



DOSSIER PROFESSIONNEL (DP)

<i>Nom de naissance</i>	➤ Serdjebi
<i>Nom d'usage</i>	➤ Serdjebi
<i>Prénom</i>	➤ Thomas
<i>Adresse</i>	➤ 3 rue Pointe à Pitre 13006 Marseille

Titre professionnel visé

Titre RNCP - Concepteur Développeur d'Applications

MODALITÉ D'ACCÈS :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.
Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente
obligatoirement à chaque session d'examen.

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.
Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel (DP)** dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

*[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels
du ministère chargé de l'Emploi]*

Ce dossier comporte :

- pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- une déclaration sur l'honneur à compléter et à signer ;
- des documents illustrant la pratique professionnelle du candidat (facultatif)
- des annexes, si nécessaire.

DOSSIER PROFESSIONNEL ^(DP)

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.

 <http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité	p.	5
▶ ChatApp	p.	6
▶ Boutique en Ligne	p.	11
▶ PDF Merger	p.	16
Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité	p.	18
▶ ChatApp	p.	18
▶ Boutique en Ligne	p.	22
	p.	
Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité	p.	27
▶ ChatApp	p.	27
▶	p.	
▶	p.	
Titres, diplômes, CQP, attestations de formation (facultatif)	p.	32
Déclaration sur l'honneur	p.	33
Documents illustrant la pratique professionnelle (facultatif)	p.	34
Annexes (Si le RC le prévoit)	p.	35

DOSSIER PROFESSIONNEL ^(DP)

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°1 - ChatApp

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

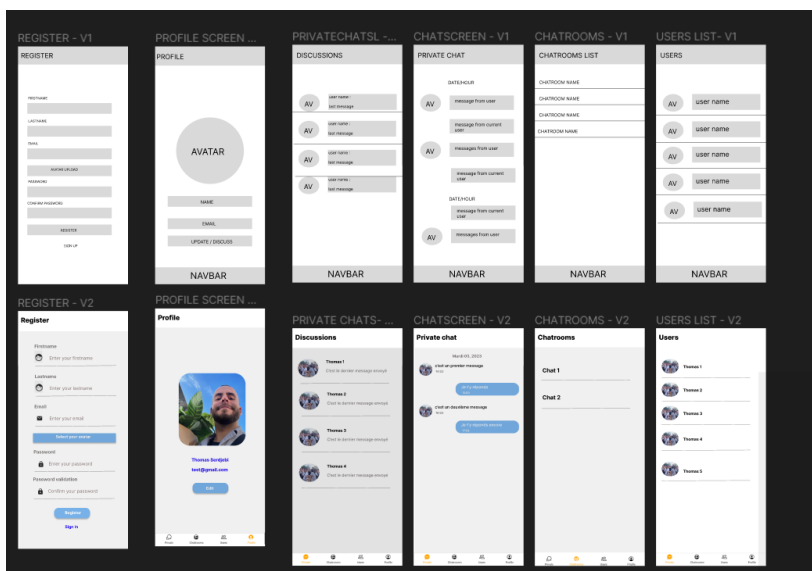
Présentation du projet

Le projet **ChatApp** est une **application de chat mobile** permettant à ses utilisateurs de s'inscrire en remplissant un formulaire avec leurs informations personnelles, se connecter, modifier leurs informations et discuter avec les autres utilisateurs via un service de discussions instantanées. En effet, les utilisateurs peuvent parler avec les autres utilisateurs dans des conversations privées ainsi que dans des salons de discussions. **Le projet ChatApp peut se diviser en 4 parties étant donné que c'est une application organisée en couches : une base de données, une API, l'application et le back-office administrateur.**

Tâches effectuées

Maquettage

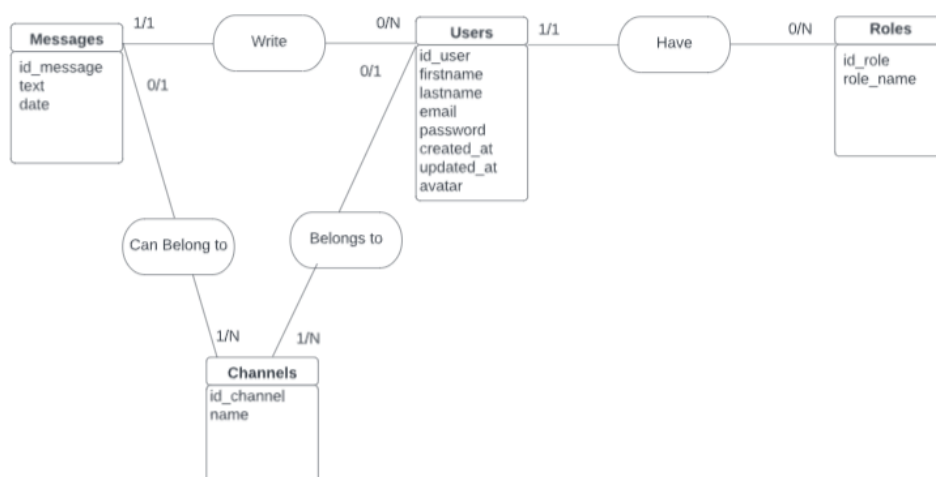
En premier lieu, **nous avons maqueté l'application à l'aide de l'outil Figma**. Nous avons d'abord établi un wireframe basse fidélité avec seulement les blocs clés des différents écrans. Par la suite, nous avons établi un wireframe détaillé de chaque écran : style des formulaires, style de la topbar, couleurs et taille des différentes typographies, barre de navigation en bas de l'écran, forme et affichage des avatars ainsi que des messages.



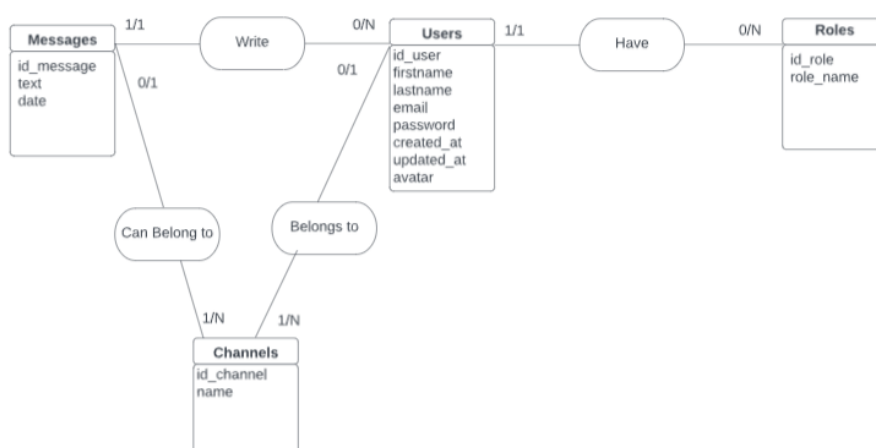
Modélisation et création de la base de données

En second lieu nous avons modélisé la base de données en suivant la méthode MERISE, et donc en établissant les modèles MCD, MLD, MPD. La base de données a ensuite été créée sur l'hébergeur PHPMYADMIN.

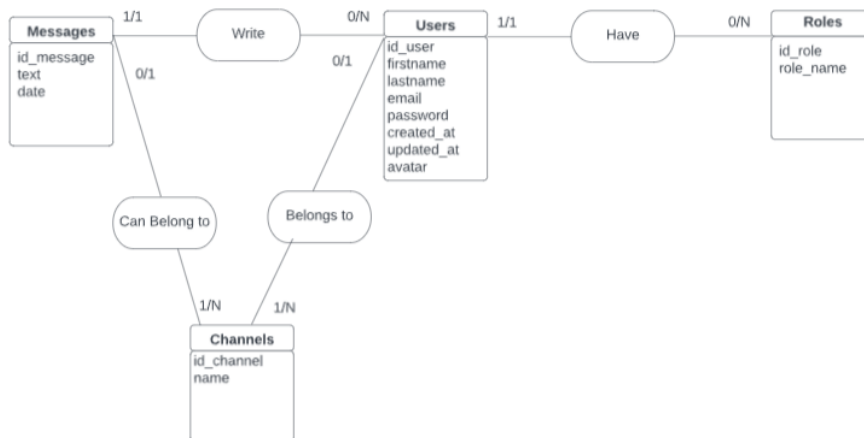
Modèle conceptuel de données :



Modèle logique de données :



Modèle physique de données :



Développement de l'API

Ensuite nous avons développé une API à l'aide du **framework ExpressJs** qui est utilisé pour la **création d'API et d'applications mobiles**, prenant en charge les détails essentiels du back-end comme la gestion des erreurs et le routage. Il est relativement simple à utiliser puisqu'il se base sur le langage Javascript et permet d'utiliser un seul langage pour le front-end et le back-end. Il est régulièrement révisé et mis à jour. Nous l'avons installé sous un environnement NodeJS. Dans cette API, **nous avons développé** :

- **des composants d'accès à la base de données dans le fichier database.js**
- des fichiers "Services" contenant les fonctions utilisant des requêtes SQL pour interagir avec la base de données,
- des fichiers "Controllers" contenant des fonctions de contrôles, **de sécurité contre les attaques XSS Cross Site Scripting/Injections SQL et de gestion des données entrantes par la vérification de la cohérence de la saisie de l'utilisateur en utilisant des regex par exemple**, avant d'appeler les fonctions contenant les requêtes SQL des fichiers "Services"
- des fichiers "Routes" paramétrant les routes à appeler, incluant pour la plupart des **middlewares qui permettent par exemple de gérer l'authentification ou le statut de l'utilisateur qui vient de se connecter (administrateur ou non)**.

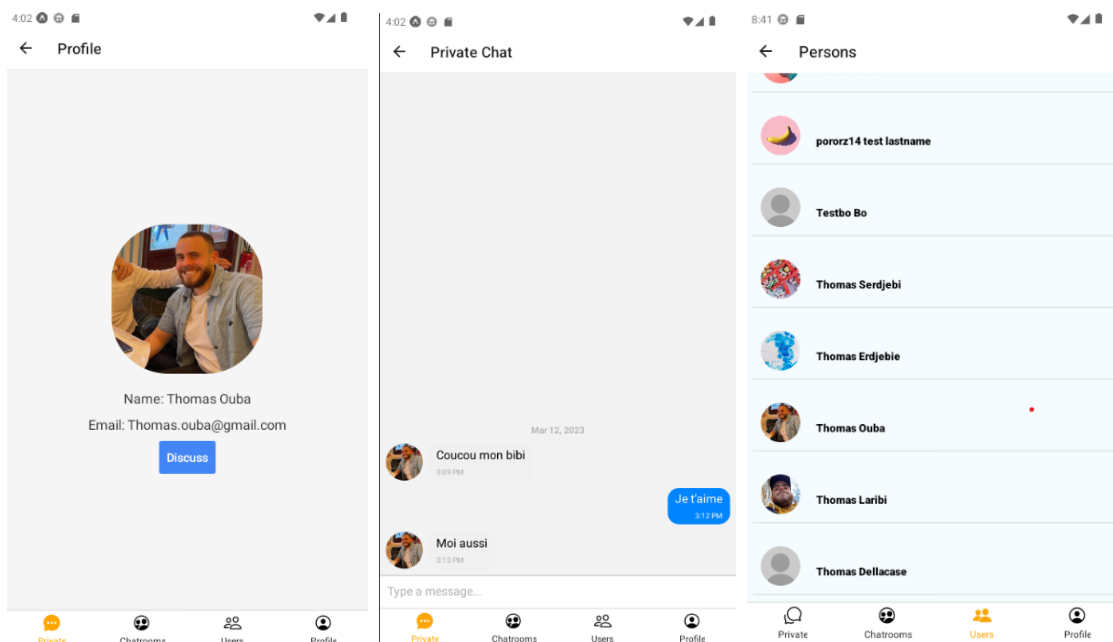
Nous avons également développé le fichier App.js. Ce fichier permet de configurer le serveur Express avec la gestion des routes ainsi que les autres fonctionnalités comme la mise en place des websockets permettant la communication en temps réel entre le client et le serveur de manière asynchrone et instantanée, essentiel pour la création d'une application mobile de discussions instantanée. Enfin, nous avons effectué les jeux d'essai à l'aide de l'outil POSTMAN.

Développement de l'application client

Pour poursuivre, nous avons développé l'application client. Nous avons installé l'environnement, défini une arborescence en amont des développements, puis modélisé une navigation. Nous avons commencé par créer un système de navigation temporaire dans le fichier App.js afin de faciliter l'accès aux différents écrans, puis **développer les différents écrans ci-dessous** :

- écran d'inscription
- écran de connexion
- écran de profil
- écran de modification du profil
- écran de la liste des utilisateurs
- écran de la liste des salons de discussions
- écran de la liste des discussions privées
- écran des discussions privées
- écran des discussions en salon
- barre de navigation

Tous ces écrans font appel aux routes de l'API pour interagir avec la base de données. **Les mesures de sécurité y sont également appliquées sur les données entrantes pour éviter les attaques XSS Cross Site Scripting et Injections SQL avant d'appeler les routes de l'API. On utilise la fonction sanitizeHtml sur les champs remplis par l'utilisateur et nous vérifions aussi la cohérence des données entrantes à l'aide de regex.** Nous avons testé les fonctionnalités sur chaque écran puis avons adapté le système de navigation existant afin de suivre la roadmap de l'utilisateur via des navigateurs imbriqués. Nous avons à nouveau effectué des tests puis amélioré le style des écrans et à nouveau effectuer des tests.



Développement du back office administrateur

Pour terminer, nous avons développé un back-office administrateur web à l'aide de React Admin qui utilise lui aussi les routes et les fonctionnalités développées dans l'API. En effet, grâce à React Admin qui facilite la création d'interface administrateur, cette couche du projet se compose de peu de choses :
un fichier App.js important les différentes librairies utilisées, composants utilisés et qui vient configurer la connexion à notre API à l'aide d'un data provider, ainsi que l'affichage du tableau administrateur
un fichier authProvider permettant la connexion au panel administrateur et de vérifier le rôle Administrateur de l'utilisateur
un fichier index.js qui permettra l'affichage du composant principal App dans le navigateur
des composants personnalisés permettant d'afficher, créer, modifier ou supprimer des éléments de la base de données.

Compétences visées

Dans le cadre de l'activité "Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité", j'ai pu acquérir les compétences suivantes :

- Maquetter une application
- Développer une interface utilisateur de type desktop
- Développer des composants d'accès aux données
- Développer la partie front-end d'une interface utilisateur web
- Développer la partie back-end d'une interface utilisateur web

2. Précisez les moyens utilisés :

- Visual Studio Code / Javascript / NodeJS / API Platform / Postman / Documentation en ligne
- Figma / React / React Native / Express / Expo / Gifted Chat / React Navigation / React Admin
- JsonWebToken / WebSockets / Android Studio / GitHub et GitKraken / LucidChat / Trello

3. Avec qui avez-vous travaillé ?

J'ai travaillé sur ce projet avec mes camarades de formation Thomas Delacase et Maxime Hadj.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Ecole La Plateforme_

Chantier, atelier, service ▶ Projet d'application de Chat

Période d'exercice ▶ Du : 01/01/23 au : 31/03/23

5. Informations complémentaires *(facultatif)*

Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n° 2 - Boutique en Ligne

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Présentation du projet

Dans le cadre de ma formation j'ai été amené à réaliser une boutique en ligne fictive. Le thème de cette boutique est la vente de briquets au détail et en gros. Le site devait comporter :

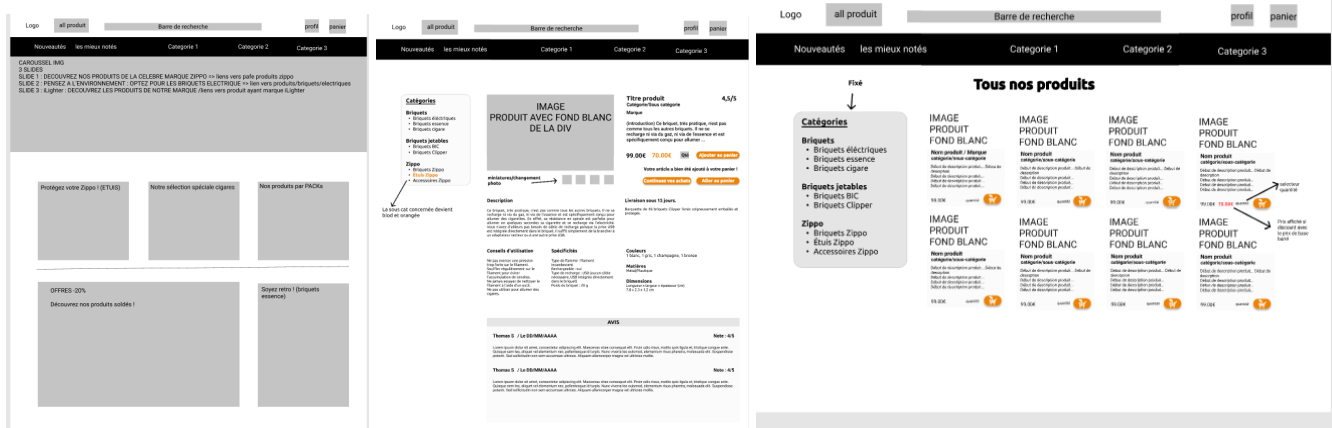
- une page d'index attractive mettant en avant les produits phares (le plus populaire, les nouveautés, et les produits les mieux notés)
- une page "Produits" comportant tous les produits de la boutique en ligne, ainsi que les catégories et sous catégories, permettant un tri d'affichage des produits dans la page
- un espace client complet permettant à l'utilisateur de s'inscrire, se connecter, consulter ses informations, modifier ses informations, consulter ses commandes
- un espace administrateur complet permettant à l'administrateur de créer, consulter, modifier, supprimer les produits, les catégories et sous catégories, les commandes, consulter les utilisateurs.

La partie front de ce projet est dynamique. Ce projet a été réalisé en groupe de 3 personnes. Il a été réalisé avec les langages suivants : **HTML / CSS / PHP / JAVASCRIPT**.

Tâches effectuées

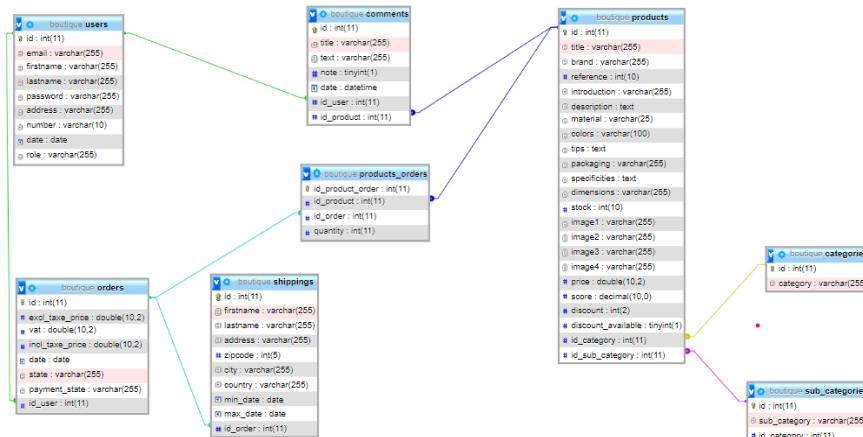
Maquettage du site

J'ai d'abord réalisé les **maquettes (wireframes) basse et haute fidélité** du parcours utilisateur afin d'avoir une vue d'ensemble des pages qui lui seront visibles, en définissant les couleurs principales utilisées, c'est-à-dire le blanc pour le fond, le noir pour le texte, le gris pour les blocs et l'orange pour les boutons.



Modélisation et création de la base de données

Par la suite, j'ai **modélisé la base de données**. Pour cela, j'ai utilisé le site **Ludichart**, un environnement de travail en ligne permettant de créer des diagrammes et notamment de modéliser des bases de données. J'ai d'abord conçu le **Modèle Conceptuel de Données** dans un premier temps puis dans un second temps le **Modèle Logique de Données**. J'ai ensuite **créé la base de données sur PHPMYADMIN**, un système en ligne de gestion de base de données. La base de données a été créée en engine = innoDB afin de maintenir l'intégrité référentielle de la base de données dont vous pouvez voir une vue dans le concepteur de PHPMYADMIN ci dessous :

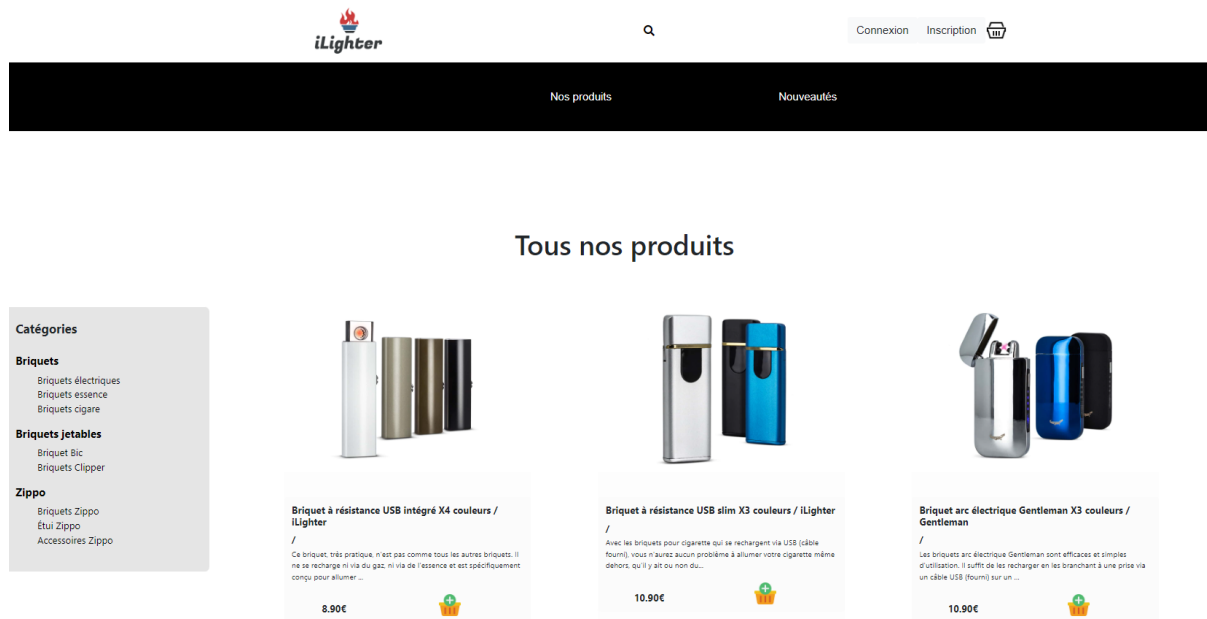


Création de la page Produits

J'ai développé la page Produits présentant tous les produits ainsi qu'un bloc affichant les catégories et sous catégories. **Le contenu de cette page est généré dynamiquement via le PHP et des requêtes SQL.** En effet, la page **inclut un fichier de traitement permettant d'appeler les fonctions de la classe Article(produits) et de retourner à la page des arrays(tableaux) contenant les informations des produits et des catégories nécessaires à l'affichage.** Grâce à **des boucles Foreach** qui parcourent ces tableaux, l'affichage est généré avec un code raccourci.

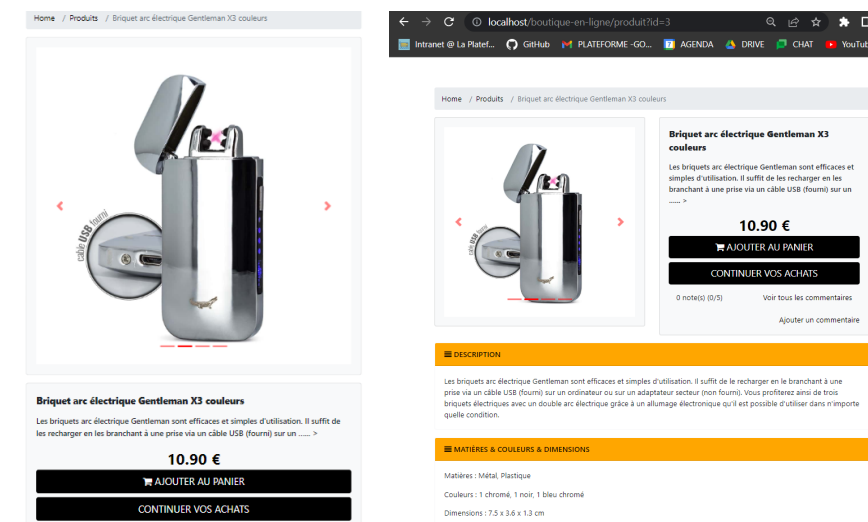
DOSSIER PROFESSIONNEL (DP)

Le **style** de la page a été géré avec du **CSS** et des **MediaQueries** pour le **responsive** et également la **méthode de mise en page flexbox** pour le placement des différents blocs. Le tri du contenu se fait via un **clik sur une catégorie ou une sous catégorie**, permettant de **filtrer les produits affichés également via les fonctions PHP de la classe Article contenant des requêtes SQL**. Le fichier de traitement inclus dans la page récupère en GET dans l'URL le paramètre de la catégorie ou sous catégorie choisie afin de renvoyer à la page les objets et informations que nous souhaitons y afficher. Chaque bloc produit comporte un lien qui amène vers la page Produit avec l'id du produit choisi en paramètre dans l'URL.



Création de la page Produit

J'ai développé la page Produit affichant tous les détails d'un produit et permettant de l'ajouter à son panier. Le **contenu de cette page est généré dynamiquement via la fonction de la classe Article** contenant une **requête SQL** appelée dans le **fichier de traitement inclus dans la page**. Ce fichier récupère en GET dans l'URL le paramètre id du produit choisi et appelle la fonction qui permet ainsi de renvoyer à la page un tableau contenant les informations du produit choisi. L'**affichage des blocs, le responsive et le style** de la page ont été générés à l'aide de **Bootstrap 5**, permettant une adaptation à tous les écrans.



Création de l'espace compte de l'utilisateur et sécurisation des formulaires

J'ai développé les pages concernant le **compte utilisateur (inscription, connexion, profil, mes commandes, modification des informations)**. Les formulaires ont été générés en HTML/CSS, le responsive de chaque page géré à l'aide des media queries afin de les adapter à tous les écrans. Les informations de l'utilisateur et de ses commandes, ainsi que le formulaire de modification du profil avec les champs pré-remplis avec les informations déjà enregistrées sont également générées à l'aide de **fonctions des classes correspondantes appelées dans les fichiers de traitement des pages**, renvoyant les tableaux contenant les informations nécessaires à la page. La page mon compte affiche les informations générales du profil, la dernière commande passée, ainsi que des liens pour accéder aux pages de modification des informations, du mot de passe, et la consultation de la dernière commande en détail ou de toutes les commandes.

Les formulaires sont en méthode POST et les champs sont sécurisés à l'aide de fonctions PHP `htmlspecialchars()` et `htmlspecialchars()` permettant de transformer les caractères éligibles en éléments HTML et de protéger le site contre les injections SQL et les attaques XSS Cross Site Scripting. Ces fonctions sont utilisées pour chacune des informations rentrées de chaque champ qui sont récupérées dans le fichier de traitement inclus dans les pages.

Mon Compte

Bonjour admin, bienvenue sur votre espace personnel

Mon profil

admin
admin
admin@gmail.com
102030405
admin

Modifier mes informations
Modifier mon mot de passe

Mes commandes

Dernière commande
Commande numéro 140
du 2022-04-13
13.08 euros

Afficher toutes mes commandes

Modifier les informations

Prénom
admin

Nom
admin

Mobile
102030405

Adresse
admin

Email
admin@gmail.com

Confirmez avec votre mot de passe

Sauvegarder

Compétences visées

Dans le cadre de l'activité "Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité", j'ai pu acquérir les compétences suivantes :

- Maquetter une application
- Développer des composants d'accès aux données
- Développer la partie front-end d'une interface utilisateur web
- Développer la partie back-end d'une interface utilisateur web

2. Précisez les moyens utilisés :

Visual Studio Code / Trello / Lucidchart / PHPMYADMIN / GitKraken et gitHub / Documentation en ligne
HTML / CSS / PHP / Bootstrap 5 / Freelogodesign / Media queries pour le responsive design

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec mes camarades Valentin Mathieu et Stéphane Mathieu

DOSSIER PROFESSIONNEL ^(DP)

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Ecole La Plateforme_

Chantier, atelier, service ▶ Projet de Boutique en Ligne fictive

Période d'exercice ▶ Du : 01/10/22 au : 31/12/22

5. Informations complémentaires *(facultatif)*

Activité-type 1

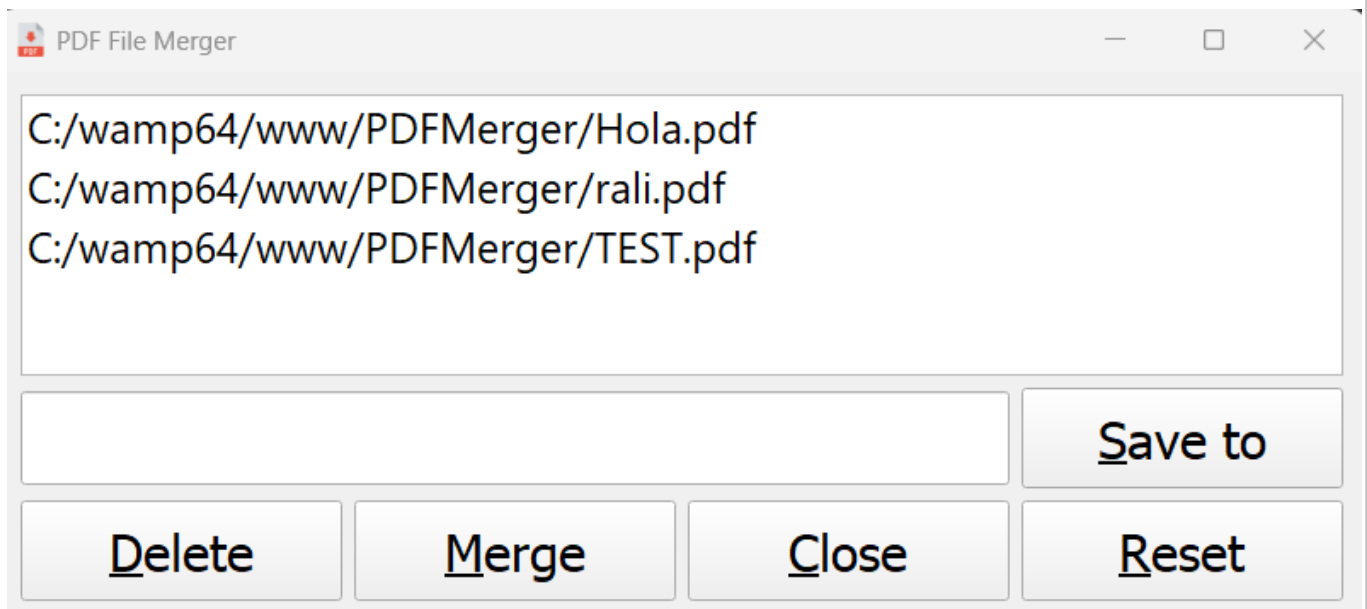
Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n° 3 - PDF MERGER

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Présentation du projet

Le projet PDF Merger est un projet Python d'application desktop permettant par du drag and drop de glisser des fichiers PDF dans l'interface de l'application et de les fusionner.



Tâches effectuées

Je me suis documenté sur Python et son installation dans mon environnement de développement. J'ai ensuite procédé aux installations. Ensuite j'ai importé les modules nécessaires au fonctionnement de l'application y compris PyQt5 pour créer l'interface utilisateur, PyPDF2 pour manipuler des fichiers PDF et d'autres modules pour la gestion des fichiers et les messages d'alerte.

DOSSIER PROFESSIONNEL ^(DP)

J'ai développé plusieurs **classes** :

- **ListWidget**, une sous classe de QListWidget représentant la zone de list où les fichiers PDF sont affichés et autorise un drag and drop et la sélection multiple des éléments de la liste
- **output_filed**, sous-classe de QLineEdit pour le champ d'entrée de l'emplacement du fichier de sortie fusionné
- **button**, sous classe de QPushButton pour les boutons de l'application
- **PDFApp**, la classe principale héritant de QWidget et représentant la fenêtre de l'application

J'ai aussi développé plusieurs **méthodes** :

- **initUi**, initialisant l'interface utilisateur avec les boutons et les zones de liste
- **deletSelected** et **clearQueue** permettant respectivement de supprimer les PDF sélectionnés et de vider la liste des fichiers
- **dialogMessage** permettant d'afficher la boîte de dialogue
- **getSaveFilePath** pour ouvrir une boîte de dialogue et sélectionner l'emplacement du fichier fusionné en sortie
- **populateFileName** qui utilise la méthode getSaveFilePath pour définir l'emplacement du fichier de sortie quand on clique sur le bouton Save To
- **mergeFile** qui vient fusionner les fichiers PDF quand on clique sur le bouton Merge

Compétences visées

Dans le cadre de l'activité "Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité", j'ai pu acquérir les compétences suivantes :

- Développer une interface utilisateur de type desktop

2. Précisez les moyens utilisés :

Visual Studio Code / Documentation en ligne / Python

3. Avec qui avez-vous travaillé ?

Seul

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Ecole La Plateforme_

Chantier, atelier, service ▶ Application Desktop

Période d'exercice ▶ Du : 01/04/23 au : 03/04/23

5. Informations complémentaires (facultatif)

Activité-type 2

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Exemple n° 1 - ChatApp

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Présentation du projet

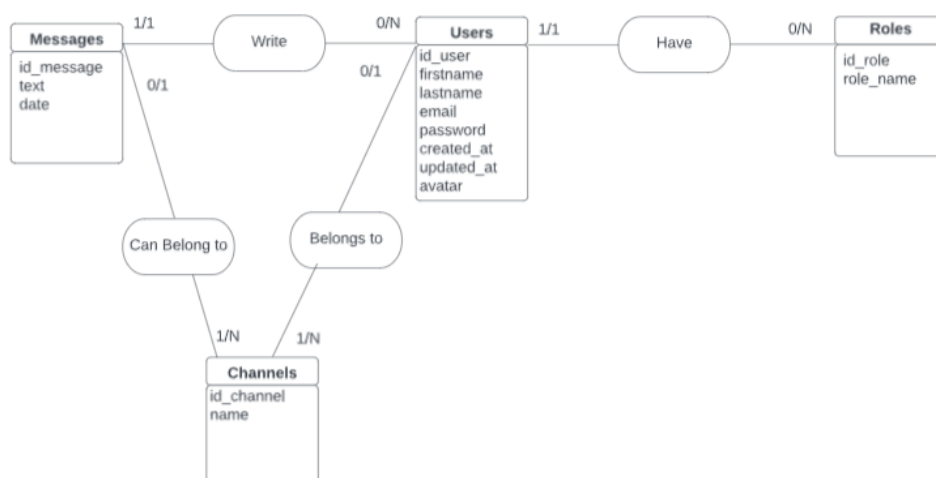
Ce projet est déjà présenté en pages 6,7 et 8. Le projet ChatApp est une application de chat mobile permettant à ses utilisateurs de s'inscrire en remplissant un formulaire avec leurs informations personnelles, se connecter, modifier leurs informations et discuter avec les autres utilisateurs via un service de discussions instantanées. En effet, les utilisateurs peuvent parler avec les autres utilisateurs dans des conversations privées ainsi que dans des salons de discussions. Le projet ChatApp peut se diviser en 4 parties étant donné que c'est une application organisée en couches : une base de données, une API, l'application et le back-office administrateur.

Dans le cadre de l'activité type numéro 2, je présente à nouveau les éléments permettant de démontrer l'acquisition des compétences qui y sont liées en y ajoutant quelques précisions :

Modélisation et création de la base de données

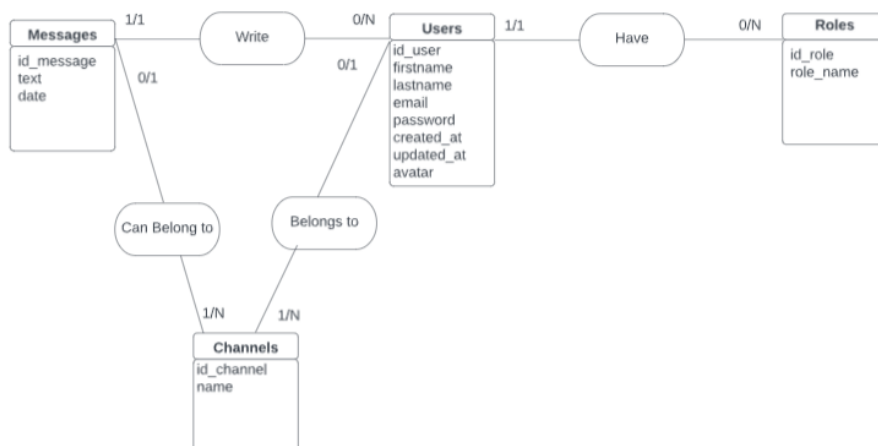
Nous avons modélisé la base de données en suivant la méthode MERISE, et donc en établissant les modèles MCD, MLD, MPD. La base de données a ensuite été créée sur l'hébergeur PHPMYADMIN.

Modèle conceptuel de données :

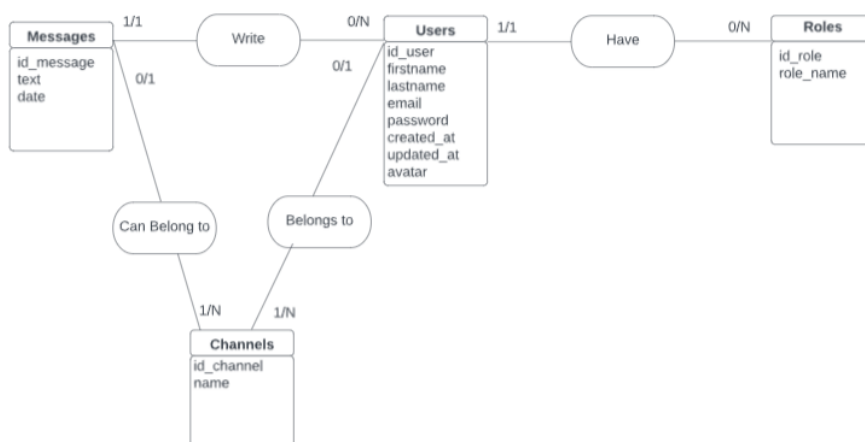


DOSSIER PROFESSIONNEL (DP)

Modèle logique de données :



Modèle physique de données :



Développement de l'API

Ensuite nous avons développé une API à l'aide du framework ExpressJs qui est utilisé pour la création d'API et d'applications mobiles, prenant en charge les détails essentiels du back-end comme la gestion des erreurs et le routage. Il est relativement simple à utiliser puisqu'il se base sur le langage Javascript et permet d'utiliser un seul langage pour le front-end et le back-end. Il est régulièrement révisé et mis à jour. Nous l'avons installé sous un environnement NodeJS. Dans cette API, **nous avons développé** :

- des composants d'accès à la base de données dans le fichier `database.js` : on notera que toutes les constantes permettant l'accès à la base de données sont en fait stockées dans un fichier `.env` afin d'éviter leur publication sur GitHub et assurer la sécurité de la base de données.

```
1 const { createPool } = require('mysql2');
2
3 const pool = createPool ({
4   port: process.env.DB_PORT,
5   host: process.env.DB_HOST,
6   user: process.env.DB_USER,
7   password: process.env.DB_PASS,
8   database: process.env.MYSQL_DB
9
10
11 });
12
13 module.exports = pool ;
```

- des fichiers "Services" contenant les fonctions utilisant des requêtes SQL pour interagir avec la base de données : à noter que ces requêtes SQL ont des placeholders(paramètres de substitutions) permettant de traiter les données entrantes envoyées par l'utilisateur comme des données brutes et non pas comme des requêtes SQL, sécurisant ainsi l'application contre les attaques de type injections SQL :

```
pool.query(
  'insert into users (firstname, lastname, email, password, role_id, created_at, updated_at, avatar) values (?, ?, ?, ?, ?, ?, ?, ?)',
  [
    data.firstname,
    data.lastname,
    data.email,
    data.password,
    defaultRole,
    created,
    data.updated_at,
    data.imageUrl
  ],
```

- des fichiers "Controllers" contenant des fonctions de contrôles, de sécurité contre les attaques XSS Cross Site Scripting/Injections SQL et de gestion des données entrantes par la vérification de la cohérence de la saisie de l'utilisateur en utilisant des regex par exemple, avant d'appeler les fonctions contenant les requêtes SQL des fichiers "Services"
- des fichiers "Routes" paramétrant les routes à appeler, incluant pour la plupart des middlewares qui permettent par exemple de gérer l'authentification ou le statut de l'utilisateur qui vient de se connecter (administrateur ou non).

Nous avons également développé le fichier `App.js`. Ce fichier permet de configurer le serveur Express avec la gestion des routes ainsi que les autres fonctionnalités comme la mise en place des websockets permettant la communication en temps réel entre le client et le serveur de manière asynchrone et instantanée, essentiel pour la création d'une application mobile de discussions instantanée. Enfin, nous avons effectué les jeux d'essai à l'aide de l'outil POSTMAN.

DOSSIER PROFESSIONNEL ^(DP)

Compétences visées :

Dans le cadre de l'activité "Concevoir et développer la persistance des données en intégrant les recommandations de sécurité" j'ai pu acquérir les compétences suivantes :

- Concevoir une base de données
- Mettre en place une base de données
- Développer des composants dans le langage d'une base de données

2. Précisez les moyens utilisés :

- Visual Studio Code / Javascript / NodeJS / API Platform / Postman / Documentation en ligne

3. Avec qui avez-vous travaillé ?

Avec mes camarades Thomas Delacase et Maxime Hadj.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Ecole La Plateforme_

Chantier, atelier, service ▶ Projet de boutique en ligne

Période d'exercice ▶ Du : 01/10/22 au : 31/12/22

5. Informations complémentaires (facultatif)

Activité-type 2

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Exemple n° 2 - Boutique en Ligne

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Présentation du projet

Dans le cadre de ma formation j'ai été amené à réaliser une boutique en ligne fictive. Le thème de cette boutique est la vente de briquets au détail et en gros. Le site devait comporter :

- une page d'index attractive mettant en avant les produits phares (le plus populaire, les nouveautés, et les produits les mieux notés)
- une page "Produits" comportant tous les produits de la boutique en ligne, ainsi que les catégories et sous catégories, permettant un tri d'affichage des produits dans la page
- un espace client complet permettant à l'utilisateur de s'inscrire, se connecter, consulter ses informations, modifier ses informations, consulter ses commandes
- un espace administrateur complet permettant à l'administrateur de créer, consulter, modifier, supprimer les produits, les catégories et sous catégories, les commandes, consulter les utilisateurs.

La partie front de ce projet est dynamique. Ce projet a été réalisé en groupe de 3 personnes. Il a été réalisé avec les langages suivants : **HTML / CSS / PHP / JAVASCRIPT**.

Le projet ayant déjà été présenté en pages 11, 12, 13 et 14 et dans le cadre de l'activité type numéro 2, je présente à nouveau les éléments permettant de démontrer l'acquisition des compétences qui y sont liées en y ajoutant quelques précisions.

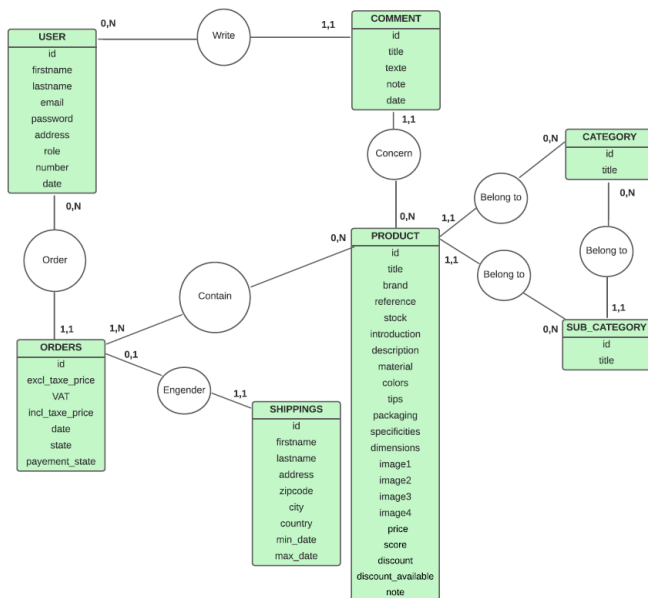
Tâches effectuées

Modélisation et création de la base de données

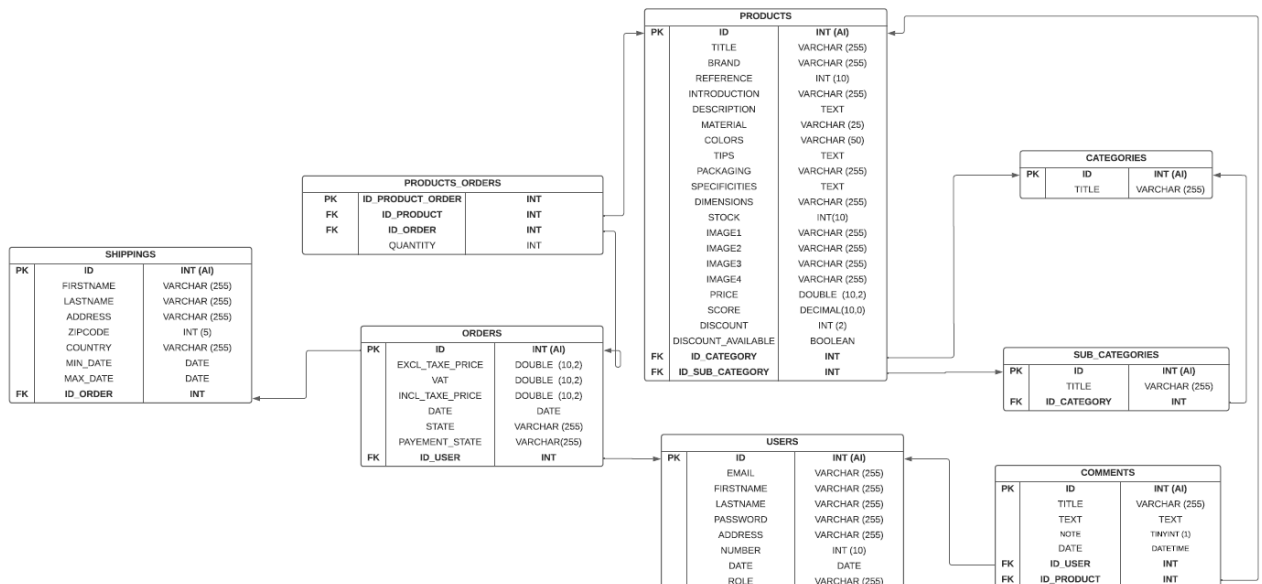
J'ai modélisé la base de données. Pour cela, j'ai utilisé le site **Ludichart**, un environnement de travail en ligne permettant de créer des diagrammes et notamment de modéliser des bases de données.

DOSSIER PROFESSIONNEL (DP)

J'ai d'abord conçu le **Modèle Conceptuel de Données** :

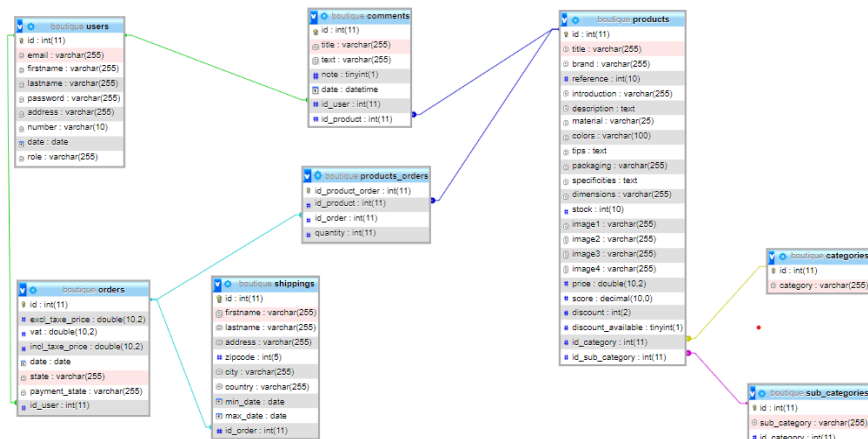


dans un second temps le **Modèle Logique de Données** :



DOSSIER PROFESSIONNEL (DP)

J'ai ensuite **créé la base de données sur PHPMYADMIN**, un système en ligne de gestion de base de données. La base de données a été créée en engine = innnoDB afin de maintenir l'intégrité référentielle de la base de données dont vous pouvez voir une vue dans le concepteur de PHPMYADMIN ci dessous :



J'ai également développé le **composant d'accès à la base de données** dans un fichier **database.js** qui crée une **classe permettant d'accéder à la base de données** et pourra être **instancié dans les différents fichiers de classe php** après une **première instanciation dans la classe Http**:

```
1 <?php
2 namespace Models;
3 use PDO;
4
5 class Database{
6
7     public static function getPdo(): PDO{
8         $pdo = new PDO('mysql:host=localhost;dbname=boutique;charset=utf8', 'root', '', [
9             PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
10            PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
11        ]);
12         return $pdo;
13     }
14
15 }
16
17
18 ?>
```



```
<?php
namespace Models;
class Http{
    //redirect(index.php)//
    public static function redirect(string $url): void {

        header("Location: $url");
        exit();
    }
}
?>
```

J'ai également développé **les fichiers de classe en PHP** contenant les différentes fonctions exécutant des **requêtes SQL** rédigée en **Programmation Orientée Objet** avec des **requêtes préparées** assurant la **sécurité contre les injections SQL** :

```
public function UpdateProfil($email,$firstname,$lastname,$address,$number,$id){

    $data = [
        'email'=>$email,
        'firstname'=>$firstname,
        'lastname'=>$lastname,
        'address'=>$address,
        'number'=>$number,
        'id'=>$id,
    ];

    $query = " UPDATE users SET email = :email, firstname = :firstname, lastname =:lastname, address = :address, number = :number WHERE id = :id";
    $update = $this->pdo->prepare($query) ;
    $update->execute($data);
}
```

Compétences visées :

Dans le cadre de l'activité "Concevoir et développer la persistance des données en intégrant les recommandations de sécurité" j'ai pu acquérir les compétences suivantes :

- Concevoir une base de données
- Mettre en place une base de données
- Développer des composants dans le langage d'une base de données

2. Précisez les moyens utilisés :

HTML / CSS / PHP / Bootstrap 5 / Visual Studio Code

3. Avec qui avez-vous travaillé ?

Avec mes camarades Valentin Mathieu et Stéphane Mathieu

DOSSIER PROFESSIONNEL ^(DP)

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Ecole La Plateforme_

Chantier, atelier, service ▶ Projet de boutique en ligne

Période d'exercice ▶ Du : 01/10/22 au : 31/12/22

5. Informations complémentaires *(facultatif)*

Activité-type 3

Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Exemple n° 1 - ChatApp

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Présentation du projet

Ce projet est déjà présenté en pages 6,7 et 8 et 18,19, 20. Le projet ChatApp est une application de chat mobile permettant à ses utilisateurs de s'inscrire en remplissant un formulaire avec leurs informations personnelles, se connecter, modifier leurs informations et discuter avec les autres utilisateurs via un service de discussions instantanées. En effet, les utilisateurs peuvent parler avec les autres utilisateurs dans des conversations privées ainsi que dans des salons de discussions. Le projet ChatApp peut se diviser en 4 parties étant donné que c'est une application organisée en couches : une base de données, une API, l'application et le back-office administrateur.

Dans le cadre de l'activité-type 3, "Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité", je vais à nouveau présenter les éléments démontrant l'acquisition des compétences avec certaines précisions. Ce projet a été réalisé en groupe de 3, avec mes camarades Thomas Delacase et Maxime Hadj.

Tâches effectuées sur la gestion de projet

- **Définition du cahier des charges et des spécifications fonctionnelles**

Le cahier des charges a été défini pour chaque couche de l'application : la base de données, l'API, l'application client et le back office administrateur. Pour la base de données, il consistait de définir les différentes tables et leurs propriétés et de les modéliser selon modèles MCD, MLD, MPD. Pour l'API, cela consistait à définir les différentes routes et fonctionnalités qui seraient appelées durant l'utilisation de l'application. Pour l'application client, cela a consisté à définir les écrans et les fonctionnalités de l'API qui y seraient associées. Pour la back office administrateur, cela a consisté à définir les fonctionnalités de l'API auxquelles l'administrateur ferait appel, et quelles informations relatives à l'application il devrait gérer.

- **Planification des tâches et des phases de tests**

Nous avons également établi un plan de développement qui suit le plan suivant : modélisation et création de la base de données, développement de l'API et de ses fonctionnalités, développement de l'application client et enfin développement du back office administrateur. En effet, l'application ne pouvant pas fonctionner sans base de données et sans API nous avons donc choisi un ordre logique en s'occupant d'abord du développement des éléments essentiels avant de développer les interfaces où ils seraient intégrés. Des phases de tests étaient appliquées après chaque création de nouvelle fonctionnalité, puis des tests de fonctionnements globaux suivis d'améliorations et de nouveaux tests avant de conclure le projet.

- **Répartition des tâches et outil de suivi**

Pour la répartition des tâches, nous avons établi un plan qui permettrait à chacun de pouvoir toucher à tout. Les tâches comme le maquetage ou le choix des technologies utilisées et de la modélisation de la base de données ont été faites en commun pour que l'on ait tous une idée identique du projet. En revanche, dans le développement pur, nous nous sommes répartis les tâches afin de pouvoir tous les trois travailler sur nos compétences : par exemple, chacun de nous à développer différentes fonctionnalités de l'API (services, contrôles, routes), chacun de nous à développer les différents écrans de l'application client. L'outil Trello nous a permis de planifier ces tâches mais aussi de nous les affecter de façon équitable en termes de charge de travail.

- **Versionning**

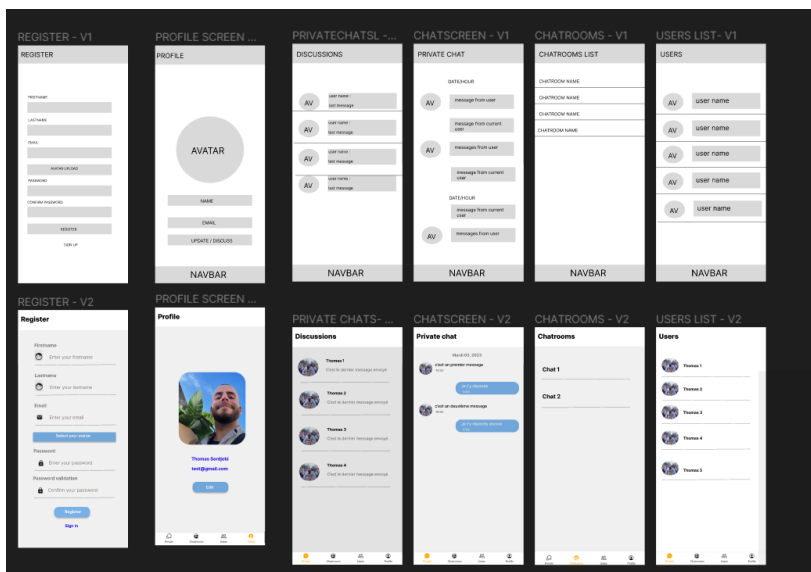
Pour la gestion du versionning, nous avons utilisé Git. En effet, nos repository sont stockés sur Github. Nous avons créé une branche à chaque développement d'une fonctionnalité / d'un fichier / d'un écran liés par la trame logique que suit l'utilisateur.

Nous effectuons plusieurs "commits" avec des sur les branches spécifiques puis, après avoir achevé un développement nous effectuons des tests, apportons les corrections et enfin nous fusionnons la branche spécifique à la branche principale.

Quand la majorité des développements étaient aboutis et afin d'apporter les dernières modifications suite à des tests globaux, nous avons changé de méthode et créé des branches avec le nom des écrans ainsi que la date de modification. Quand tous les tests étaient terminés, nous fusionnions ces branches spécifiques datées à la branche principale.

● Maquettage

Nous avons maqueté l'application à l'aide de l'outil Figma. Nous avons d'abord établi un wireframe basse fidélité avec seulement les blocs clés des différents écrans. Par la suite, nous avons établi un wireframe détaillé de chaque écran : style des formulaires, style de la topbar, couleurs et taille des différentes typographies, barre de navigation en bas de l'écran, forme et affichage des avatars ainsi que des messages.



● Déploiement

Nous avons développé notre application à l'aide d'Expo. Le déploiement d'une application est un processus qui permet de rendre l'application mobile disponible pour les utilisateurs, et Expo est justement un ensemble d'outils qui facilite le développement d'applications en utilisant notamment React Native. Nous avons donc installé Node.js et Expo CLI dans notre projet développé avec Visual Studio Code, puis créé le projet grâce à la commande `expo init` dans le terminal qui vient structurer le projet de base avec les fichiers et dossiers préconfigurés. Au cours du développement, nous avons utilisé la commande `'expo start'` qui permet de démarrer le serveur de développement Expo et permet de tester l'application via notre émulateur Android Studio (permettant de générer une image de différents appareils de différentes tailles iPhone, Android, tablettes...) ou directement sur notre téléphone en scannant le QR code généré dans le terminal et ouvrant notre application dans l'application mobile Expo Go. Expo offre aussi la possibilité de publier l'application et de la rendre disponible via le service d'hébergement d'Expo, de la distribuer par la génération de fichiers binaires ou de les soumettre aux stores respectifs.

Tâches effectuées sur le développement

Base de données :

- Modélisation
- Création

API :

- Documentation sur la construction d'une API en javascript avec Express
- Installation de l'environnement
- Développement du fichier App.js configurant le serveur et les routes associées
- Développement des fichiers de Services(fonctions et requêtes SQL sécurisées), Controllers(gestion des erreurs et sécurité) et Routes(définition des routes) pour quelques tables de la base de données
- Tests des routes sur l'outil Postman

Application :

- Maquettage
- Documentation sur NodeJS, Expo, Android Studio, React, React Navigation et tous les autres outils
- Installation de l'environnement
- Développement d'une navigation temporaire permettant une accessibilité facilitée aux écrans pour les développements
- Développement d'une partie des écrans et des fonctionnalités associées appelant l'API en intégrant des contrôles de sécurité sur les données entrantes envoyées par les formulaires comme la sécurisation des champs et le contrôle de cohérence des données envoyées par des regex
- Développement d'une navigation définitive
- Déploiement sur Expo
- Tests par phases successives

Back office administrateur :

- Documentation sur React Admin
- Installation de l'environnement
- Développement de la plateforme web
- Tests par phases successives

DOSSIER PROFESSIONNEL ^(DP)

Compétences visées

Dans le cadre de l'activité type-3 "Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité" m'a permis d'acquérir les compétences suivantes :

- Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement
- Concevoir une application
- Développer des composants métier
- Construire une application organisée en couches
- Développer une application mobile
- Préparer et exécuter les plans de tests d'une application
- Préparer et exécuter le déploiement d'une application

2. Précisez les moyens utilisés :

Visual Studio Code / Trello / Lucidchart / PHPMYADMIN / GitKraken et gitHub / Documentation en ligne
HTML / CSS / PHP / Bootstrap 5 / Freelogodesign / Media queries pour le responsive design

3. Avec qui avez-vous travaillé ?

Avec mes camarades Thomas Delacase et Stéphane Mathieu

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Ecole La Plateforme_

Chantier, atelier, service ▶ Application mobile de chat

Période d'exercice ▶ Du : 01/01/23 au : 31/03/23

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL ^(DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Titre professionnel Développeur Web et Web Mobile	Académie d'Aix-Marseille - délivré par l'Etat	06/09/2022

Déclaration sur l'honneur

Je soussigné(e) Thomas Serdjebi ,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis
l'auteur(e) des réalisations jointes.

Fait à Marseille le 15/07/2023

pour faire valoir ce que de droit.

Signature :



Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL ^(DP)

ANNEXES

(Si le RC le prévoit)