



République Tunisienne Ministère de
l'Enseignement Supérieur et de la
Recherche Scientifique
Université de Sfax



Rapport de stage technicien

Présenté à

**L'École Nationale d'Électronique
et des Télécommunications de
Sfax**

Effectué au :

MTD Group

Réalisé par :

Basma Gouiaa

Encadré par :

Mme. Sirine Ben Amor

APPLICATION WEB DU BOUTIQUE DE JOUETS

Période du stage :

11 JUILLET 2022 au 11 AOÛT 2022

Remerciements

Ce stage était pour moi le 2ème contact avec le milieu professionnel qui Suite à ma réussite en 2ème année ENETCOM, un stage technicienne est devenu fondamental pour compléter et appliquer les acquis théoriques reçus cette année.

Pour ce fait, j'ai choisi que mon stage se déroule à MTD Group afin d'accroître mes connaissances.

Je tiens à remercier l'équipe de MTD Group qui m'a permis de faire ce stage d'un mois parmi eux.

Je tiens à signaler qu'un accueil chaleureux m'a été réservé avec une grande disponibilité des responsables de la société tout au long de mon stage.

Je remercie tout particulièrement Madame Sirine Ben Amor et mes maîtres de stage qui m'ont fait découvrir de nombreuses choses.

Veuillez trouver dans ce travail l'expression de ma reconnaissance et le témoignage de mon profond respect

Sommaire

Introduction générale.....	1
1 Présentation de l'organisme d'accueil	2
1.1 Présentation générale	2
1.2 Domaine d'activité	3
2 Présentation et conception du projet	4
2.1 Objectif du projet :	4
2.2 Capture des besoins :	4
2.3 Besoins fonctionnels :	4
2.4 Besoins non fonctionnels :	5
3 Environnement de travail.....	6
3.1 Outils de développement et de conception :	6
3.1.1 Visual Studio Code :	6
3.1.2 StarUml :	6
3.1.3 CakePhp :	6
3.1.4 PhpMyAdmin :	7
3.2 Conception	8
3.2.1 Diagramme de classe.....	8
3.2.2 Diagramme de cas d'utilisation :	9
4 Architecture MVC	10
4.1 Modèle :	10
4.2 Vue :	11
4.3 Contrôleur :	11
5 La base de données :	11
6 Réalisation	12
6.1 Interface tickets.....	12
6.1.1 Bouton View :	13
6.1.2 Bouton édit.....	14
6.1.3 Bouton delete.....	15
6.2 Interface add ticket	16
6.3 Interface articles :	18
6.4 Interface règlements :	18
6.5 Interface modes :	19
Conclusion :	20

Liste des figures :

Figure 1 : logo MTD Group	2
Figure 2 : Les clients du MTD Group	2
Figure 3: Visual Studio Code.....	6
Figure 4: StarUml.....	6
Figure 5: CakePHP	6
Figure 6: PhpMyAdmin.....	7
Figure 7: Diagramme de classe.....	8
Figure 8: Diagramme de cas d'utilisation	9
Figure 9: Modèle MVC.....	10
Figure 10: Les tables de la base de données.....	11
Figure 11: Interface ticket	12
Figure 12: Code tickets	13
Figure 13: Interface View	13
Figure 14: Code de l'interface view	14
Figure 15: Interface Edit Ticket.....	14
Figure 16: Code de l'interface edit ticket	15
Figure 17: Alerte pour la suppression	15
Figure 18: Code d'alerte	15
Figure 19: interface add ticket	16
Figure 20: Code de l'interface add ticket	16
Figure 21: Les options du mode de paiement.....	17
Figure 22: Mode 'cheque'.....	17
Figure 23:Code du mode de paiement.....	17
Figure 24: Interface articles.....	18
Figure 25: Interface règlements	18
Figure 26: Interface modes	19

Introduction générale

Suite à ma réussite en deuxième année, un stage technique se trouve primordial pour enrichir ma formation et appliquer d'une manière correcte les connaissances théoriques reçues. Dans ce cadre, je suis menée à assister à un stage au sein de MTD Group.

Ce rapport récapitule tous ce que j'ai vu, j'ai appris et j'ai pratiqué durant le stage à MTD Group.

Ainsi, ce projet consiste à réaliser une application web qui permet de bien gérer et d'organiser les données afin d'aboutir à un résultat unique censé de satisfaire le besoin d'une boutique, dans notre cas boutique de jouets.

C'est dans ce contexte que se situe notre travail qui consiste à concevoir et à implémenter une application web centralisée de gestion des articles de la boutique pour améliorer ses services, augmenter la gestion des informations des articles et faciliter leurs tâches administratives.

1 Présentation de l'organisme d'accueil

1.1 Présentation générale :

Fondée en 2000, MTD groupe est une entreprise spécialisée en solution médiatique et logistique qui couvrent tous les besoins des sociétés pour la gestion des ressources et le développement de l'entreprise.

Conscient de l'importance de l'Internet, MTD groupe n'a cessé de croître et d'investir dans l'apprentissage et la maîtrise des techniques les plus avancées. Depuis 7 ans, le groupe s'est engagé dans une stratégie de croissance pour fournir aux entreprises une solution complète pour une meilleure gestion et communication.



Figure 1 : logo MTD Group

Aujourd'hui, MTD groupe opère dans des activités allant de la création des sites au développement des applications mobiles et à la promotion digitale des entreprises. De même, le groupe propose des logiciels ERP et des applications web ; MTD erp , web expert , MFV, carte fid....

En tant qu'entreprise de développement informatique, MTD Groupe s'engage à offrir la meilleure solution de gestion informatisée pour répondre aux besoins spécifiques aux entreprises.



Figure 2 : Les clients du MTD Group

1.2 Domaine d'activité :

L'intégration du système d'information MTD ERP permet l'optimisation de votre processus de gestion par la cohérence et l'homogénéité des informations ainsi par la protection de toute la base de donnée

MTD ERP, votre solution cloud pour toute gestion d'entreprise est une solution qui propose une mise en œuvre simple et adaptable aux objectifs des sociétés. C'est un logiciel de gestion open source qui permet à l'entreprise d'intégrer différentes fonctionnalités telle que :

- **La gestion commerciale :**

Gestion des clients : Organisez le portefeuille client de votre société

Gestion des achats : Planifiez le niveau d'inventaire, la demande de matériel, la périodicité des stocks

Gestion des fournisseurs : Établissez une évaluation de vos fournisseurs afin de gérer des relation d'affaires de bonnes compétences

Gestion du stock : Maintenez la stabilité de votre entreprise en assurant l'équilibre entre la rotation des stocks et l'approvisionnement

- **La gestion comptable :**

MTD ERP gère tous les aspects de gestion financière, que soit comptabilité générale ou comptabilité analytique. Ce module aide à gérer vos coûts, contrôler vos dépenses et optimiser vos liquidités afin d'augmenter vos profits.

- **La gestion des ressources humaines :**

Cet outil permet le suivi du personnel depuis le recrutement jusqu'au calcul de la paie ainsi l'édition des documents relatifs aux déclarations sociales (impôt, CNSS, Taxes...) et des documents comptables et financiers (livre de paie...).

2 Présentation et conception du projet

2.1 Objectif du projet :

ce projet consiste à réaliser une application web qui permet de bien gérer et d'organiser les données dans une boutique de jouets

La mise en place de cette application permet d'améliorer la méthode de travail. Cette application est destinée aux administrateurs de boutique afin de pouvoir ajouter, modifier, supprimer les articles, les achats ainsi contrôler le stock de la boutique avec une base de données bien déterminée.

2.2 Capture des besoins :

La capture des exigences techniques permet d'identifier toutes les contraintes concernant la conception de l'application

2.3 Besoins fonctionnels :

Les besoins fonctionnels sont les fonctionnalités que le système doit fournir à ses utilisateurs. L'outil n'est considéré opérationnel que s'il garantit la disponibilité de ses fonctionnalités. Dans ce qui suit, nous repérons les besoins fonctionnels que notre système doit satisfaire :

- **Authentification** : L'administrateur est amené à entrer correctement leur logins et leur mot de passe, il peut aussi les changer.
- **Gestion de règlements** : choisir le mode de paiement
- **Gestion des articles** : L'administrateur peut gérer les articles (ajouter, éditer, supprimer et afficher).
- **Gestion des tickets** : Création des tickets, Modifier, supprimer et afficher
- **Gestion de stock** : Contrôler le stock

2.4 Besoins non fonctionnels :

Les besoins non fonctionnels représentent les exigences qui ne concerne pas le comportement du système mais des contraintes qui doivent être respectées. Nous identifions les besoins non fonctionnels suivants :

- **Sécurité :**
 - L'application doit assurer la confidentialité des données de l'utilisateur.
- **Rapidité :**
 - L'application doit garantir la rapidité et la précision du résultat obtenu.
- **Ergonomie :**
 - Les interfaces doivent être simples, claires et bien organisées.
- **Extensibilité :**
 - L'application doit permettre l'ajout des modules supplémentaires dans le cas de besoin.
- **Tolérance aux pannes :**
 - Capacité du logiciel de fonctionner dans un environnement dégradé après une erreur.
- **Stabilité :**
 - Sensibilité aux changements qui pourrait causer des nouvelles défaillances.

3 Environnement de travail

3.1 Outils de développement et de conception :

3.1.1 Visual Studio Code :

On a utilisé cet éditeur de code open source pour le développement front-end et back-end puisqu'il supporte un grand nombre de langages grâce à multiples extensions



Figure 3: Visual Studio Code

3.1.2 StarUml :

C'est un logiciel de modélisation Uml gérant la majorité des diagrammes spécifiés dans la norme UML 2.0



Figure 4: StarUml

3.1.3 CakePhp :

CakePHP est un framework de développement rapide pour PHP, gratuit et open-source. C'est un ensemble de briques élémentaires pour les programmeurs qui créent des applications web. Il suit le motif de conception Modèle-Vue-Contrôleur, il facilite l'utilisation de Bases de données



Figure 5: CakePHP

3.1.4 PhpMyAdmin :

phpMyAdmin est un logiciel libre écrit en PHP qui a pour mission de s'occuper de l'administration d'un serveur de base de données MySQL ou MariaDB. Vous pouvez utiliser phpMyAdmin pour réaliser la plupart des tâches d'administration, ceci incluant la création de base de données, l'exécution de demandes, et l'ajout de comptes utilisateur.



Figure 6: PhpMyAdmin

3.2 Conception :

3.2.1 Diagramme de classe :

Le diagramme de classes nous présente les classes de notre système ainsi que les différentes relations entre ces classes. La figure montre notre diagramme de classes global.

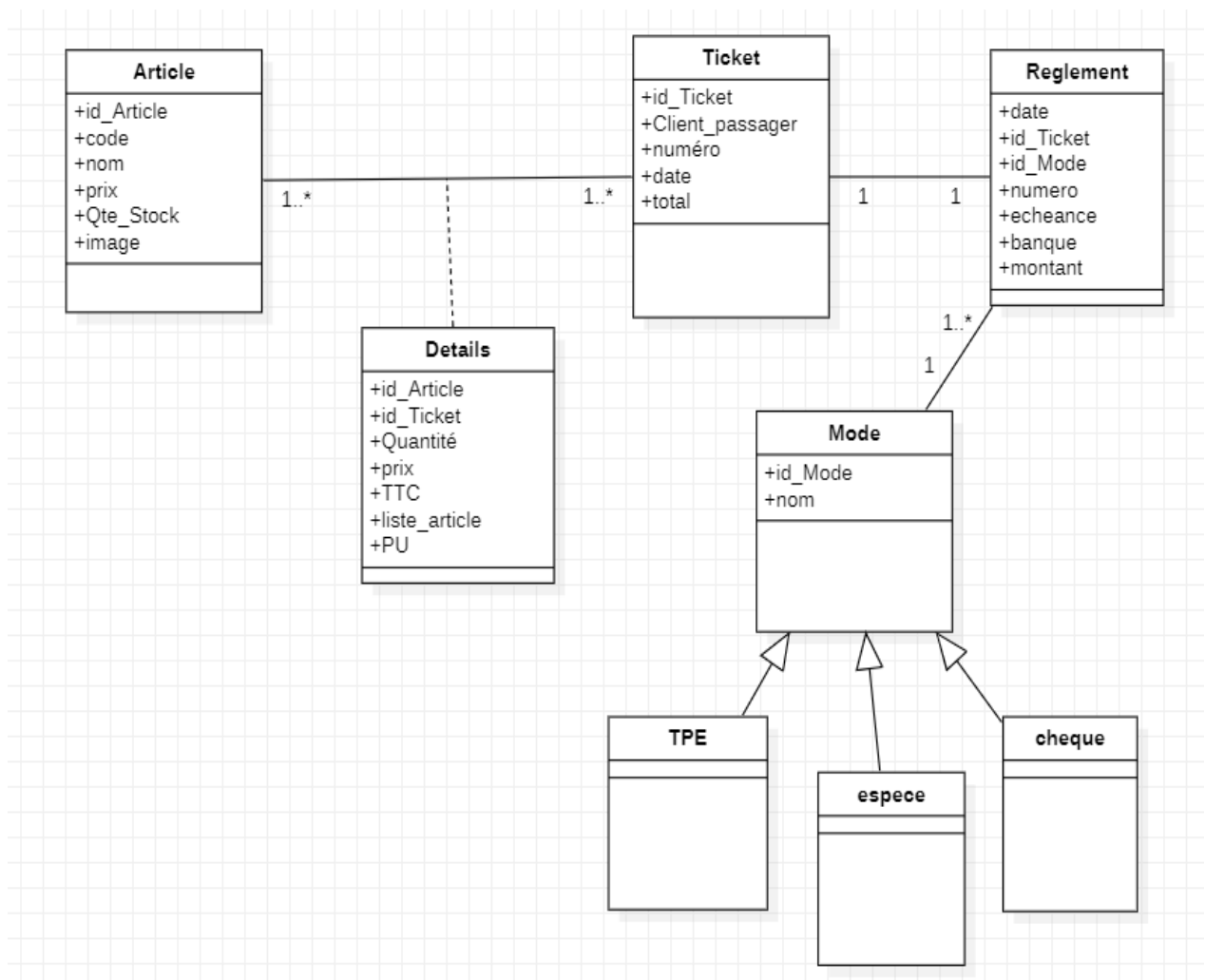


Figure 7: Diagramme de classe

3.2.2 Diagramme de cas d'utilisation :

Cette figure présente les différents cas d'utilisation exprimant toutes les possibilités d'interaction entre l'administrateur et la boutique de jouets et met en valeur l'ensemble des fonctionnalités données par notre application.

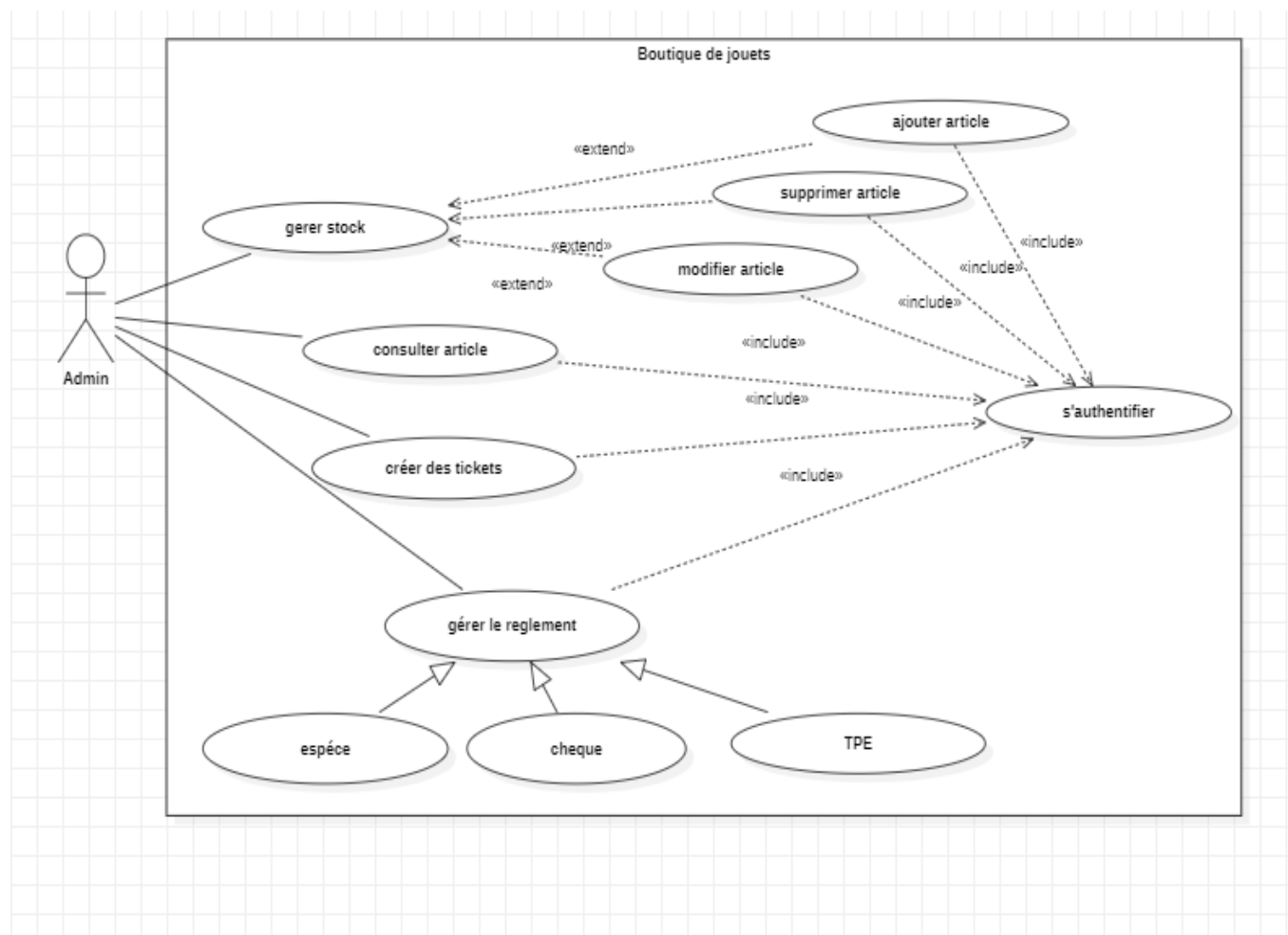


Figure 8: Diagramme de cas d'utilisation

Cas d'utilisation	Authentification
Acteur	Administrateur
Scenario nominal	<ul style="list-style-type: none"> - L'administrateur peut gérer le stock - L'administrateur peut consulter les articles - L'administrateur peut créer des tickets - L'administrateur peut gérer le règlement

4 Architecture MVC

Nous avons en fait reproduit le même raisonnement que de nombreux développeurs avant nous. En fait, il y a des problèmes en programmation qui reviennent tellement souvent qu'on a créé toute une série de bonnes pratiques que l'on a réunies sous le nom de design patterns.

Un des plus célèbres design patterns s'appelle MVC, qui signifie Modèle - Vue – Contrôleur.

C'est celui que nous allons découvrir maintenant.

Le pattern MVC permet de bien organiser son code source. Il va vous aider à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts.

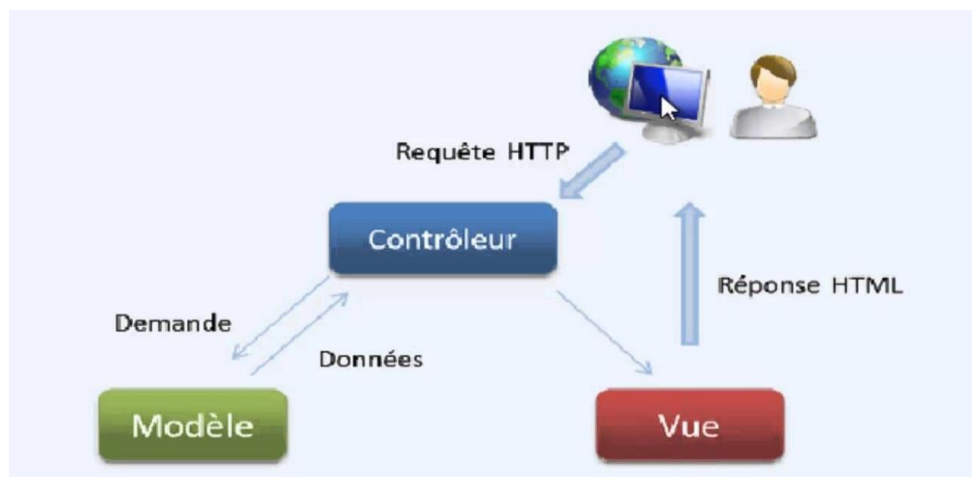


Figure 9: Modèle MVC

4.1 Modèle :

Cette partie gère les données de votre site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc entre autres les requêtes SQL.

4.2 Vue :

Cette partie se concentre sur l’affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu’elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple une liste de messages.

4.3 Contrôleur :

Cette partie gère la logique du code qui prend des décisions. C’est en quelque sorte l’intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. C’est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d’accès).

5 Base de données :

Bien entendu tout site de e-commerce a nécessairement besoin d’une base de données ou ce qu’on appelle un système d’information qui va contenir toutes les informations sur les produits.

Table ▲	Action	Lignes ⚙	Type	Interclassement	Taille	Perte
<input type="checkbox"/> articles	★      	4	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> details	★      	18	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> modes	★      	6	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> reglements	★      	6	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> tickets	★      	7	InnoDB	utf8mb4_general_ci	16,0 kio	-
5 tables	Somme	41	InnoDB	utf8mb4_general_ci	80,0 kio	0 o

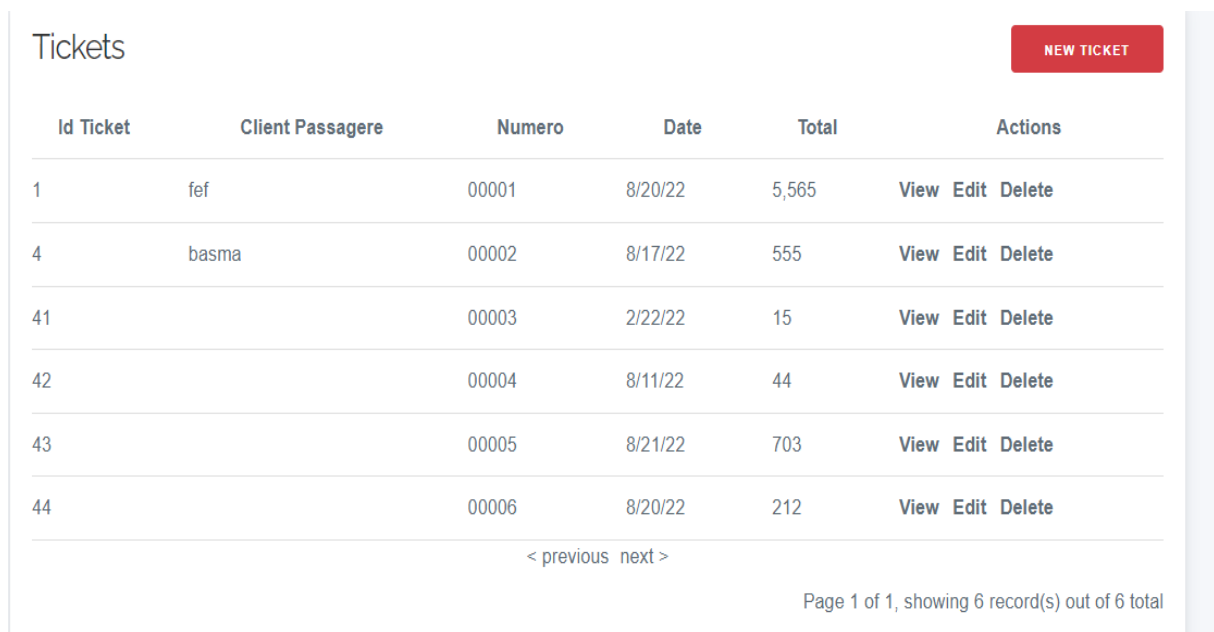
Figure 10: Les tables de la base de données

6 Réalisation

Les interfaces Homme/Machine constituent un élément important dans la réussite d'une application. Cette contribution est d'autant plus importante lorsqu'il s'agit d'une application web. Ainsi, nous essayons dans cette partie de présenter les interfaces les plus significatives de notre application.

6.1 Interface tickets :

L'interface présentée dans la figure 10 est l'interface de gestion des tickets qui affiche la liste des tickets créée dans la base de données. L'administrateur qui a la permission de consulter, modifier ou supprimer le ticket à partir des boutons présentés.



Tickets					
Id Ticket	Client Passagere	Numero	Date	Total	Actions
1	fef	00001	8/20/22	5,565	View Edit Delete
4	basma	00002	8/17/22	555	View Edit Delete
41		00003	2/22/22	15	View Edit Delete
42		00004	8/11/22	44	View Edit Delete
43		00005	8/21/22	703	View Edit Delete
44		00006	8/20/22	212	View Edit Delete

< previous next >

Page 1 of 1, showing 6 record(s) out of 6 total

Figure 11: Interface ticket


```

<h3><?= __('Tickets') ?></h3>
<div class="table-responsive">
    <table>
        <thead>
            <tr>
                <th><?= $this->Paginator->sort('id_ticket') ?></th>
                <th><?= $this->Paginator->sort('client_passagere') ?></th>
                <th><?= $this->Paginator->sort('numero') ?></th>
                <th><?= $this->Paginator->sort('date') ?></th>
                <th><?= $this->Paginator->sort('total') ?></th>
                <th class="actions"><?= __('Actions') ?></th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($tickets as $ticket): ?>
                <tr>
                    <td><?= $this->Number->format($ticket->id_ticket) ?></td>
                    <td><?= h($ticket->client_passagere) ?></td>
                    <td><?= h($ticket->numero) ?></td>
                    <td><?= h($ticket->date) ?></td>
                    <td><?= $this->Number->format($ticket->total) ?></td>
                    <td class="actions">
                        <?= $this->Html->link(__('View'), ['action' => 'view', $ticket->id_ticket]) ?>
                        <?= $this->Html->link(__('Edit'), ['action' => 'edit', $ticket->id_ticket]) ?>
                        <?= $this->Form->postLink(__('Delete'), ['action' => 'delete', $ticket->id_ticket], ['con
                    </td>
                </tr>
            </tbody>
        </table>
    </div>

```

Figure 12: Code tickets

6.1.1 Bouton View :

Lorsque L'administrateur clique sur le bouton « view » l'interface ci-dessous s'affiche
L'administrateur est donc autorisé à consulter les données nécessaires du ticket.

CakePHP
Documentation API

Actions
Edit Ticket
Delete Ticket
List Tickets
New Ticket

1

Client Passagere	fef
Numero	00001
Id Ticket	1
Total	5,565
Date	8/20/22

Related Details
Related Reglements

Figure 13: Interface View

```

templates > Tickets > view.php
11 <?= $this->Html->link(__('Edit Ticket'), ['action' => 'edit', $ticket->id_ticket], ['class' => 'side-
12 <?= $this->Form->postLink(__('Delete Ticket'), ['action' => 'delete', $ticket->id_ticket], ['confirm'
13 <?= $this->Html->link(__('List Tickets'), ['action' => 'index'], ['class' => 'side-nav-item']) ?>
14 <?= $this->Html->link(__('New Ticket'), ['action' => 'add'], ['class' => 'side-nav-item']) ?>
15 </div>
16 </aside>
17 <div class="column-responsive column-80">
18 <div class="tickets view content">
19 <h3><?= h($ticket->id_ticket) ?></h3>
20 <table>
21 <tr>
22 <th><?= __('Client Passagere') ?></th>
23 <td><?= h($ticket->client_passagere) ?></td>
24 </tr>
25 <tr>
26 <th><?= __('Numero') ?></th>
27 <td><?= h($ticket->numero) ?></td>
28 </tr>
29 <tr>
30 <th><?= __('Id Ticket') ?></th>
31 <td><?= $this->Number->format($ticket->id_ticket) ?></td>
32 </tr>
33 <tr>
34 <th><?= __('Total') ?></th>
35 <td><?= $this->Number->format($ticket->total) ?></td>
36 </tr>
37 <tr>
38 <th><?= __('Date') ?></th>
39 <td><?= h($ticket->date) ?></td>
40 </tr>

```

Figure 14: Code de l'interface view

6.1.2 Bouton édit :

En cliquant sur le bouton « édit » cette interface s'affiche pour que l'administrateur puisse modifier les données nécessaires et valider son choix afin d'enregistrer le ticket.

CakePHP
Documentation API

Actions
Delete
List Tickets

Edit Ticket

Client Passagere

Numero

Date

Total

SUBMIT

Figure 15: Interface Edit Ticket

```

19 <div class="column-responsive column-80">
20 <div class="tickets form content">
21 <?= $this->Form->create($ticket) ?>
22 <fieldset>
23 <legend><?= __('Edit Ticket') ?></legend>
24 <?php
25     echo $this->Form->control('client_passagere');
26     echo $this->Form->control('numero');
27     echo $this->Form->control('date');
28     echo $this->Form->control('total');
29 <?>
30 </fieldset>
31 <?= $this->Form->button(__('Submit')) ?>
32 <?= $this->Form->end() ?>
33 </div>

```

Figure 16: Code de l'interface edit ticket

6.1.3 Bouton delete :

La fonction du bouton « delete » est la suppression d'un ticket après l'affichage de cette alerte qui confirme la suppression.

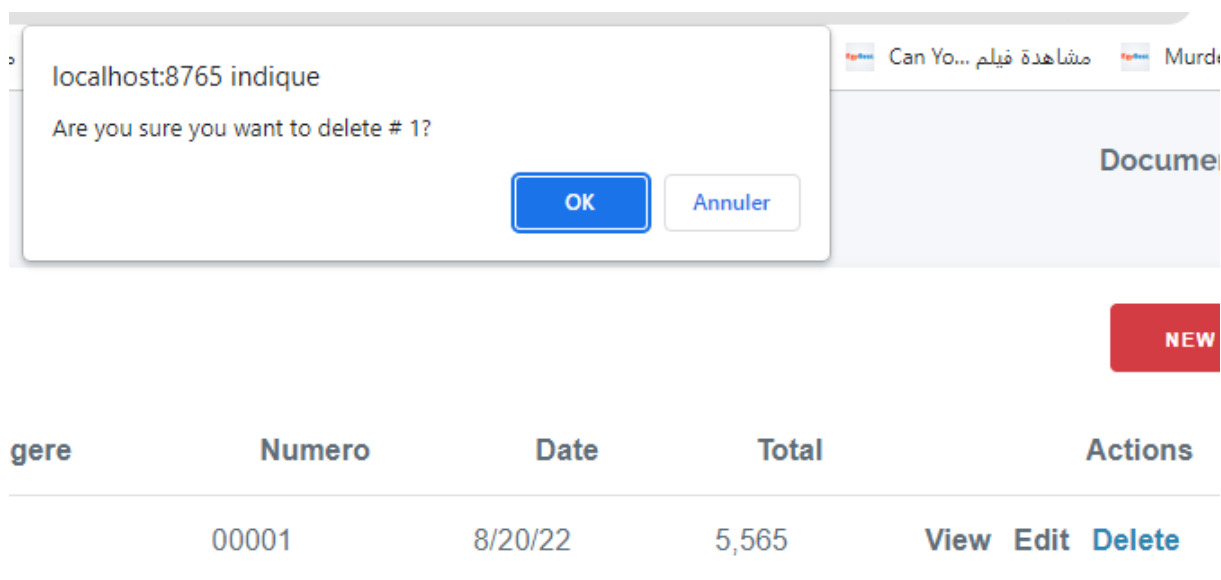


Figure 17: Alerte pour la suppression

```

?>
<div class="row">
<aside class="column">
<div class="side-nav">
<h4 class="heading"><?= __('Actions') ?></h4>
<?= $this->Form->postLink(
    __('Delete'),
    ['action' => 'delete', $ticket->id_ticket],
    ['confirm' => __('Are you sure you want to delete # {0}?', $ticket->id_ticket), 'class' => 'side-
    ) ?>
<?= $this->Html->link(__('List Tickets'), ['action' => 'index'], ['class' => 'side-nav-item']) ?>
</div>
</aside>

```

Figure 18: Code d'alerte

6.2 Interface add ticket :

L'interface présentée dans la figure 19 est l'interface d'ajout d'un ticket. Le numéro s'affiche automatiquement dès que l'administrateur ouvre la page et les informations sont enregistrées dans la base de données après avoir rempli toutes les informations.

Figure 19: interface add ticket

```

    <div class=numero>
    <p>Numero:</p> <input type="number" name="numero" value="<?php echo $c;?>" />
    </div>
    <div class=date>
    <p>Date:</p> <input type="date" name="date" />
    </div>
</div>
<div>
    <input class="btn-article" type="button" value="+" onClick="addRow('dataTable')" />
</div>
<div class=articles>
<table id="dataTable">
    <tr>
        <th> Article</th>
        <th> quantité</th>
        <th> PU</th>
        <th> total</th>
    </tr>
    <tr>
        <td>
            <select name="pets[0][article_id]" id="pet-select">
            <?php
            foreach($listArticle as $elem)
            {
                ?>
                <option value="<?php echo $elem ?>"><?php echo $elem ?></option>

```

Figure 20: Code de l'interface add ticket

La dernière étape dans l'interface d'ajout ticket c'est

le mode de paiement, on a trois choix : espèce , TPE et cheque.

Lorsque L'administrateur choisit cheque les trois champs s'ajoutent dans la page :

Le numéro pour identifier le cheque

L'échéance c'est la date limite de paiement

Le nom de la banque

Figure 21: Les options du mode de paiement

Figure 22: Mode 'cheque'

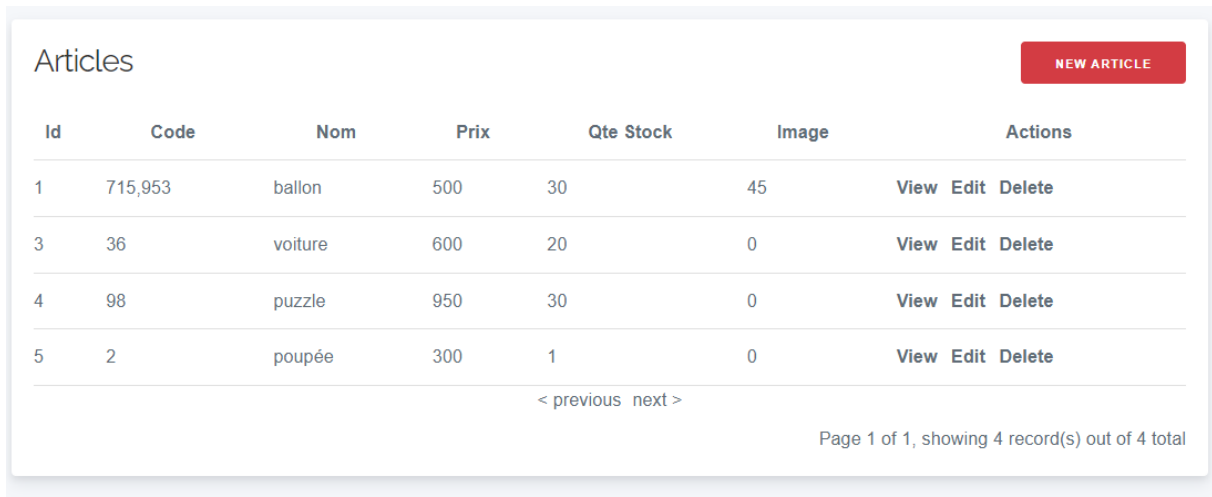
```
const select = document.querySelector("#mode");
let d1 = document.getElementById("cheque");
select.addEventListener('change', setMode);
d1.style.display = "none";
function setMode() {
  const choice = select.value;
  if (choice === 'Cheque') {
    d1.style.display = "block";
  }
  else {
    d1.style.display = "none";
  }
}

144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
<div class="mode">
<p>Mode:</p>
<select name="mode" id="mode" value="choisir une option">
  <option value="choisir une option">choisir une option</option>
  <option value="Espèce">Espèce</option>
  <option value="TPE">TPE</option>
  <option value="Cheque">Cheque</option>
</select>
</div>
<div id="cheque" class="cheque">
<div class="num">
<p>Numéro:</p> <input type="number" name="num" id="numero"/>
</div>
<div class="echeance">
<p>Echeance:</p> <input type="date" name="echeance" />
</div>
<div class="banque">
<p>Banque:</p> <input type="text" name="banque" />
</div>
</div>
</div>
```

Figure 23: Code du mode de paiement

6.3 Interface articles :

L'interface présentée dans la figure 24 est l'interface des articles qui affiche la liste crée dans la base de données dont nous pouvons consulter, ajouter, modifier ou supprimer selon l'utilisation à partir des boutons présentés.



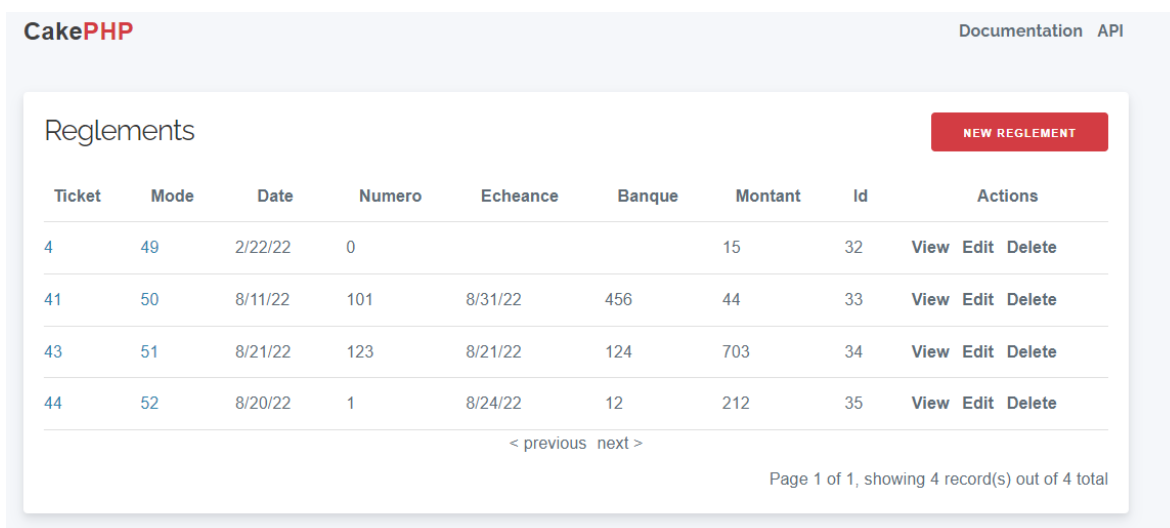
The screenshot shows a web interface titled 'Articles'. At the top right is a red button labeled 'NEW ARTICLE'. Below it is a table with the following columns: Id, Code, Nom, Prix, Qte Stock, Image, and Actions. The table contains 4 records. At the bottom of the table are navigation links '< previous next >' and a footer indicating 'Page 1 of 1, showing 4 record(s) out of 4 total'.

Id	Code	Nom	Prix	Qte Stock	Image	Actions
1	715,953	ballon	500	30	45	View Edit Delete
3	36	voiture	600	20	0	View Edit Delete
4	98	puzzle	950	30	0	View Edit Delete
5	2	poupée	300	1	0	View Edit Delete

Figure 24: Interface articles

6.4 Interface règlements :

L'interface présentée dans la figure 25 est l'interface du règlements qui affiche la liste crée dans la base de données dont nous pouvons consulter, ajouter, modifier ou supprimer selon l'utilisation à partir des boutons présentés.



The screenshot shows a web interface titled 'Règlements' under the 'CakePHP' logo. At the top right are links for 'Documentation' and 'API'. Below is a red button labeled 'NEW REGLEMENT'. The table has columns: Ticket, Mode, Date, Numero, Echeance, Banque, Montant, Id, and Actions. It displays 4 records. Navigation links '< previous next >' and a footer 'Page 1 of 1, showing 4 record(s) out of 4 total' are at the bottom.

Ticket	Mode	Date	Numero	Echeance	Banque	Montant	Id	Actions
4	49	2/22/22	0			15	32	View Edit Delete
41	50	8/11/22	101	8/31/22	456	44	33	View Edit Delete
43	51	8/21/22	123	8/21/22	124	703	34	View Edit Delete
44	52	8/20/22	1	8/24/22	12	212	35	View Edit Delete

Figure 25: Interface règlements

6.5 Interface modes :

L'interface présentée dans la figure 26 est l'interface du mode du paiement pour chaque ticket il affiche la liste créées dans la base de données dont nous pouvons consulter, ajouter, modifier ou supprimer selon l'utilisation à partir des boutons présentés.

Modes

NEW MODE

Id	Nom	Actions
22	Cheque	View Edit Delete
49	Espèce	View Edit Delete
50	Cheque	View Edit Delete
51	Cheque	View Edit Delete
52	Cheque	View Edit Delete

< previous next >

Page 1 of 1, showing 5 record(s) out of 5 total

Figure 26: Interface modes

Conclusion :

En guise de conclusion, ce projet était une véritable opportunité pour apprécier et enrichir notre formation à L'Ecole Nationale d'Electronique et des Télécommunications de Sfax (ENET'Com) en apprenant des nouvelles technologies. Il était aussi, une véritable clef d'accès vers les meilleures perspectives de la future vie active et professionnelle.

Nous avons commencé par présenter notre projet dans son cadre général, puis nous avons présenté les spécifications des besoins fonctionnels et non fonctionnels, auxquels devra répondre notre application. Une fois la conception élaborée, nous avons exposé le travail effectué à travers quelques captures d'écran.

L'objectif de ce projet nous permet de s'améliorer dans le développement d'applications web, d'approfondir et d'enrichir notre base de connaissances. De plus, il nous a permis de nous familiariser avec la gestion des services informatiques de manière efficace et systématique.