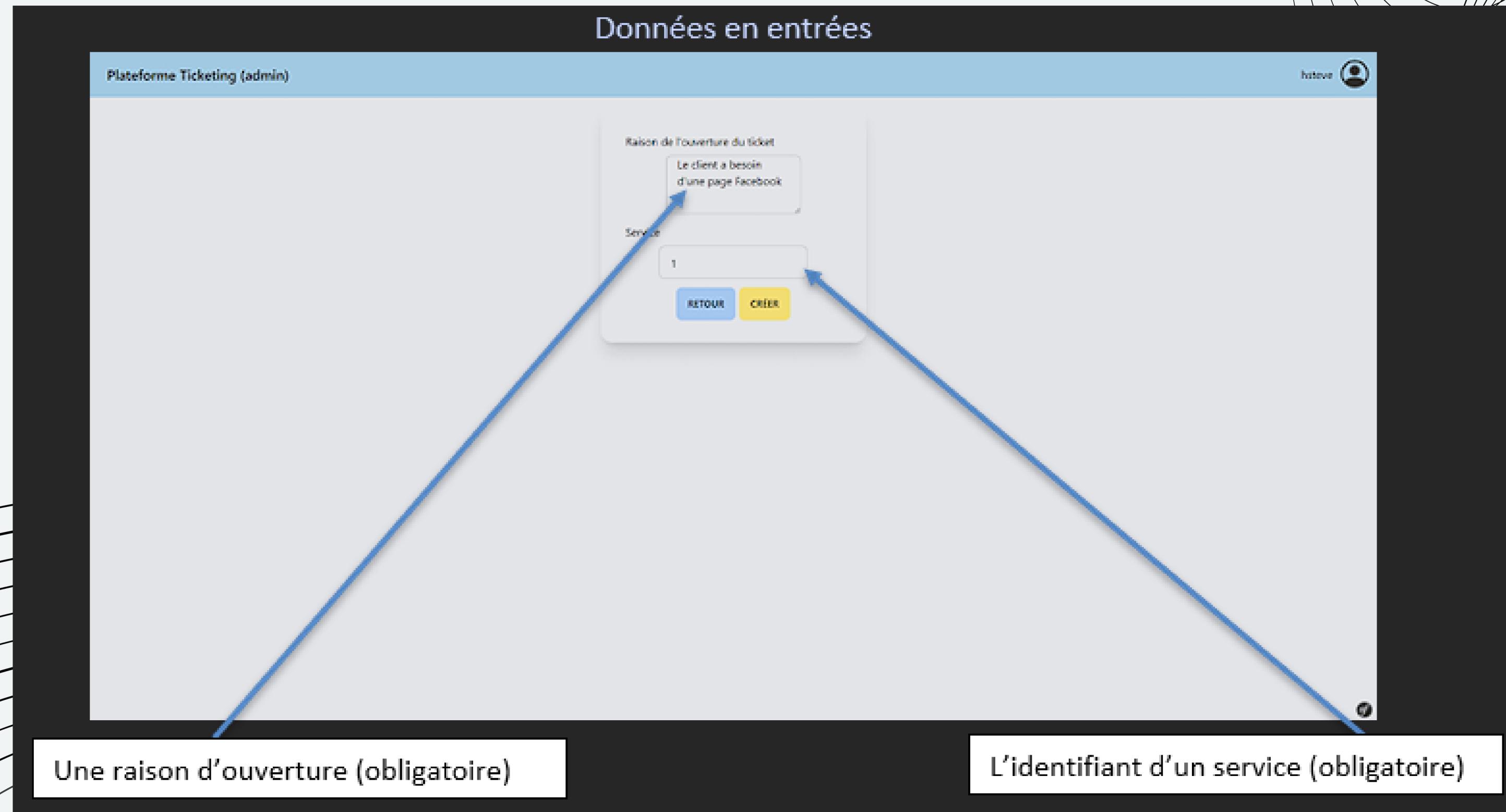
```
(X else X)(# L'utilisateur n'est pas ADMIN -> Mode user #)
<div>
  <a class="btn btn-secondary shadow-2xl ml-1" href="(( path('app_ticket_new') ))">CRÉER UN TICKET</a>
  <a class="btn btn-success" href="(( path('app_main') ))">VOIR LES TICKETS DE MON SERVICE</a>
  <a class="btn btn-ghost" href="(( path('app_main_my_tickets') ))">VOIR LES TICKETS QUE J'AI CRÉER</a>
<div class="text-center">
  <div class="badge badge-primary">SERVICE {{ app.user.service.name }}</div>
<div>
  (X if status == "EN ATTENTE" X)
    <a class="btn btn-success" href="(( path('app_main') ))">EN ATTENTE</a>
  (X else X)
    <a class="btn btn-ghost" href="(( path('app_main') ))">EN ATTENTE</a>
  (X endif X)
  (X if status == "EN COURS" X)
    <a class="btn btn-success" href="(( path('app_main') ))?status=EN_COURS">EN COURS</a>
  (X else X)
    <a class="btn btn-ghost" href="(( path('app_main') ))?status=EN_COURS">EN COURS</a>
  (X endif X)
  (X if status == "FERME" X)
    <a class="btn btn-success" href="(( path('app_main') ))?status=FERME">FERMÉ</a>
  (X else X)
    <a class="btn btn-ghost" href="(( path('app_main') ))?status=Ferme">FERMÉ</a>
  (X endif X)
</div>
<div class="grid grid-cols-3 mt-3 gap-4">
  (X set ticket_in_cat = false X)(# variable qui va servir à afficher ou pas le message de filtrage (instruction en bas)
  (X for ticket in app.user.service.ticket X)
    (# status est la catégorie qui filtre par status et qui est défini en requête GET #)
    (X if status == "EN ATTENTE" or (ticket.treatments.last != false and status in ticket.treatments.last.status) X)
      (X set ticket_in_cat = true X)
      <div class="card bg-gray-200 shadow-1xl">
        <div class="card-body">
          <h2 class="card-title">
            <span class="badge badge-warning">
              (X if ticket.treatments.last == false or ticket.treatments.last.status is null X)
              EN ATTENTE
            (X else X)
              (( ticket.treatments.last.status )) // le ticket récupère le statut du dernier traitement
            (X endif X)</span>
            <span class="badge badge-ghost">Ouvert le: (( ticket.createDate|date('d/m/Y H:i') ))</span>
            <span class="badge badge-ghost">(X if ticket.resultate is not null X)Fermé le: (( ticket.resultate|date('d/m/Y H:i') ))</span>
            <p>Créer par (( ticket.creator.firstname + " " + ticket.creator.name ))</p>
            <p>Traitements: (( ticket.treatments.length ))</p>
          <div class="card-actions justify-end">
            <a class="btn btn-primary">
              href="(( path('app_ticket_see', {id: ticket.id} )) )>voir
              <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16">
                fill="currentColor" class="bi bi-eye-fill" viewBox="0 0 16 16">
                  path d="M0.5 8a2.5 2.5 0 1 1-5 2.5 0 0 1 5 2.5</path>
                  path d="M0 8a5.5 5.5 0 1 1 10 0 0 1 5 5.5 0 0 1 5 5.5</path>
                </svg>
              </a>
            </div>
          </div>
        </div>
      </div>
    (X endif X)
    (X if not ticket_in_cat X)(# ... #)
      Il n'y a pas de ticket dans cette catégorie...
    (X endif X)
  </div>
</div>
(Xendif X)
```

```
(X else %)(# L'utilisateur n'est pas ADMIN -> Mode user %)
<div>
  <a class="btn btn-secondary shadow-2xl ml-1" href="{{ path('app_ticket_new') }}>CRÉER UN TICKET</a>
  <a class="btn btn-success" href="{{ path('app_main') }}>VOIR LES TICKETS DE MON SERVICE</a>
  <a class="btn btn-ghost" href="{{ path('app_main_my_tickets') }}>VOIR LES TICKETS QUE J'AI CRÉER</a>
<div class="text-center">
  <h1 class="badge badge-primary">SERVICE {{ app.user.service.name }}</h1>
<div>
  (% if status == "EN ATTENTE" %)
    <a class="btn btn-success" href="{{ path('app_main') }}>EN ATTENTE</a>
  (% else %)
    <a class="btn btn-ghost" href="{{ path('app_main') }}>EN ATTENTE</a>
  (% endif %)
  (% if status == "EN COURS" %)
    <a class="btn btn-success" href="{{ path('app_main') }}?status=EN_COURS>EN COURS</a>
  (% else %)
    <a class="btn btn-ghost" href="{{ path('app_main') }}?status=EN_COURS>EN COURS</a>
  (% endif %)
  (% if status == "Fermé" %)
    <a class="btn btn-success" href="{{ path('app_main') }}?status=Fermé>FERMÉ</a>
  (% else %)
    <a class="btn btn-ghost" href="{{ path('app_main') }}?status=Fermé>FERMÉ</a>
  (% endif %)
</div>
<div class="grid grid-cols-3 mt-3 gap-4">
  (% set ticket_in_cat = false %)(# variable qui va servir à afficher ou pas le message de filtreage (instruction on bas)
  (% for ticket in app.user.service.ticket %)
    (% status est la catégorie qui filtre par status et qui est défini en requête GET %)
    (% if status == "EN ATTENTE" or (ticket.treatments.last != false and status in ticket.treatments.last.status) %)
      (% set ticket_in_cat = true %)
      <div class="card bg-gray-200 shadow-xl">
        <div class="card-body">
          <h2 class="card-title">
            <span class="badge badge-warning">
              (% if ticket.treatments.last == false or ticket.treatments.last.status is null %)
                EN ATTENTE
              (% else %)
                {{ ticket.treatments.last }} // le ticket récupère le statut du dernier traitement
              (% endif %)</span>
            <span class="badge badge-ghost">Ouvert le: {{ ticket.createDate|date('d/m/Y H:i') }}</span>
            <span class="badge badge-ghost">(% if ticket.resultDate is not null %)Fermé le: {{ ticket.resultDate }}</span>
            <span class="badge badge-secondary">TICKET #{{ ticket.id }}</span></h2>
          <p>Créer par {{ ticket.creator.firstname ~ " " ~ ticket.creator.name }}</p>
          <p>Traitement: {{ ticket.treatments|length }}</p>
          <div class="card-actions justify-end">
            <a class="btn btn-primary">
              href="{{ path('app_ticket_see',{'id': ticket.id }) }}>voir
              <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16">
                fill="currentColor" class="bi bi-eye-fill" viewBox="0 0 16 16">
                  <path d="M10.5 8a2.5 2.5 0 1 1-5 0 2.5 2.5 0 0 1 5 0z"/>
                  <path d="M0 8a2.5 2.5 0 1 1-5 0 2.5 2.5 0 0 1 5 0z"/>
                </svg>
            </a>
          </div>
        </div>
      </div>
    (% endif %)
  (% endfor %)
  (% if not ticket_in_cat %)(# ... %
    Il n'y a pas de ticket dans cette catégorie...
  (% endif %)
</div>
</div>
(% endif %)
```



PRÉSENTATION DU JEU D'ESSAI

CRÉATION DE TICKET



CRÉATION DE TICKET

Données attendues

- Une raison d'ouverture non vide
- Un identifiant d'un service non vide et existant
- L'utilisateur connecté qui est remonté automatiquement dans le code

Données obtenues

<u>id</u>	<u>service_id</u>	<u>creator_id</u>	<u>create_date</u>	<u>problem</u>	<u>result</u>	<u>result_date</u>	<u>transferred</u>
12	1	1	2023-09-07 17:09:58	Le client a besoin d'une page Facebook	NULL	NULL	0

↳ Le ticket a été crée et ajouté dans la base de données

CRÉATION DE TICKET | EN CAS D'ERREUR

Raison de l'ouverture du ticket

Le client CL001 a
besoin de...

Service



Veuillez remplir ce champ.

Raison de l'ouverture du ticket

Le client a besoin de ...

Service

Rechercher... ▼



Veuillez remplir ce champ.

○ Ce service n'existe pas...

Raison de l'ouverture du ticket

Le client a besoin de ...

Service

S

RETOUR

CRÉER

MODIFICATION D'UN UTILISATEUR (ADMIN SEULEMENT)

Données en entrées

Plateforme Ticketing (admin)

Modifier un utilisateur

Nom d'utilisateur:

E-mail:

Nom:

Prénom:

Service:

Un nom d'utilisateur (obligatoire)

Un nom (obligatoire)

Une adresse email(obligatoire)

L'identifiant d'un service (obligatoire)

Un prénom (obligatoire)

The diagram illustrates the five mandatory fields required for modifying a user. It consists of five callout boxes with arrows pointing to specific input fields on the 'Modifier un utilisateur' form:

- A black box labeled "Un nom d'utilisateur (obligatoire)" points to the "Nom d'utilisateur" field.
- A white box labeled "Un nom (obligatoire)" points to the "Nom" field.
- A white box labeled "Une adresse email(obligatoire)" points to the "E-mail" field.
- A white box labeled "L'identifiant d'un service (obligatoire)" points to the "Service" field.
- A white box labeled "Un prénom (obligatoire)" points to the "Prénom" field.

MODIFICATION D'UN UTILISATEUR (ADMIN SEULEMENT)

Données attendues

- Un nom d'utilisateur non vide et non existant (sauf si le nom d'utilisateur est inchangé)
- Une adresse e-mail non vide et de type e-mail et non existant (sauf si l'adresse e-mail est inchangée)
- Un nom non vide
- Un prénom non vide
- L'identifiant d'un service non vide et existant

Données obtenues

<u>id</u>	<u>service_id</u>	<u>email</u>	<u>username</u>	<u>roles</u> <small>(<u>OC2Type:json</u>)</small>	<u>password</u>	<u>name</u>	<u>firstname</u>	<u>active</u>
2	1	jdoe@gmail.com	jdoe	["ROLE_USER"]	\$2y\$13\$SmDlrZZjKFNlo3Wv.8qvhs.0A0dN4INnwRmmH9aGQC8H...	John	Doe	1

↶ Avant la modification

<u>id</u>	<u>service_id</u>	<u>email</u>	<u>username</u>	<u>roles</u> <small>(<u>OC2Type:json</u>)</small>	<u>password</u>	<u>name</u>	<u>firstname</u>	<u>active</u>
2	1	john.doe@gmail.com	jdoe	["ROLE_USER"]	\$2y\$13\$SmDlrZZjKFNlo3Wv.8qvhs.0A0dN4INnwRmmH9aGQC8H...	John	Doe	1

↶ Après la modification

MODIFICATION D'UN UTILISATEUR | EN CAS D'ERREUR

Nom d'utilisateur

E-Mail

Veuillez remplir ce champ.

Plateforme Ticketing (admin)

hstevie

Ce nom d'utilisateur est déjà pris..

Modifier un utilisateur

Nom d'utilisateur	<input type="text" value="hstevie"/>
E-Mail	<input type="text" value="john.doe@gmail.com"/>
Nom	<input type="text" value="John"/>
Prénom	<input type="text" value="Doe"/>
Service	<input type="text" value="1"/>

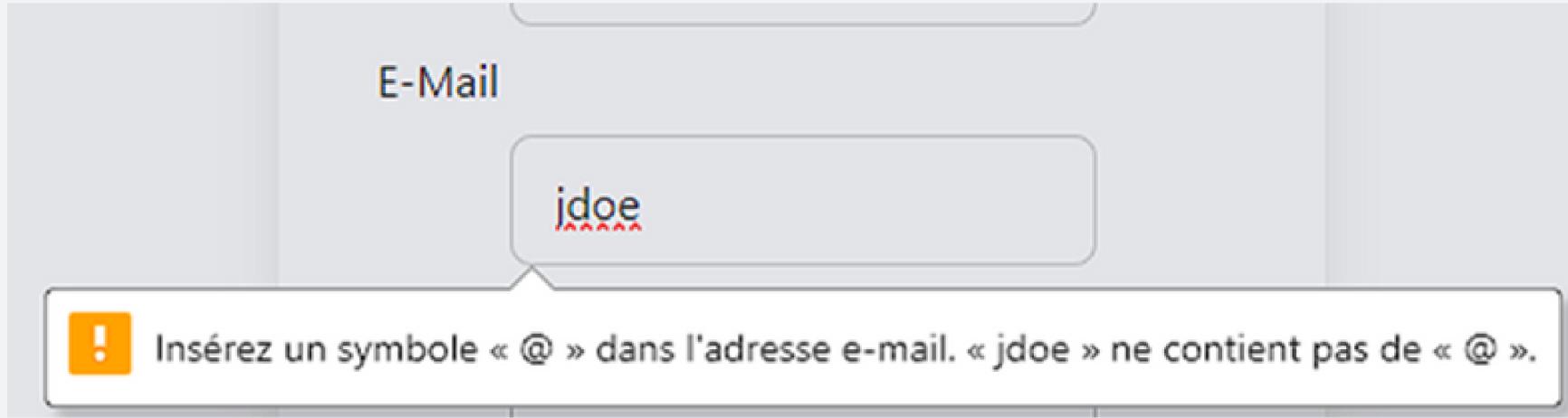
RETOUR **METTRE À JOUR**

E-Mail

Nom

Veuillez remplir ce champ.

MODIFICATION D'UN UTILISATEUR | EN CAS D'ERREUR



Plateforme Ticketing (admin)

Cet adresse mail est déjà prise...

Modifier un utilisateur

Nom d'utilisateur	jdoe
E-Mail	steve.hoareau1@gmail.co
Nom	John
Prénom	Doe
Service	1

RETOUR METTRE À JOUR

Plateforme Ticketing (admin)

Ce service n'existe pas...

Modifier un utilisateur

Nom d'utilisateur	jdoe
E-Mail	jdoe@gmail.com
Nom	Doe
Prénom	John
Service	5

RETOUR METTRE À JOUR

VEILLE DE VULNÉRABILITÉ



OWASP

Open Web Application
Security Project



Recherche des vulnérabilités

Consécration sur les 7 failles
les plus possibles



Mise en place d'un système
de sécurité

Essai de sécurisation de la
plateforme au maximum



Des possibles failles
peuvent être présentes à
cause des mises à jour des
composants qui sont trop
récentes

CONCLUSION



EXPER.net

CENTRE DE FORMATION
INFORMATIQUE