

RAPPORT DE
STAGE

Remerciement :

Merci à l'entreprise IDKIDS de m'avoir accompagné tout au long de ce stage.

Merci au groupe MCO de m'avoir pris dans leur groupe et de m'avoir appris de nouvelles choses.

Merci à Nilton De Oliveira de m'avoir fait confiance pour réaliser ce projet.

Merci à Gauthier Lecroart et Maxance de m'avoir aidé sur les choses que je ne comprenais pas.

Table des Matières

1.	Introduction.....	3
2.	Environnement.....	3
2.2.	L'entreprise.....	3
2.3.	L'équipe projet.....	6
2.4.	Ressource Fournies et/ou utiliser.....	6
2.5.	Contraintes particulières.....	6
3.	Conception et réalisation de l'application Web.....	7
3.1.	Description, résultats attendus et état d'avancement.....	7
3.2.	Étude du besoin.....	7
3.2.1.	Contexte.....	7
3.2.2.	Choix de développement.....	8
3.3.	Choix des technologies.....	8
3.3.1.	Choix du langage de programmation.....	8
3.3.2.	Choix du framework CSS.....	10
3.4.	Rédaction du cahier des charges.....	12
3.5.	Modélisation de l'application.....	12
3.5.1.	Modélisation de la base de données de l'application.....	13
3.6.	Réalisation de la maquette.....	14
4.	Conclusion.....	20
5.	Sources.....	21

1. Introduction

Dans le cadre de ma première année en BTS SIO à l'Epsi de Arras, je dois effectuer un stage de 5 semaines. Ce stage en deux parties 5 semaines la première année et 5 autres la 2eme année vise à clôturer le BTS SIO. Il me permet d'être formé au sein d'une entreprise dans le but d'acquérir des connaissances sur un secteur d'activité tout en me permettant de mettre en pratique ce que j'ai appris lors de cette année.

Dans ce rapport je présente mon environnement de travail ainsi que la mission que j'ai réalisée au sein de l'entreprise IDKIDS situé à Roubaix, à savoir la réalisation d'une FAQ. En effet le Groupe IDKIDS possède des FAQ sur leur site marchand mais pas pour leur groupes interne c'est à dire que si une personne dans une équipe se pose une question cette dernière pourrait avoir soit une réponse donné soit la personne concerné et qui sera la plus qualifiée pour y répondre

2. Environnement

2.1. L'entreprise



Figure 1 : Logo IDKIDS

A. Historique :

1996 : Jean Duforest et Jean-Luc Souflet reprennent ensemble l'activité enfant du groupe Camaïeu, soit 30 magasins et 100 collaborateurs. Leur projet : "Entreprendre pour que le monde progresse au service de l'enfant qui grandit".

2000 : Lancement officiel de la marque Okaïdi.

2001 : Développement de Okaïdi à l'international.

2002 : Okaïdi lance Obaïbi, dédiée au bien-être des petits de 0 à 5 ans.

2005 : Intégration de Jacadi, marque de prestige inspirée du patrimoine de la mode enfantine française. L'entreprise, désormais multimarques, devient IDGROUP.

2010 : L'engagement WE ACT FOR KIDS rend visible pour nos clients la mission fondatrice de 1996 « entreprendre pour que le monde progresse au service de l'enfant qui grandit ».

2010 : Intégration de la marque éveil et jeux, qui devient Oxybul - Eveil et jeux.

2015 : Fusion des réseaux de crèches Des étoiles plein les yeux et Rigolo Comme La Vie. Rigolo Comme La Vie passe du statut associatif à celui de business à caractère social.

2015 : Lancement de www.idkids.com , le lieu d'accueil digital des communautés de marques, d'experts et de clients de toutes les marques du groupe.

2017 : IDGROUP devient la communauté IDKIDS.

B. Activités :

Le groupe IdKids propose différents services liés à l'enfance, à l'accompagnement des enfants et de leurs parents.

- ➔ Okaïdi propose des vêtements pour les enfants de 2 à 14 ans avec des produits de plus en plus respectueux de l'environnement tout en restant accessibles. La marque est contemporaine avec une esthétique dans l'air du temps pour permettre à chaque enfant d'exprimer ses goûts et son identité
- ➔ Obaïbi conçoit des vêtements pour les bébés de 0 à 3 ans. Ces habits sont colorés et sensu-ludiques pour éveiller l'enfant au plus tôt. Les produits sont aussi respectueux de l'environnement et de l'enfant
- ➔ Jacadi est une marque de prestige pour enfants de la naissance à 12 ans. En recherchant une qualité irréprochable, Jacadi perpétue le patrimoine artisanal de la confection de tradition française. Elle propose aux jeunes des collections respectueuses de la tendresse et de la fraîcheur de l'enfance, pour tous les jours et tous les moments de vie.
- ➔ Oxybul conçoit et sélectionne des jeux et jouets qui permettent à chaque enfant de découvrir et développer chaque jour ses talents de cœur, de corps et d'esprit, pour grandir en autonomie et en estime de soi. La marque se rend accessible, durable et respectueuse de l'enfant.
- ➔ Absorba et Catimini proposent des vêtements pour les enfants de la naissance jusqu'à 10 ans pour Absorba et 14 pour Catimini. Ils proposent des produits innovants et de qualités, respectueux de l'enfant et de l'environnement.
- ➔ Les autres marques de IdKids (Bubble, Consobaby, Rigolo comme la vie, Joyvox et Cnabulle) proposent des services pour accompagner les enfants et surtout les parents dans leurs vies quotidiennes.

C. Implantation géographique

Le groupe est implanté internationalement à travers ses magasins et ses sites internet.

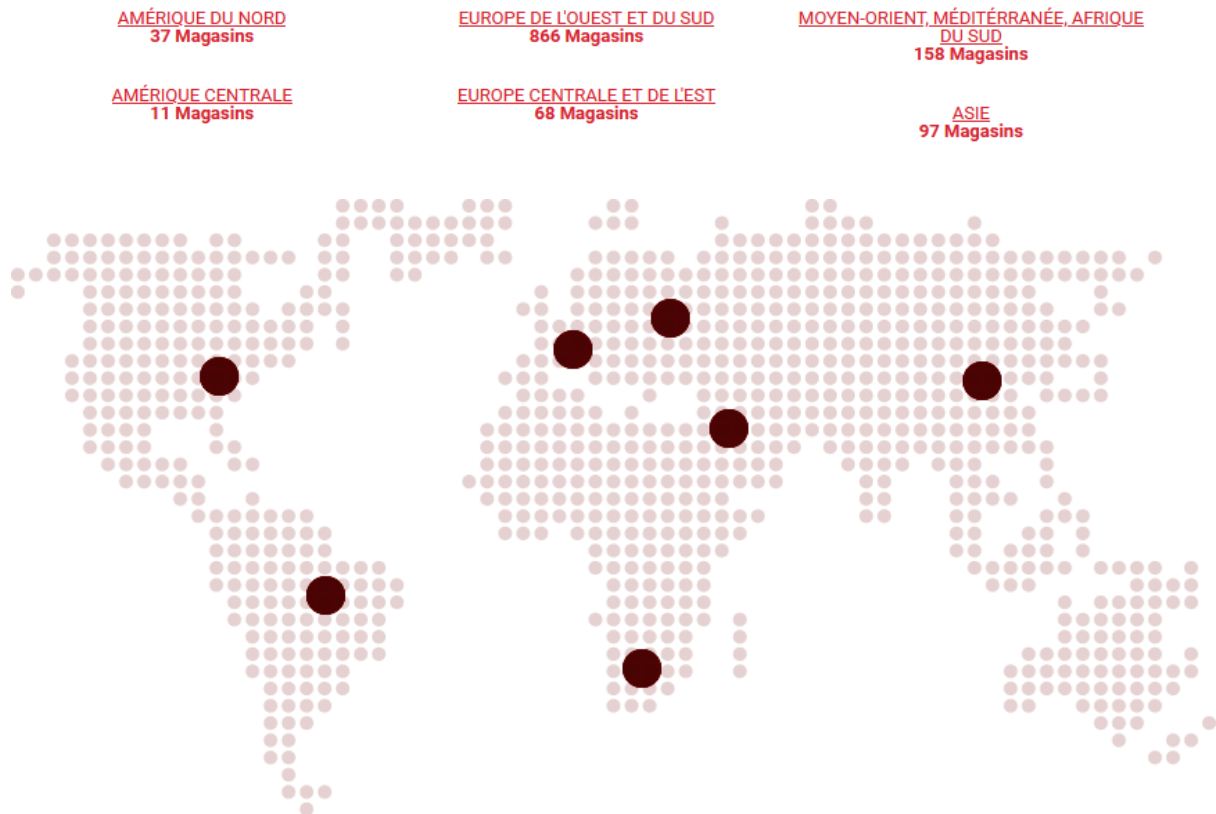


Figure 2 : Carte des magasin du groupe IDKIDS

D. Chiffre clés :

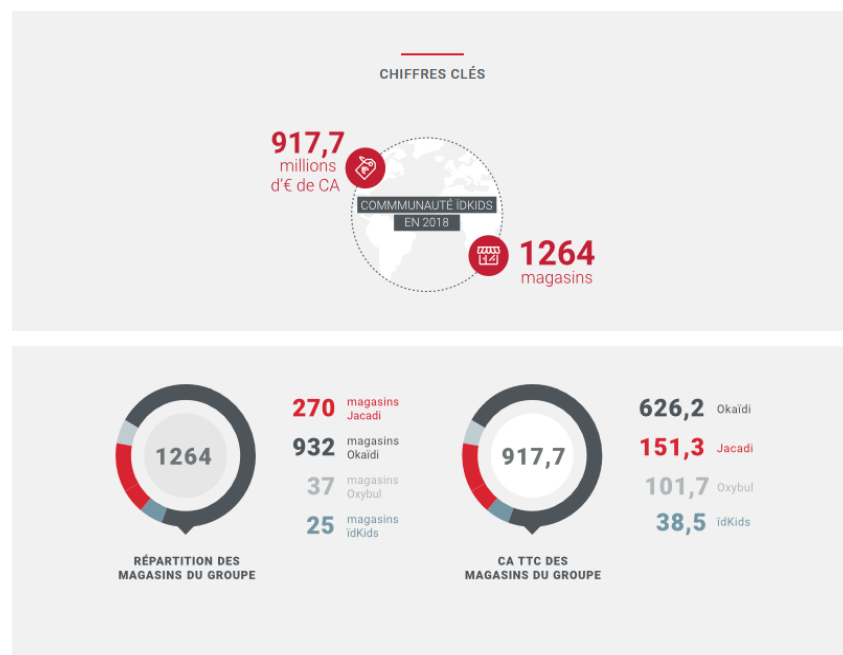


Figure 3 : Chiffre clés de IDKIDS

2.2. L'équipe Projet

L'équipe du projet est constituée de quatre personnes :

- Nilton De oliveira, Responsable de production digital,
- Gauthier Lecroart, Responsable d'application web,
- Maxence, Responsable d'application web,
- Moi-même, Thomas Delos, stagiaire en développement informatique

Mon tuteur en entreprise pendant ce stage est M. Gauthier Lecroart. Les étapes de développement de l'application seront validées par M. Nilton De oliveira et M. Maxence. Mes choix technologiques quant à eux, sont soumis à l'approbation de mon tuteur de stage en entreprise.

Les membres du service informatique sont là pour me permettre d'avoir accès à l'infrastructure informatique de l'entreprise, me conseiller sur les technologies à utiliser, m'apporter des pistes de réflexion et m'aider dans la réalisation de certaines fonctionnalités.

Suite à cela j'ai observé qu'ils travaillaient selon la méthode Agile SCRUM.

La méthode AGILE est une méthodologie de gestion de projet. Il existe en réalité plusieurs méthodes qui ont toutes un point commun : elles découlent toutes du Manifeste Agile. Édité en 2001, le manifeste AGILE a été développé par plusieurs développeurs de logiciels. Son but : améliorer leur process et réduire leur taux d'échec. Pour cela, ils placent le client au cœur du projet et ils s'adaptent tout le long du fil du projet. C'est donc une toute nouvelle façon de voir les choses, et d'aborder le développement d'un produit, d'un service ou d'un projet. Depuis, les méthodes qui s'inscrivent dans la philosophie de ce manifeste sont appelées méthodes agiles. Certains parlent également de lean agile.

Une méthodologie agile prévoit la fixation d'objectifs à court terme utilisés dans la gestion de projets. Alors que la méthode traditionnelle prévoit la planification totale du projet avant même la phase développement. Le projet est ainsi fragmenté en plusieurs sous-parties que les équipes de développement qui en a la charge se doivent d'atteindre progressivement en ajustant si nécessaire les objectifs pour répondre le plus possible aux attentes du client. On parle alors de sprints. Chaque spring ayant pour objectif de clôturer une brique du projet. Les méthodes agiles permettent de renforcer les relations entre les membres d'une équipe, mais aussi entre l'équipe et le client. C'est pour cette raison que la flexibilité et la souplesse sont deux piliers forts dans la méthode AGILE. On parle d'un mode itératif qui permet de délivrer des prestations avec efficacité et réactivité le tout en respectant des plannings exigeants. Cela permet de gérer ses projets de façon performante.

2.3. Ressources fournies et/ou à utiliser

Plusieurs ressources m'ont été fournies dans le cadre de mon stage. Dans un premier temps, un PC a été mis à ma disposition, avec le système d'exploitation Windows. Connecté au réseau de IDKIDS, celui-ci m'a permis développer l'application Visual Studio. J'ai également eu un accès au serveur sur lequel était stockée la base.

2.4. Contraintes Particulières

Les application déjà existante étant C# j'ai dû pour un problème de compatibilité me former a ce derniers durant la première semaine de mon stage afin de pouvoir comprendre et développer sur ce nouveau langage tout en utilisant la méthode MVC, qui est une méthode de structuration de projet.

3. Conception et réalisation de l'application

3.1. Description, résultats attendus et état d'avancement

La phase de conception de l'application est la plus importante du projet. Elle consiste à recueillir les besoins des agents, comprendre l'outil déjà en place, modéliser l'application et préparer son développement, de manière à ce qu'elle s'intègre correctement dans le système informatique d'IDKIDS.

3.2. Etude du besoin

3.2.1 Contexte

Les utilisateurs de la Faq doivent être capable de trouver une question rapidement avec la meilleure réponse possible. Toutes les questions posées doivent être enregistrées dans une base de données avec la réponse correspondante. Il est possible de se connecter afin d'ajouter de nouvelles questions et réponses. Il y a une partie administration permettant de gérer les droits des utilisateurs afin de savoir si oui ou non ils ont accès à l'ajout de question.

Si la réponse renvoie une liste de personnes comme réponse nous avons la possibilité de leur envoyer un mail ou de programmer une réunion avec ces personnes afin de répondre au plus vite au besoin.

3.2.2 Choix de développement

J'ai choisi de développer l'application en respectant plusieurs règles ayant attrait au Green IT à savoir :

- L'utilisation de bonnes pratiques de programmation usuelles
- L'utilisation d'un langage de programmation adapté

Ces bonnes pratiques permettront de réduire la charge CPU des équipements en la répartissant du côté client et serveur. En effet, la production de matériel informatique étant plus polluante que leur utilisation, les préserver, dans le but de les garder plus longtemps, réduit l'empreinte carbone des entreprises

3.3. Choix des technologie

3.3.1 Choix du langage de programmation

Le langage de programmation utilisé va beaucoup influencer sur le projet et la manière dont celui-ci sera développé, en fonction des avantages et des inconvénients du langage. Il convient de le choisir en considérant les besoins de l'entreprise, pour éviter de devoir changer de langage en cours de projet, ce qui constituerait une perte de temps considérable. De plus, un langage optimisé et facile à apprendre permettra d'avoir une meilleure optimisation de la charge du CPU, ce qui aura pour conséquence de préserver le matériel et de faciliter la maintenance, ce qui est en cohérence avec l'approche Green IT du projet.

L'entreprise passant tous ses application Web sous le langage C#.

C#

Le C# est un langage de programmation orienté objet commercialisé par Microsoft depuis 2002. Il est utilisé au sein du Framework .NET de Microsoft. Ce langage est utilisé dans un environnement ASP.NET dans le cas d'une utilisation pour le web.

Le C# est un langage typé dérivé du C++. Il comporte un ramasse-miettes et un système de gestion d'exceptions. Il y a beaucoup de similarité entre C# et Java.

Il est, à l'origine, destiné à être essentiellement utilisé sur Windows. Cependant, il existe de solutions alternatives, telles que Mono, pour utiliser C# sur des systèmes d'exploitation comme Linux ou Mac OSX.

L'avantage de ce langage est qu'il est fortement couplé avec les outils `Microsoft` et permet ainsi de mieux interagir avec eux. Par exemple, il est possible de générer des documents Microsoft Word via la technologie OLE Automation.

Le langage possède de nombreux composants pour communiquer avec d'autres protocoles comme IMAP et POP3 ou encore pour l'utilisation de sockets.

Dans le cadre de notre développement, l'utilisation du `C#` peut être très avantageux de par son couplage fort avec Microsoft et ses outils en général, les serveurs de Veolia étant principalement équipés de Microsoft Windows Server 2008.

Tout en suivant la méthode MVC qui est en développement par programmation orientée objet, MVC (Model-View-Controller, ou modèle-vue-contrôleur) est le nom d'une méthodologie ou d'un motif de conception visant à faire le lien entre l'interface utilisateur et les modèles de données sous-jacents.

Le motif MVC se compose de trois modules ou objets principaux à utiliser dans le développement de logiciels :

- Un *modèle* représentant la structure logique sous-jacente des données dans une application logicielle, ainsi que la classe supérieure qui y est associée. Ce modèle d'objet ne contient aucune information sur l'interface utilisateur.
- Une *vue*, autrement dit un ensemble de classes représentant les éléments de l'interface utilisateur (tous ceux que l'utilisateur voit à l'écran et avec lesquels il peut interagir : boutons, boîtes de dialogue, etc.).
- Un *contrôleur* représentant les classes qui se connectent au modèle et à la vue, et servant à la communication entre les classes dans le modèle et la vue.

3.3.2 Choix du framework CSS

Les feuilles de style en cascade, ou en anglais, Cascading StyleSheet (CSS), forment un langage informatique décrivant la manière dont les éléments d'une interface HTML doivent être affichés. Introduit dans les années 1990, CSS est un langage utilisé par tous les navigateurs. C'est le langage standard pour la réalisation d'interfaces web riches.

L'utilisation de framework CSS permet, comme dans le cas du framework PHP, d'améliorer la maintenabilité du code, son évolution, et plus généralement le design de l'application, la rendant ainsi plus agréable à utiliser. De plus, la plupart de ces frameworks implémentent des notions comme le design responsive qui permettent à l'application d'être universelle et de s'adapter à tout type d'écran. Nous choisirons donc de comparer 4 frameworks, une fois encore parmi les plus connus, pour permettre de gagner du temps à la fois pour la production, la maintenance et l'évolution de l'application.

Il est difficile d'obtenir des statistiques concernant la popularité d'utilisation des frameworks CSS. Cependant, plusieurs noms reviennent très souvent sur les sites consacrés au design. Parmi eux, on retrouve : Bootstrap, UIKit, Foundation et Skeleton.

Bootstrap

Bootstrap est un framework front-end gratuit pour le développement web. Il contient plusieurs outils utiles à la réalisation de sites interactifs. Il permet entre autres de concevoir plus facilement un design responsive qui va permettre d'adapter l'affichage de l'application à tout type d'écran. Il fait partie des frameworks les plus populaires. Bootstrap est un framework très récent.

Il a été conçu en 2010 par deux développeurs de chez Twitter : Mark Otto et Jacob Thornton. Son but était de diminuer les coûts de maintenance dus aux incohérences entre les différentes bibliothèques existantes. Il est conçu pour être compatible avec tous les navigateurs majeurs tels que Google Chrome, Firefox, Safari ou encore Opera. Il fonctionne également en mode dégradé sur des navigateurs plus anciens.

Le concept de Bootstrap est basé sur les grilles. Chaque élément de l'interface se situe à l'intersection d'une ligne et d'une colonne. Cette grille sert d'armature à la totalité de

l'interface et est également très pratique en termes de design responsive. Le framework est publié sous licence MIT.

UIKit

UIKit est un framework de développement front-end de site web. Il a été développé par la société YOOTHEME. Il a pour avantage d'être plus léger et plus modulaire que les autres. Il est également facile à personnaliser et peut être étendu avec des thèmes.

Comme Bootstrap, UIKit fonctionne sur le système de grilles pour permettre d'adapter plus facilement l'interface réalisée à tous types d'écrans. Il est compatible avec tous les navigateurs récents et est distribué sous licence MIT.

Foundation

Foundation est un projet open-source créé par la société de design Zurb. Il fournit une grille de design HTML responsive ainsi que des composants d'interfaces standardisés (boutons, champs de texte, formulaires etc.) et réutilisables. Foundation utilise SASS comme préprocesseur CSS par rapport aux autres qui utilisent LESS, ce qui modifie la syntaxe standard des éléments à l'intérieur du framework. Il inclut également des composants et plugins JavaScript.

Foundation utilise, comme la plupart des frameworks CSS, le système de grille pour permettre de développer des applications responsives. Il est publié sous licence MIT.

Skeleton

Skeleton est également un projet open-source développé par Dave Gamache. Il permet de prototyper rapidement des interfaces et accélère également le développement des interfaces web en général, et ce quelle que soit leur dimension. Cependant, Skeleton est différent des autres solutions car il s'agit d'un kit de développement. En effet, il est très épuré. En effet, il ne donne qu'une grille responsive par rapport à ses éléments HTML. Il n'inclut pas de thème par défaut, il permet simplement, comme son nom l'indique, d'avoir un squelette d'interface autour duquel construire son propre design.

3.3.2.1 framework CSS utilisé

Le choix du framework CSS utilisé s'oriente vers le bootstrap. En effet, il s'agit d'un framework que j'ai déjà pratiqué lors de mes projets Web, ("Hotel Neptune") se qui me permet d'être plus rapide lors de son utilisation.

3.4. Rédaction du Cahier des Charges

La rédaction du cahier des charges est une partie importante de la réalisation du projet car il va permettre de fixer les limites de celui-ci. Il va également permettre de mettre par écrit les fonctionnalités attendues dans l'application. Une fois celui-ci terminé, il a été validé par mon maître de Stage. Aucune difficulté particulière n'a été rencontrée pour la rédaction de ce cahier car une étude préalable a été réalisée en amont et les besoins ont été clairement définis au début du stage.

3.5. Modélisation de l'application

La connaissance des fonctionnalités à implémenter est essentielle pour établir le diagramme de cas d'utilisations de l'application. Une fois encore, l'étude réalisée pour la compréhension de l'application et l'écriture du cahier des charges ont permis d'avoir des éléments solides pour lister les fonctionnalités à implémenter et faciliter la réalisation de ce diagramme de cas d'utilisation. Dans cette section, nous allons détailler les différents diagrammes de cas d'utilisation réalisés.

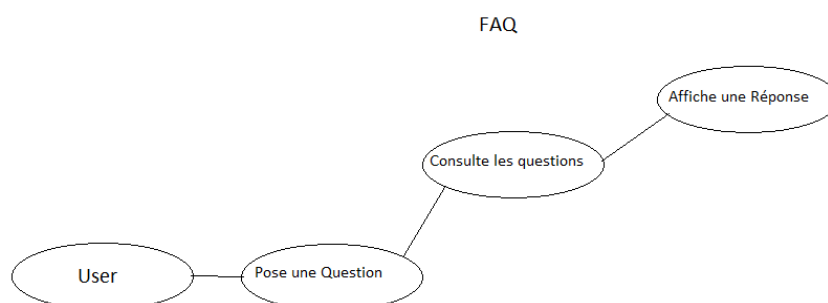


Figure 4 : Diagramme de cas d'utilisation par une utilisateur

La figure ci-dessus représente le diagramme de cas d'utilisation principal, à savoir celui concernant la FAQ. Comme nous pouvons le constater, l'acteur principal de ce diagramme est l'utilisateur. L'application doit donc pouvoir leur permettre de poser des questions afin d'obtenir des réponses, cela implémenter des fonctionnalités comme, la recherche.

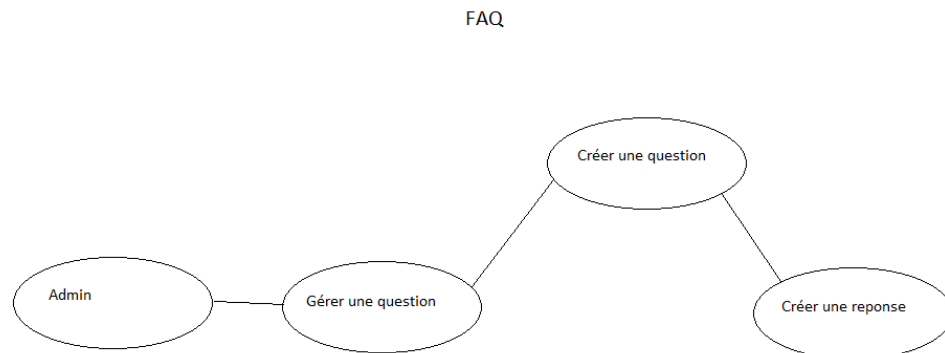


Figure 5 : Diagramme de cas d'utilisation par un Admin

La figure ci-dessus représente le diagramme de cas d'utilisation principal par un Admin, il possède toutes les fonctionnalités CRUD (Create Read Update Delete) en général. Ainsi que de gérer les utilisateurs pour soit les passer en Admin soit les supprimer en cas d'utilisation non conforme au règlement.

3.5.1 Modélisation de la base de données de l'application

La modélisation de la base de données est également une tâche très importante car il s'agit du cœur de l'application réalisée. L'application n'étant pas mise en place. Il va donc falloir mettre en place un schéma de la base de données qu'il faut créer, mais tout en restant conforme au cahier des charges.

Au cours de la phase de réalisation, plusieurs versions de la base de données ont été réalisées. En effet, certains processus (comme par exemple la mise en place de liste des différents types de question) sont plus compliqués que les autres et nécessitent une optimisation de la base de données pour gagner en efficacité.

Nous allons, dans cette partie, voir la version finale de la base de données et commenter.

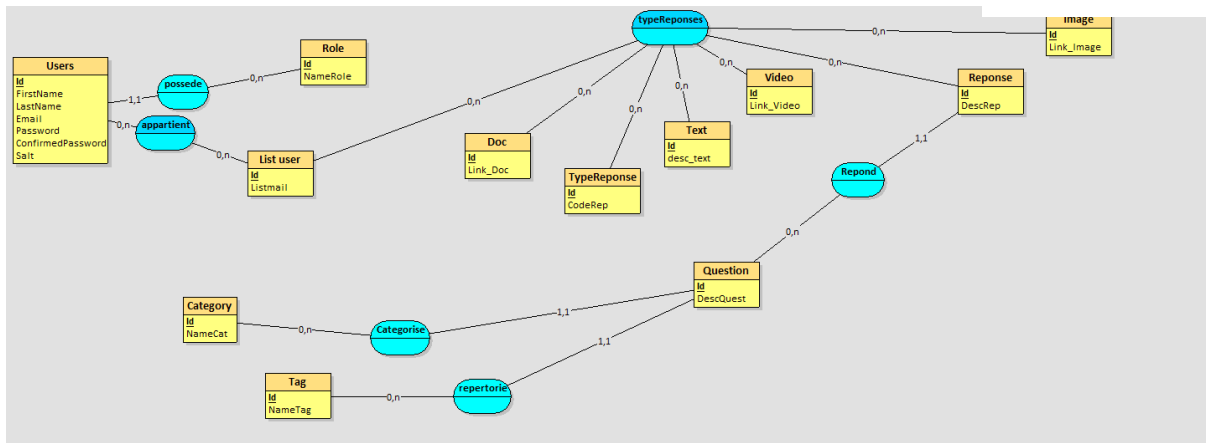


Figure 5 : Schéma de la base sous looping

La figure ci-dessus représente la base de données réalisée sous Looping. Nous pouvons constater qu'une question peut posséder plusieurs réponse.

Dans la table User nous pouvons constater qu'il y a un champ qui s'appelle le Salt en effet lors de la création d'un utilisateur nous devons pouvoir récupérer ce dernier. Le Salt est une partie de l'algorithme de Hachage de mot de passe, si on ne le récupère pas lors de la connexion nous ne pouvons pas vérifier et comparer les mots de passe afin de se login.

Pour Faciliter nos recherche nous créons une table Tag et Catégorie afin d'y reporter les informations dont nous avons besoin pour faire une recherche avec des listes.

3.6. Réalisation des maquettes

Les maquettes présentent de nombreux avantages pour le développement d'une application. Tout d'abord, elles sont une bonne manière de montrer aux clients à quoi ressemblera potentiellement la version finale de l'application. Elle va leur permettre d'émettre une première critique de l'application, non seulement au niveau de l'aspect global, mais également sur le contenu et la visibilité des fonctionnalités. Elles permettent donc d'avoir une vision un peu plus concrète de ce qui va certainement être réalisé, et permet de s'assurer que chaque partie comprenne bien ce qu'il se passe et soit d'accord sur la direction globale que prend le projet. Elles nous permettent également d'avoir un plan d'organisation précis des interfaces et de gagner du temps lors de l'élaboration de celles-ci. Enfin, au cas où certaines interfaces ne peuvent pas être réalisées, elles permettent à d'autres développeurs de réaliser les vues manquantes.

Les maquettes ont été réalisées uniquement pour certaines vues, le but ici n'étant pas de connaître chaque section de l'application dans le détail, mais de donner une vision globale de l'interface, qui sera reprise par la suite dans les autres vues.

logo

question

Réponse

tag

Catégorie

tdelos@Ookaidi.fr

tag

Catégorie

Recherche

Question 1

Question 2

Réponse 2

Figure 6 : Maquette Faq page Accueil connecté

J'ai proposé ce type d'accueil pour la Faq qui a été accepté par Gauthier il est assez clair et concis. Comme je travaille dans un modèle MVC et que la règle est qu'un modèle appartient à une vue j'ai dû créer une vue partielle afin de pouvoir utiliser plusieurs modèles dans cette page. Ainsi mon controller me permet donc de faire du requêtage sur les tables categories et tag.

Pour afficher les question de la base de données nous utilisons la façon suivante :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using Faq.Data;
using Faq.Models;

namespace Faq.Controllers
{
    1 référence
    public class FaqController : Controller
    {
        private readonly FaqContext _context;

        0 références
        public FaqController(FaqContext context)
        {
            _context = context;
        }

        // GET: Reponses
        0 références
        public async Task<IActionResult> Index()
        {
            var faqContext = _context.Reponse.Include(r => r.Doc).Include(r => r.Image).Include(r => r.Text).Include(r => r.video).Include(r => r.Question);
            return View(await faqContext.ToListAsync());
        }
    }
}
```

Figure 7 : Controller Faq

Ceci est mon controller de Faq il me permet de récupérer dans ma variable faqcontexte le contenu de mes tables. Ainsi dans vue je peux donc utilisé mes questions ainsi que toutes les réponses associées.

Si nous regardons du cotés de la vue on obtient donc :


```

@model IEnumerable<Faq.Models.Response>

@{
    ViewData["Title"] = "Index";
}

<h1>Consulter la Foire au Questions ! </h1>

@foreach (var item in Model)
{
    <div class="accordion" id="accordionExample">
        <div class="accordion-item">
            <h2 class="accordion-header" id="heading@item.Id">
                <button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapse @item.Id" aria-expanded="true" aria-controls="collapse @item.Id">
                    @Html.DisplayFor(modelItem => item.Question.Desc_Question)
                </button>
            </h2>
            <div>
                @if (item.DocId != null)
                {
                    <div id="basicAccordionCollapse @item.Id" class="accordion-collapse collapse" aria-labelledby="heading @item.Id" data-mdb-parent="#accordionExample" style="">
                        <div class="accordion-body">
                            <strong>
                                @Html.DisplayFor(modelItem => item.Doc.Link_Doc)
                            </strong>
                        </div>
                    </div>
                }

                @if (item.TextId != null)
                {
                    <div id="basicAccordionCollapse @item.Id" class="accordion-collapse collapse" aria-labelledby="heading @item.Id" data-mdb-parent="#accordionExample" style="">
                        <div class="accordion-body">
                            <strong>
                                @Html.DisplayFor(modelItem => item.Text.Desc_Text)
                            </strong>
                        </div>
                    </div>
                }

                @if (item.VideoId != null)
                {
                    <div id="basicAccordionCollapse @item.Id" class="accordion-collapse collapse" aria-labelledby="heading @item.Id" data-mdb-parent="#accordionExample" style="">
                        <div class="accordion-body">
                            <strong>
                                @Html.DisplayFor(modelItem => item.video.Link_Video)
                            </strong>
                        </div>
                    </div>
                }

                @if (item.Image != null)
                {
                    <div id="basicAccordionCollapse @item.Id" class="accordion-collapse collapse" aria-labelledby="heading @item.Id" data-mdb-parent="#accordionExample" style="">
                        <div class="accordion-body">
                            <strong>
                                @Html.DisplayFor(modelItem => item.Image.Link_Img)
                            </strong>
                        </div>
                    </div>
                }
            </div>
        </div>
    </div>
}

```

Figure 8 : View Faq

Grâce à des boucles Foreach nous pouvons les boucler sur les données de la base afin de tous les prendre en compte ensuite nous vérifions si il est null ou non en fonction de cela on les affiche ou pas. Le `@Html.DisplayFor` est une méthode de Razor afin de compacter du code html dans du C#.

Comme vue précédemment nous utilisons bien le framework Bootstrap afin de décorer notre page avec un modèle d’affichage en accordéon pour chacune de nous question. Cela nous permet d’avoir notre question en titre et avoir une réponse une fois que l’on clic dessus cela optimise la place sur votre écran.

Réponse

Video

Doc

Image

Text

Question

Create

Figure 9 : Maquette Faq page Creation réponse

Pour chaque ajout dans la base de données nous aurons un affichage qui ressemble à celui-ci bien sur les champs sont adaptés en fonction de la table où l'on ajoute les informations.

Nous n'arrivons pas directement sur cette page mais elle n'est pas représentée sous forme de maquette. En effet nous arrivons sur une page de détails où se trouve toutes les informations de la tables et à partir de cet endroits nous avons accès soit à la création soit à l'édition soit à la suppression d'une donné.

Pour afficher la creation de données nous utilisons la façon suivante :

```
// GET: Reponses/Create
0 références
public IActionResult Create()
{
    ViewData["DocId"] = new SelectList(_context.Doc, "Id", "Id");
    ViewData["DocName"] = new SelectList(_context.Doc, "Id", "link_Doc");
    ViewData["ImageId"] = new SelectList(_context.Image, "Id", "Id");
    ViewData["ImageName"] = new SelectList(_context.Image, "Id", "link_Img");
    ViewData["TextId"] = new SelectList(_context.Text, "Id", "Id");
    ViewData["TextName"] = new SelectList(_context.Text, "Id", "Desc_Text");
    ViewData["VideoId"] = new SelectList(_context.Video, "Id", "Id");
    ViewData["VideoName"] = new SelectList(_context.Video, "Id", "link_Video");
    ViewData["QuestionId"] = new SelectList(_context.Question, "Id", "Id");
    ViewData["QuestionName"] = new SelectList(_context.Question, "Id", "Desc_Question");

    return View();
}

// POST: Reponses/Create
// To protect from overposting attacks, enable the specific properties you want to bind to.
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
0 références
public async Task<IActionResult> Create([Bind("Id,VideoId,DocId,TextId,ImageId,QuestionId")] Response response)
{
    _context.Add(response);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

#pragma warning disable CS0162 // Unreachable code detected
    ViewData["DocId"] = new SelectList(_context.Doc, "Id", "Id", response.DocId);
#pragma warning restore CS0162 // Unreachable code detected
    ViewData["DocName"] = new SelectList(_context.Doc, "Id", "link_Doc", response.DocId);
    ViewData["ImageId"] = new SelectList(_context.Image, "Id", "Id", response.ImageId);
    ViewData["ImageName"] = new SelectList(_context.Image, "Id", "link_Img", response.ImageId);
    ViewData["TextId"] = new SelectList(_context.Text, "Id", "Id", response.TextId);
    ViewData["TextName"] = new SelectList(_context.Text, "Id", "Desc_Text", response.TextId);

    ViewData["VideoId"] = new SelectList(_context.Video, "Id", "Id", response.VideoId);
    ViewData["VideoName"] = new SelectList(_context.Video, "Id", "link_Video", response.VideoId);
    ViewData["QuestionId"] = new SelectList(_context.Question, "Id", "Id", response.QuestionId);
    ViewData["QuestionName"] = new SelectList(_context.Question, "Id", "Desc_Question", response.QuestionId);
    return View(response);
}
```

Figure 10 : Réponse Controller

Nous référençons dans un ViewData les différentes listes dont nous avons besoin pour créer une réponse. Dans le ViewData['DocName'] on récupère la liste de link_doc de la table Doc en fonction de son Id.

De la même manière que pour afficher une question nous avons un vue pour afficher Créer une réponse :

```

<div class="row">
  <div class="col-md-4">
    <form asp-action="Create">
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>
      <div class="form-group">
        <label asp-for="VideoId" class="control-label"></label>
        <select asp-for="VideoId" class="form-control" asp-items="ViewBag.VideoName">
          <option value="null"></option></select>
      </div>
      <div class="form-group">
        <label asp-for="DocId" class="control-label"></label>
        <select asp-for="DocId" class="form-control" asp-items="ViewBag.DocName">
          <option value="Null"></option></select>
      </div>
      <div class="form-group">
        <label asp-for="TextId" class="control-label"></label>
        <select asp-for="TextId" class="form-control" asp-items="ViewBag.TextName">
          <option value="Null"></option></select>
      </div>
      <div class="form-group">
        <label asp-for="ImageId" class="control-label"></label>
        <select asp-for="ImageId" class="form-control" asp-items="ViewBag.ImageName">
          <option value="Null"></option>
        </select>
      </div>
      <div class="form-group">
        <label asp-for="QuestionId" class="control-label"></label>
        <select asp-for="QuestionId" class="form-control" asp-items="ViewBag.QuestionName">
          <option value="Null"></option>
        </select>
      </div>
      <div class="form-group">
        <input type="submit" value="Create" class="btn btn-primary" />
      </div>
    </form>
  </div>
</div>

<div>
  <a asp-action="Index">Back to List</a>
</div>

@section Scripts {
  @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}

```

Figure 11 : View Réponse

Cela nous permet d'avoir des listes de sélections avec les données de la base afin de sélectionner la meilleure réponse possible en fonction de la question posée.

4. Conclusion

Ce stage m'est apparu comme une expérience très satisfaisante et enrichissante. Faire partie d'une équipe aussi dynamique et accueillante, travailler dans d'aussi bonnes conditions et pouvoir mettre en pratique le savoir que j'ai acquis sont autant de choses positives que j'en retire.

J'ai pu, grâce à ce stage, améliorer mes compétences sociales et élargir mes connaissances, en étant confronté à des professionnels.

J'ai également pu en apprendre plus sur les actions menées par un service informatique au sein d'une entreprise et leur rôle.

Le projet en lui-même a également représenté une bonne expérience pour moi. En effet, la programmation d'application web est une tendance de plus en plus adoptée par les entreprises de par ses nombreux avantages. Ainsi cette application m'a permis de former à une nouvelle méthode et un nouveau langage de programmation.

Le résultat de ce stage est donc une application réalisée dans le respect de plusieurs règles de bonne pratique de programmation web, qui permettent d'optimiser l'application. Parmi ces bonnes pratiques, on retrouve l'utilisation de framework ou encore l'utilisation d'un design simple et adapté au web.

Pour compléter ces bonnes pratiques et autres démarches adoptées, l'accent a été mis sur le choix de plusieurs outils connus et faciles d'apprentissage pour permettre une maintenance plus efficace.

4. Source

C# :

- https://fr.wikipedia.org/wiki/C_sharp
- <https://docs.microsoft.com/en-us/dotnet/csharp/>

Bootstrap :

- <https://getbootstrap.com/docs/5.2/getting-started/introduction/>

MVC :

- <https://www.lemagit.fr/definition/modele-vue-controlleur-MVC#:~:text=En%20d%C3%A9veloppement%20par%20programmation%20orient%C3%A9e,mod%C3%A8les%20de%20donn%C3%A9es%20sous%2Djaccents.>

Méthode Agile :

- <https://www.amalo-recrutement.fr/blog/methode-agile/>

Idkids :

- <https://corporate.idkids.com/groupe/qui-sommes-nous/>