

RAPPORT DE STAGE

GAROT Thomas

Développeur Web et Web Mobile
Session Novembre 2020



Sommaire

I.	AVANT-PROPOS	3
A.	REMERCIEMENTS	3
II.	PRÉSENTATION	4
A.	L'ENTREPRISE	4
B.	LE SITE.....	4
C.	LE STAGIAIRE : MON PROFIL ET PARCOURS	5
D.	LE PROJET.....	6
1.	Résumé du projet	6
2.	Déroulement	7
3.	Contexte de développement.....	8
4.	Compétences Couvertes	8
E.	DEMARCHE GENERALE	9
III.	ANALYSE.....	10
A.	CAHIER DES CHARGES.....	10
1.	Expression des besoins	10
2.	Spécifications fonctionnelles détaillées	10
B.	DIAGRAMME DE CAS D'UTILISATION.....	13
1.	Représentation graphique.....	13
2.	Description textuelle des cas d'utilisation.....	14
C.	DIAGRAMME DE NAVIGATION.....	15
D.	DIAGRAMME DE SEQUENCE SYSTEME.....	16
IV.	LA BASE DE DONNEES	17
A.	DEMARCHE UTILISEE	17
B.	DIAGRAMME DE CLASS (DCL).....	18
C.	SCHEMA DE LA BASE DE DONNEES.....	19
D.	CREATION DE LA BASE DE DONNEES.....	19
E.	STRUCTURE ET CREATION DES TABLES	20
V.	CONCEPTION DU SITE	24
A.	DIAGRAMME DE SEQUENCE DETAILLE	24
B.	DIAGRAMME D'ACTIVITE	25
VI.	DEVELOPPEMENT DU SITE	26
A.	TECHNOLOGIES UTILISEES	26
B.	ENVIRONNEMENT DE DEVELOPPEMENT.....	27
C.	ARCHITECTURE DU SITE.....	28
D.	LE STYLE RESPONSIF	29
1.	Page Web.....	29
2.	Code HTML.....	31
3.	Code CSS	33
E.	FONCTIONNALITE CHANGEMENT D'ORDRE	35
1.	Page Web.....	35
2.	Code jQuery.....	36
3.	Code PHP.....	37
F.	SECURITE : LES FAILLES XSS ET SQL	42
G.	DIFFICULTES RENCONTREES.....	44
VII.	GIT	45
VIII.	RECHERCHE ANGLOPHONE	46

IX. CONCLUSION	47
X. ANNEXES	48
A. STATISTIQUES.....	48
1. <i>Vues Page Statistiques</i>	48
2. <i>Code</i>	51
B. MONDAY.COM.....	54
1. <i>Vue Application Monday</i>	54
2. <i>Code</i>	54
C. JOINTURE SQL	57

I. Avant-Propos

A. Remerciements

Merci à Laura ma conjointe pour m'avoir soutenue depuis le début dans ce projet de reconversion et qui m'a même exempté de ménage tout du long 😊 ❤️. Merci à mes amis qui m'ont encouragé et félicité ce qui m'a encore plus motivé, et plus particulièrement à Baptiste qui a pris le temps de m'aider quand je ne voyais plus d'issue. Merci à Transition Pro qui a accepté de financer mon projet et m'a aidé à rectifier une erreur dans mon dossier qui aurait pu tout faire tomber à l'eau, ils m'ont permis d'aller au bout. Merci à Pascal et Sébastien qui m'ont accompagné tout au long de la formation et transmis un peu de leur savoir. Merci à Neil et Bertrand qui m'ont fait confiance et m'ont offert l'opportunité de réaliser un stage où j'ai beaucoup appris, j'ai beaucoup apprécié travailler avec eux.

II. PRÉSENTATION

A. L'entreprise

QRMiam a été créé en 2020 à Lyon lors du premier confinement français dû à la pandémie de la Covid-19 afin de répondre au besoin de distanciation sociale dans les structures de restauration.

Les deux associés ont des parcours différents : Bertrand Dodane est un entrepreneur aguerri avec plus de 20 ans d'entrepreneuriat, Neil Benoit est un étudiant en finance d'entreprise qui a longtemps exercé en tant que serveur en restauration.

Cette association permet à QRMIAM de bénéficier d'expertises métiers, opérationnelles et financières.

Cette entreprise propose

Les supports sont marqués au laser en Bourgogne par leur entreprise partenaire NWS-Laser qui est spécialisée depuis plus de 20 ans dans les applications logicielles et matérielles de traçabilité.

B. Le site

QRMiam propose une solution simple et économique qui permet de digitaliser les menus de tous types de structures (restaurant, bar, café, hôtel, etc...) grâce à l'affichage d'un QR code permettant à leurs clients de visualiser les cartes sur leur téléphone.

Il y a 2 abonnements disponibles, « Lite » et « Premium » avec un paiement mensuel ou annuel. Le site propose de s'inscrire gratuitement et de profiter d'un essai de 7 jours permettant de découvrir toutes les fonctionnalités Premium. A l'issue de ces 7 jours, l'utilisateur doit s'abonner à l'une des deux options afin de pouvoir continuer à utiliser le service et que ses clients puissent continuer à voir ses cartes en ligne. Le site propose également de fournir des supports d'affichage pour QR code que les utilisateurs peuvent choisir dans la « boutique ».

Ils auront accès à une interface leur permettant entre autres de :

- Gérer leurs documents, l'ordre et l'heure de leur affichage
- Produire et télécharger leur QR code
- Visualiser les statistiques des vues de leur page

Les clients pourront accéder aux cartes des utilisateurs en scannant le QR Code qui les dirigera vers la page appropriée.

C. Le stagiaire : mon profil et parcours

Issue d'un bac général et technologique, j'ai poursuivi mes études en licence STAPS, dans le sport, où je me suis spécialisé vers le management de structures sportives.

A 24 ans je suis parti faire un voyage de 5 ans en auto-stop, j'ai visité plus de 200 villes dans une trentaine de pays sur 4 continents. J'ai travaillé à plusieurs reprises dans différent pays pour pouvoir continuer mon voyage, j'ai appris à parler anglais, espagnol et portugais (et des fragments de pleins d'autres langues) et j'ai surtout beaucoup ouvert mon esprit et mis au défi ma détermination.

« *I can try, I can fail... but I can never fail to try* »

Je suis revenu en France afin de débuter ma vie professionnelle, avec l'idée mais pas la certitude de me lancer dans le développement web. J'ai fait beaucoup de recherches sur les différents métiers qui pourraient me convenir et particulièrement sur le métier de développeur qui, petit à petit, semblait être une évidence. Je vivais déjà avec ce type de mentalité :

« *Never spend 6 minutes doing something by hand when you can spend 6 hours to automate it* »

Le métier de développeur web « Full Stack » est celui qui me semblait le mieux me correspondre. Les compétences matchaient les nécessités des différentes réalisations personnelles que j'aimerais développer (celles qui ont au départ attiré mon attention vers ce métier). Et surtout j'ai toujours aimé faire les choses par moi-même et être capable de savoir faire le plus de choses possibles.



Le coût des formations et le temps nécessaire à cet apprentissage ne me permettaient pas de me lancer facilement. J'ai donc préparé patiemment le moment de cette reconversion, réunissant petit à petit les conditions nécessaires à la mise en route de ma future carrière, j'ai commencé à me former seul pour augmenter le plus possible mes chances de réussite. J'ai pu intégrer M2i Formation en vue de décrocher un titre RNCP qui m'aiderait à trouver du travail. Me voici maintenant en vue de la ligne d'arrivée, qui est paradoxalement aussi la ligne d'un nouveau départ.

D. Le projet

1. Résumé du projet

Lors de mon stage, ma première mission a été de participer à l'amélioration du site web de QRMiAm sous WordPress, en collaboration avec un « développeur WordPress ». Je devais l'assister pour la refonte graphique, et de mon côté améliorer les fonctionnalités existantes défaillantes et en ajouter de nouvelles.

J'ai donc créé différents modules dans un thème enfant sur WordPress pour remodeler entièrement la page de « gestion des documents », « mon QR code », « statistiques » et « profil public ». Pour le développement j'ai utilisé les langages HTML, CSS, jQuery et PHP. Quelques fonctionnalités à développer étaient la gestion des documents en ligne, de l'affichage du nom d'établissement sur le profil public, afficher les statistiques des vues sur des périodes données dans des graphiques, gérer les types d'abonnement, etc...

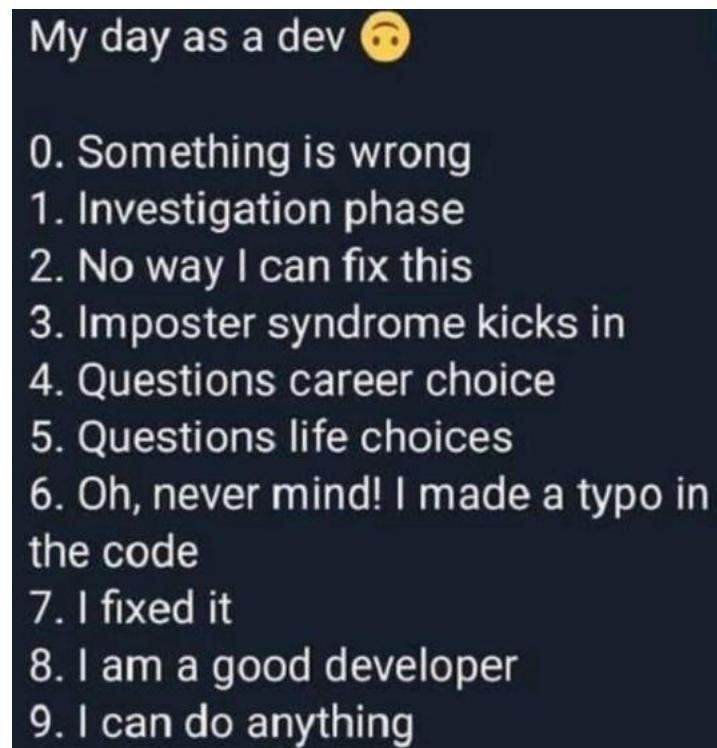
La deuxième mission a été de lier la base de données du site au CRM (Customer Management System) Monday et d'implémenter des tâches d'automatisation afin que le tableau des utilisateurs sur Monday soit automatiquement mis à jour à chaque modification des données d'un utilisateur sur le site.

J'ai dû parcourir la documentation de Monday pour pouvoir utiliser son API et apprendre à utiliser GraphQL, un langage de requête utilisé par cette API. J'ai utilisé les « hooks » de WordPress et de WooCommerce pour déclencher les mises à jour de données au moment où la création d'un utilisateur ou les modifications d'informations étaient effectuées.

J'ai eu le temps de remplir toutes mes missions initiales et presque toutes celles qui se sont rajoutées. Je devais être force de proposition et autonome car personne ne pouvait m'assister en PHP. Le site web est fonctionnel juste à temps pour la réouverture des restaurants !

2. Déroulement

- Découverte du site existant, explication des missions principales (1/2 jour)
- Découverte/appropriation du code PHP gérant les pages sur lesquelles je dois travailler (7 ½ jours)
- Implémentation des différentes fonctionnalités demandées (17 jours)
- Découverte du CRM Monday, de son API et du langage GraphQL (5 jours)
- Implémentation de la liaison CRM - Base de données (6 jours)
- Réglages des derniers bug, implémentations de fonctionnalités complémentaires (6 jours)



3. Contexte de développement

Le stage s'est effectué à distance entièrement, le contact avec les membres de l'entreprise se faisait principalement par téléphone, réunion Google Meet ou sur le CRM Monday. Je travaillais donc avec mon matériel.

Bertrand mon tuteur, a des connaissances techniques avancées en informatique mais pas en développement PHP ou JavaScript et Henry, chargé de la refonte graphique sur WordPress, a des connaissances sur le fonctionnement de Wordpress mais très légères en PHP ou jQuery. J'ai donc dû travailler en autonomie en me livrant à un travail perpétuel de recherche, d'autant plus que mes connaissances initiales de WordPress et de son fonctionnement étaient très limitées.

4. Compétences Couvertes

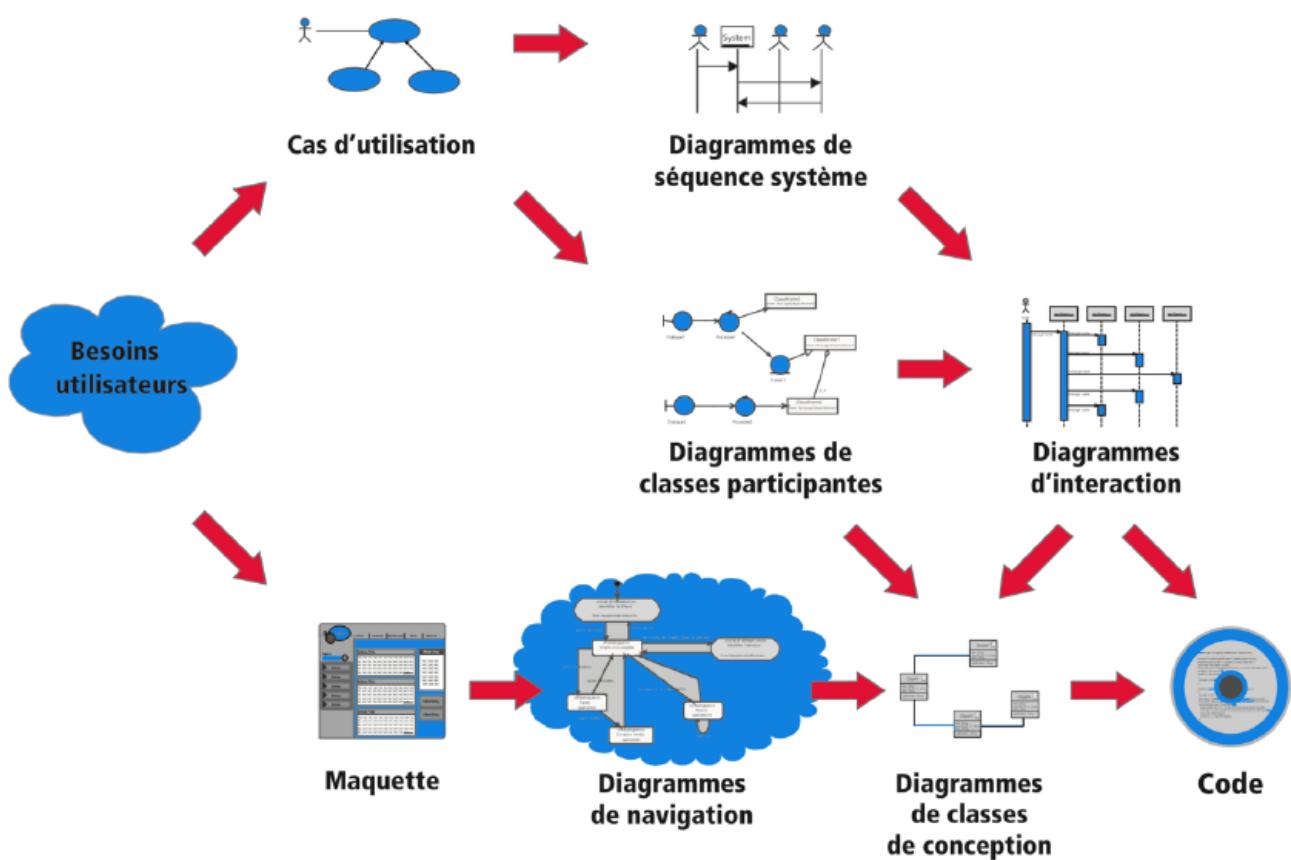
Compétences	Activité	Couverte ?
Maquetter une application	1	OK
Réaliser une interface utilisateur web statique et adaptable	1	OK
Développer une interface utilisateur web dynamique	1	OK
Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce	1	NON
Créer une base de données	2	OK
Développer les composants d'accès aux données	2	OK
Développer la partie back-end d'une application web ou web mobile	2	OK
Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce	2	NON

E. Démarche générale

Avant toute étape de développement, il est indispensable d'anticiper et d'analyser l'ensemble des besoins utilisateurs.

Afin de mener à bien ce projet et de répondre correctement au besoin de l'entreprise, j'ai choisi de suivre, avec quelques aménagements, la démarche découverte lors de ma formation chez M2i, qui est préconisée par Pascal Roques dans son livre « UML2 - Modéliser une application web », publié chez Eyrolles.

Voici le schéma de cette démarche :



III. Analyse

A. Cahier des Charges

1. Expression des besoins

« Assister le Designer Web pour la refonte de l'esthétique du site internet.
Utiliser le code existant et le modifier afin d'ajouter les fonctionnalités nécessaires à la gestion des menus des utilisateurs et de leurs informations personnelles, en respectant la nouvelle charte graphique.
Lier la base de données du site QRMIam à notre CRM Monday.com.
Apprendre à connaître Monday et pouvoir implémenter des tâches d'automatisation. »

2. Spécifications fonctionnelles détaillées

Pour ce projet 4 acteurs ont été définis :

- Administrateur : les gestionnaires du site
- Visiteur : Restaurateur visitant le site avant de s'inscrire
- Utilisateur : Restaurateur inscrit sur le site ayant ou non un abonnement en cours
- Consommateur : Hôtes des structures de restauration qui visualisent les profils publics

Général :

- Mettre en place un thème enfant permettant la mise à jour du thème sans perte de données
- Indentation, appropriation et factorisation du code existant qui réunit le contenu de 4 pages web ainsi que toutes les fonctions de requêtes en 1 seul fichier de 2100 lignes
- Suppression des variables ou fonctions non utilisées ainsi que du code mis en commentaire.

Abonnement :

- Mise en place dès l'inscription d'une période d'essai de 7 jours avec un accès aux fonctionnalités Premium et passer le statut à « non-inscrit » à l'issue des 7 jours
- Modifier automatiquement le statut d'un abonné Premium ou Lite à « Perdu » après un mois ou un an suivant son abonnement
- Supprimer la bannière et les documents au format pdf si l'utilisateur souscrit à un abonnement Lite (option disponible seulement pour les Premium)

Mes informations :

- Modification du formulaire de mise à jour des informations : le séparer en 2 formulaires, « Informations personnelles » et « Changement mot de passe » avec un titre et un bouton de validation pour chaque formulaire

Mes documents en ligne :

- Réparation de la fonction de modification de la couleur du nom d'établissement
- Création des fonctions de modification du nom d'établissement, de sa taille et de sa police.
- Modification de la requête de mise à jour qui met à jour tous les champs de la page même en cas de modification unique
- Retirer la ligne d'affichage de statistique et la mettre dans la page statistique après modification
- Gérer la suppression des documents qui ne s'enlèvent que de la base de données mais pas du fichier de stockage
- Créer une nouvelle fonction de gestion de l'ordre des documents car celle existante est défaillante. L'ordre doit être mis à jour en temps réel à chaque insertion, suppression ou modification sans nécessité de cliquer sur le bouton « mise à jour »
- Simplification des champs d'heures d'affichage
- Réorganiser la structure des formulaires de mise à jour et de suppression de document, réunir les deux formulaires en un
- Redesigner la page entière pour qu'elle soit en accord avec le nouveau design du site
- Rendre la page responsive pour les écrans pc, les tablettes et les téléphones en utilisant les « Media Queries »
- Afficher une interface différente en fonction du type d'abonnement, avec une interface de réglage plus avancé pour les profils premium
- Rediriger l'utilisateur vers la page « abonnement » lorsqu'il visite la page « mon compte » et qu'il n'a pas d'abonnement valide

Mon QR Code personnel :

- Ajouter un bouton de téléchargement du QR code et un lien vers la visualisation du profil public

Statistiques :

- Ajouter une ligne de statistique qui donne le nombre de vue du jour et le total depuis l'inscription (visible pour tous)
- Ajouter un tableau de réglage pour les statistiques avancées (visible mais désactivé pour les utilisateurs « Lite ») permettant :
 - Une visibilité par jour ou par période
 - Un accès rapide à des périodes clé grâce à des boutons (Depuis le début, 30 derniers jours, Mois en cours)
 - Un affichage des statistiques par utilisateur (pour les administrateurs)
- Afficher les statistiques (flouté pour les Lite)
 - Quotidiennes ou par période dans des graphiques différents
 - Moyennes sur la période avec 2 choix de graphique possible
- Les valeurs utilisées dans les champs doivent être sauvegardées après la validation du formulaire
- La date de fin ne peut être inférieure à la date de début

Profil public :

- Affichage de la bannière du site lorsqu'on est sur le profil d'un utilisateur Lite, non-inscrit ou non-existent et de celle que l'utilisateur a choisi s'il est Premium
- Afficher les documents si l'utilisateur du profil à un abonnement valide, sinon afficher un message d'erreur
- Affichage du nom d'établissement dans le format choisi par l'utilisateur
- Vérification de la validité d'abonnement de l'utilisateur à chaque login sur son compte et chaque visite de son profil par un consommateur

MySQL :

- Harmonisation des données de la base de données car les modifications de certains champs de formulaire dans Wordpress ont également modifié le nom ou le format de la métadonnée correspondante
- Faire passer tous les profils « Gratuit » à « Essai 7 jours » et modifier la date d'abonnement au jour de la modification

CRM Monday.com :

- Découvrir le CRM Monday et le fonctionnement de son API et du « mode développeur »
- Lier la base de données du site WordPress au CRM :
 - Automatiser l'ajout d'un utilisateur dans Monday à chaque nouvelle inscription
 - Automatiser la mise à jour des infos de l'utilisateur à chaque modification dans le site
 - Automatiser l'ajout de la méta donnée monday_user_id dans la base de données à chaque nouvelle inscription
- Ajout de la méta donnée monday_user_id pour tous les profils existants
- Mise à jour de tous les profils existants sur Monday :
 - Ajout du user_id et de la date d'abonnement
 - Mise à jour des données existantes

B. Diagramme de Cas d'Utilisation

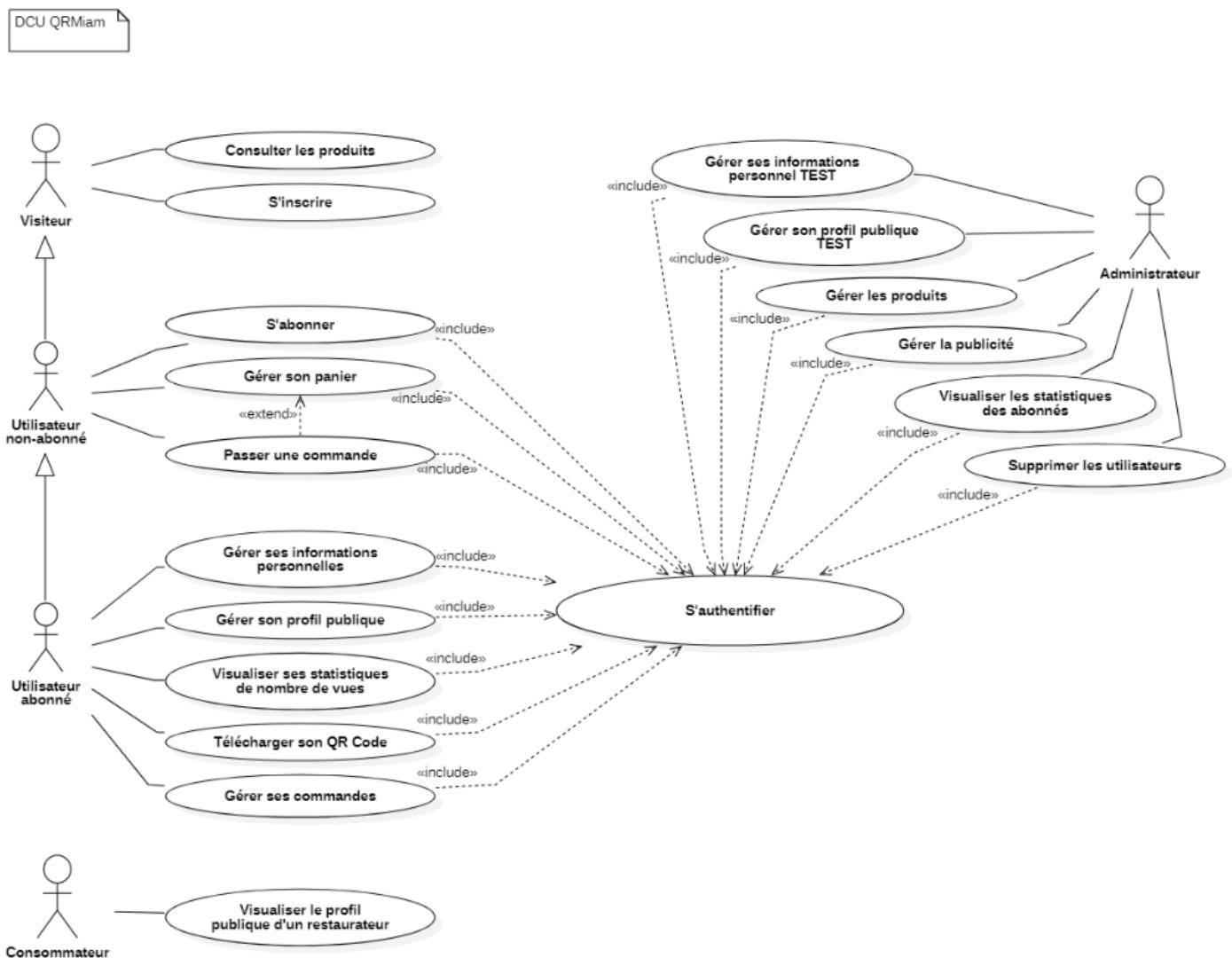
1. Représentation graphique

Un diagramme de cas d'utilisation (DCU) représente un ensemble de cas d'utilisation (CU).

Un CU est un **scénario**, ou un déroulement concret, entre l'utilisateur et le système.

UN DCU est un moyen **simple** d'exprimer les besoins d'un utilisateur, en illustrant les **fonctionnalités** de l'application (recueillir, analyser et organiser les besoins) dont il aura besoin, via des **schémas très compréhensibles**.

Ici j'ai identifié 5 **acteurs** représentant les utilisateurs **intervenant sur le système**, et j'en ai déduit les différents **Cas d'Utilisation** :



2. Description textuelle des cas d'utilisation

Scénario nominal :

Déroulement idéal des actions (quand tout va pour le mieux).

Scénarios alternatifs :

Éventuelles étapes différentes liées aux choix de l'utilisateur (cas des étapes liées à des conditions).

Scénarios d'exception : déroulement causé par la présence d'un événement anormal. Par exemple, lorsqu'une recherche de client ne trouve aucun client correspondant aux critères fournis.

Scénario d'erreur : correspond à un cas d'erreur.

Post-conditions : indiquent un résultat vérifiable après l'arrêt du cas d'utilisation témoignant du bon fonctionnement.

Etape	Description
Identification CU	Inscription sur le site
Pré-conditions	Avoir cliqué sur le bouton inscription qui ouvre le formulaire d'inscription sous forme de pop-up
Scénario nominal	<ul style="list-style-type: none">• Le visiteur renseigne les données demandées (identité, adresse, nom d'établissement, etc...)• il clic sur le bouton « valider »• Le système effectue un control de surface pour vérifier si toutes les données obligatoires sont rentrées et si les contraintes de saisies sont respectées• Si OK, le système vérifie si le nom d'utilisateur n'existe pas déjà dans la BD• Si OK, Les données sont enregistrées dans BD
Scénario d'erreur du cas nominal	<p><u>Control de surface</u> : Le visiteur reste sur la même fenêtre, un message d'erreur est noté près des champs où il a des erreurs</p> <p><u>Control userlogin</u> : Le visiteur reste sur la même fenêtre, un message d'erreur est noté sur la page</p>
Post-conditions	<ul style="list-style-type: none">• Un email est envoyé au nouvel utilisateur• L'utilisateur est redirigé vers la page d'authentification

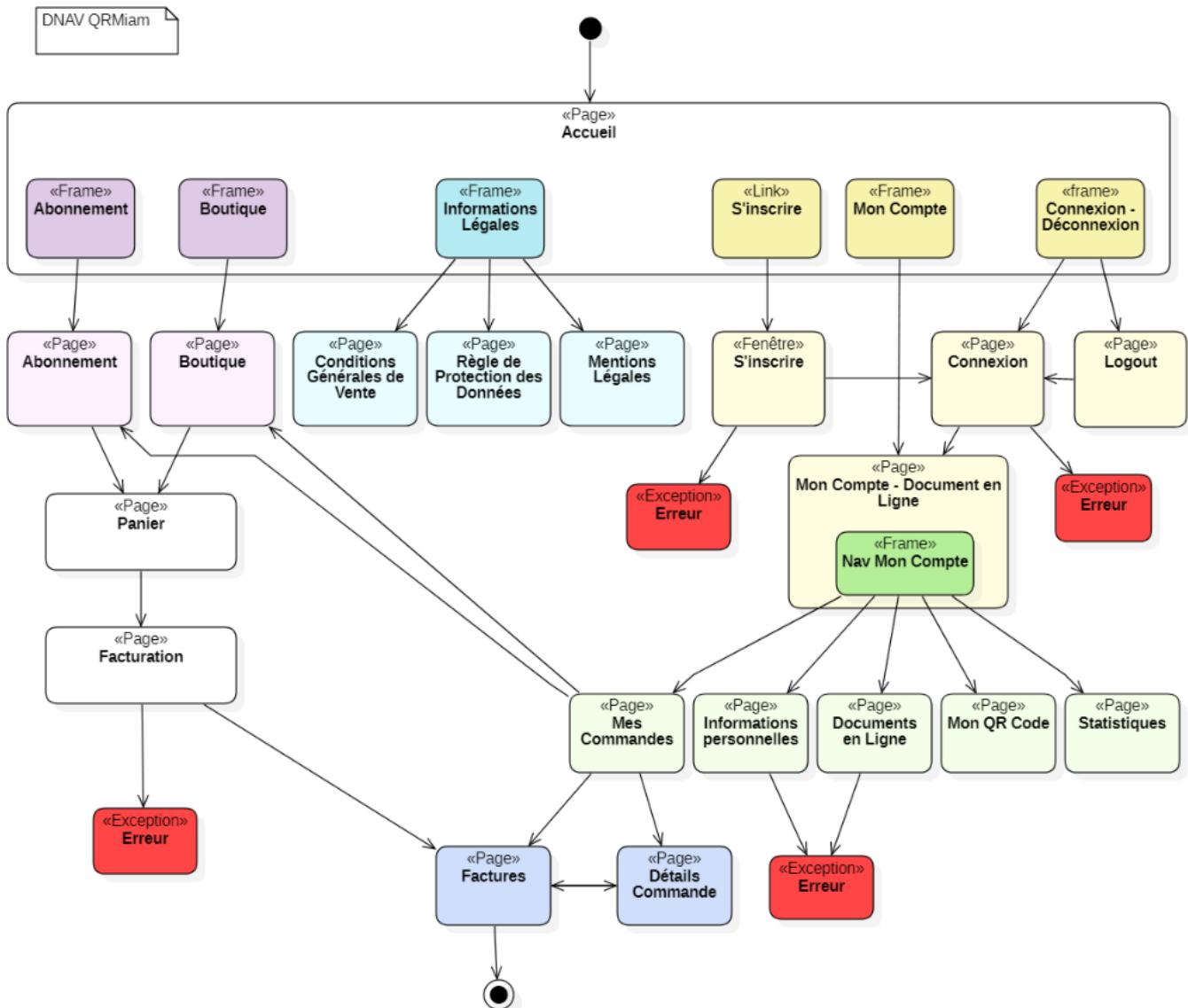
C. Diagramme de Navigation

Le diagramme de navigation (DNav) nous permet de bien comprendre la **navigation** entre les différentes pages de l'application, représentées par les différentes **transitions** d'un écran (IHM : Interface Homme Machine) à un autre.

Une navigation a un début et éventuellement une fin :

- Un rond plein représente un pseudo-état initial
- Un rond plein encerclé représente un pseudo-état final.

Un état est un écran, une page, qui est représenté par un rectangle avec les coins arrondis.
Les transitions sont représentées par une flèche qui pointe vers un état



D. Diagramme de Séquence Système

Le Diagramme de Séquence Système décrit le comportement du système qui découle d'un Cas d'Utilisation défini précédemment.

Il formalise les interactions entre les acteurs et le système ou les classes, grâce à des messages, avec un point de vue chronologique et spatial, le temps étant représenté à la verticale, et l'espace à l'horizontal.

Il représente généralement deux objets : l'utilisateur du système, et le système.

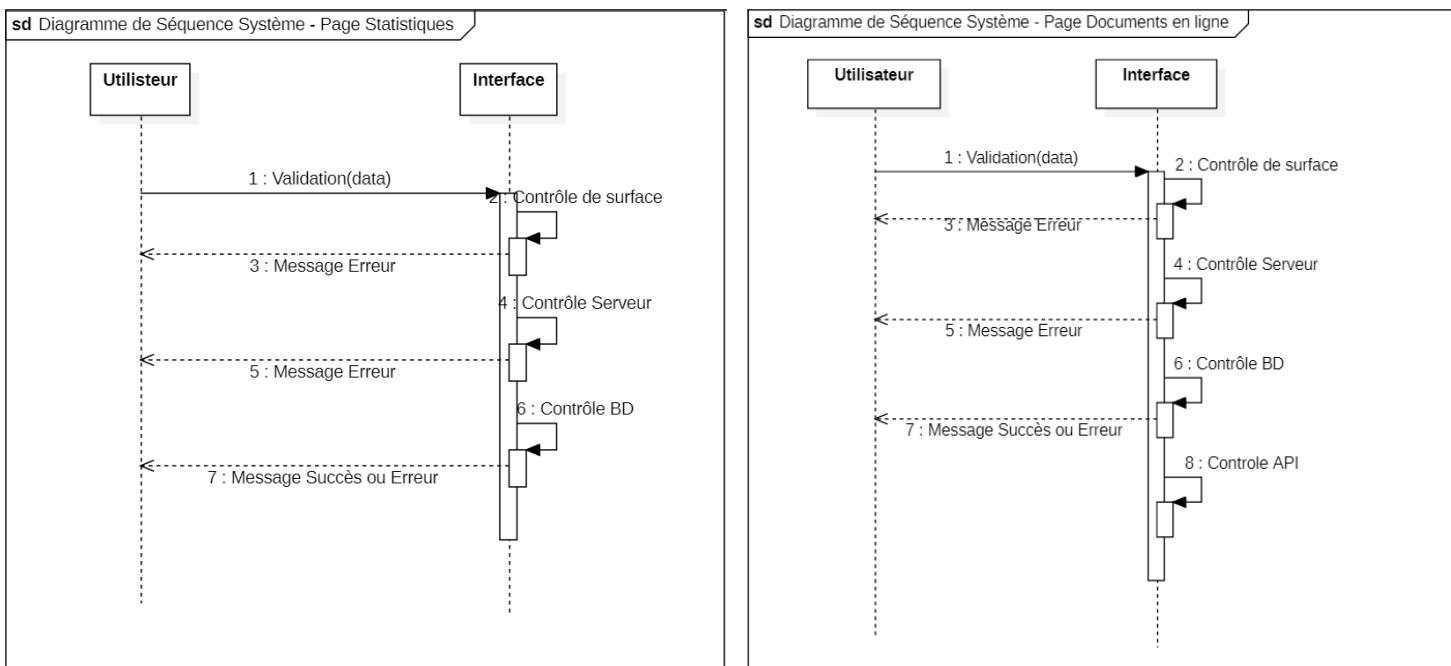
Chaque objet est représenté par un rectangle, sous lequel se dessine une ligne en pointillé, qui descend vers le bas depuis son centre. On appelle cela une ligne de vie.

Les messages sont représentés par des flèches horizontales, pleines ou en pointillé, de l'émetteur vers le destinataire.

Il existe 5 types de messages :

- CALL : flèche fermée et pleine avec un trait continu,
- SEND : flèche ouverte avec un trait continu,
- RETURN : flèche ouverte avec un trait en pointillé,
- CREATE : flèche fermée et pleine avec un trait plein surmonté du mot « create » entre chevrons,
- DESTROY : flèche fermée et pleine avec un trait plein surmonté du mot destroy entre chevrons. La ligne de vie se termine par une croix de Saint-André.

Voici 2 diagrammes en exemple :



IV. La base de données

Une base de données (BD) sert à stocker des informations pour ensuite être capable de les traiter, afin d'en extraire les données.

La BD travaille avec des tables. Chaque table est composée de champs que l'on appelle des colonnes, et chaque table contient de 0 à plusieurs enregistrements qu'on appelle des lignes.

Pour accéder à la BD, nous allons soumettre des requêtes en utilisant le langage **SQL** au **Systèmes de Gestion de Bases de Données Relationnelle (SGBDR)**.

Les SGBDR sont les programmes qui se chargent du stockage de données.

La base de données utilisée pour QRMiam est gérée par **MySQL**, et les tables sont basées sur le moteur **InnoDB** qui prend en charge les clefs étrangères et la gestion des transactions.

Les clés primaires servent à identifier une ligne de manière unique.

Les clés étrangères permettent de gérer des relations entre plusieurs tables, elles garantissent la cohérence des données, et l'intégrité référentielle.

A. Démarche utilisée

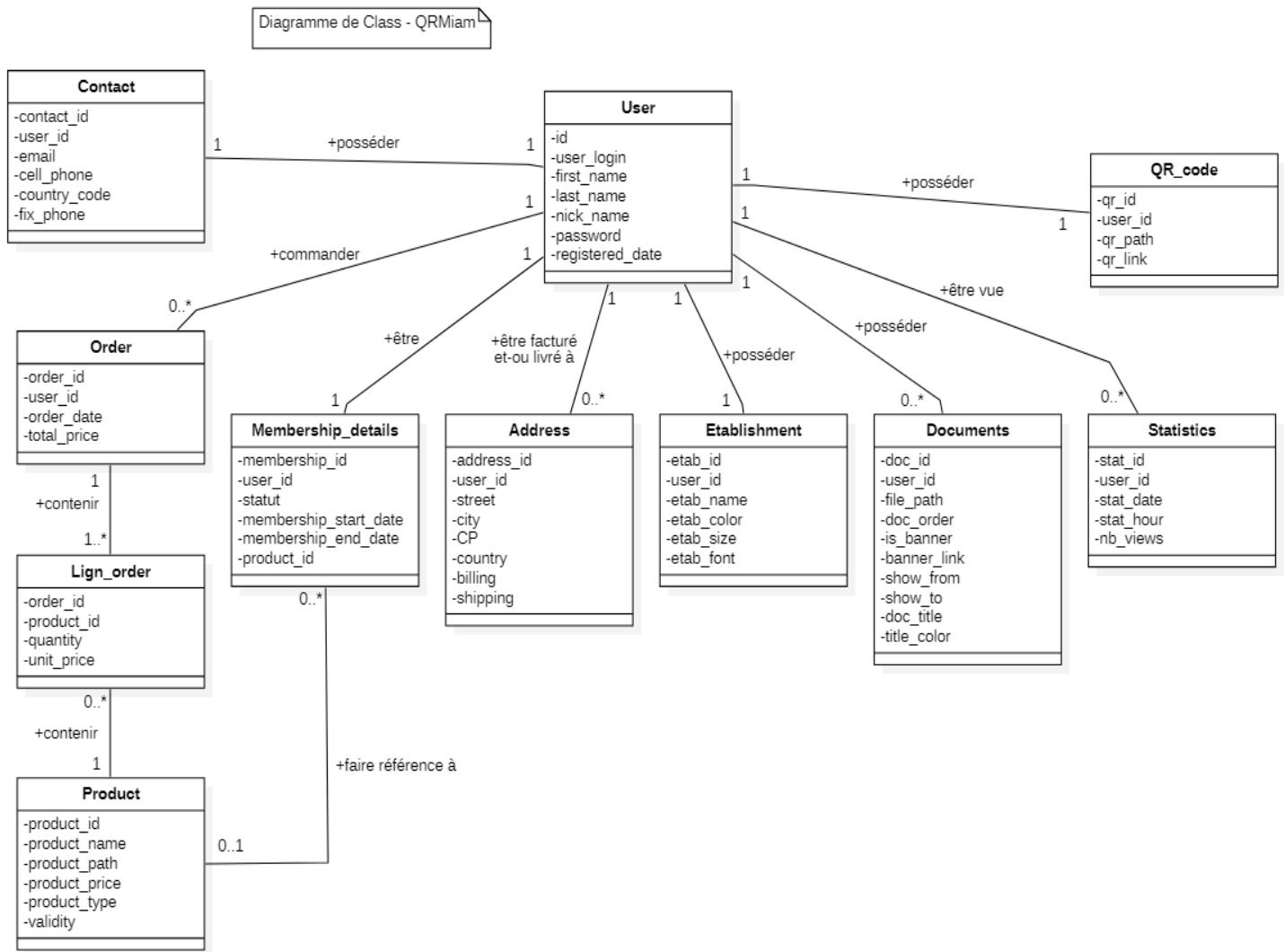
Dans le cadre de mon stage je n'avais pas pour mission de créer une Base de Données (BD) car le site WordPress a sa propre BD. C'est donc afin de compléter ce dossier que j'en ai créé une nouvelle dans le cadre d'un projet futur qui serait de refaire le site entièrement sans utiliser WordPress.

En effet le site devient de plus en plus complexe avec des plugins qui s'additionnent et le ralentissent. C'est donc avec l'objectif de le rendre plus clair, moins lourd, plus rapide et plus simple à maintenir que je leur ai recommandé de le refaire et que j'ai créé la BD qui leur servirait.

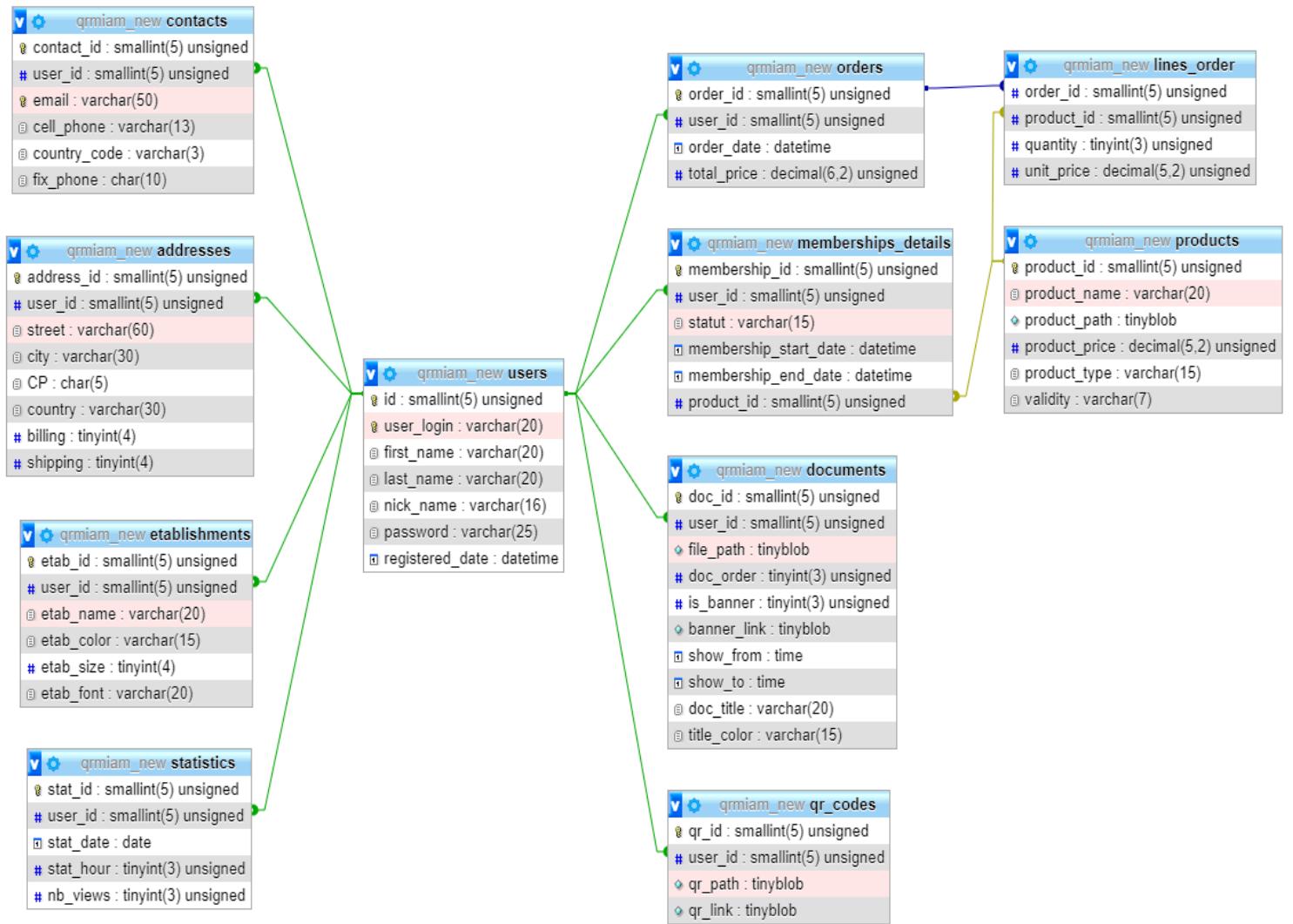
En analysant le site, sa BD existante, ses fonctionnalités et les données utilisées, j'ai identifié les différentes classes nécessaires qui correspondent aux nouvelles tables de la BD. Ensuite j'ai déterminé les données utiles et en ai déduit les attributs de chaque classe qui correspondent, eux, aux noms des colonnes.

Je les ai ensuite schématisées dans un diagramme de classe afin de repérer leurs connexions, ce qui m'a permis ensuite de coder la création de la BD et ses tables.

B. Diagramme de Class (DCL)



C. Schéma de la Base de Données



D. Crédit de la base de données

```
-- Crédit de la Base de Données
```

```
DROP DATABASE IF EXISTS qrmiam_new;
CREATE DATABASE qrmiam_new DEFAULT CHARACTER SET utf8mb4;
```

E. Structure et création des tables

```
-- Création des tables
USE qrmiam_new;
SET FOREIGN_KEY_CHECKS = 0;

-- Create Table "users"
DROP TABLE IF EXISTS users;
CREATE TABLE IF NOT EXISTS users (
    id SMALLINT UNSIGNED AUTO_INCREMENT,
    user_login VARCHAR (20) UNIQUE NOT NULL,
    first_name VARCHAR (20) NOT NULL,
    last_name VARCHAR (20) NOT NULL,
    nick_name VARCHAR (16),
    password VARCHAR (25) NOT NULL,
    registered_date DATETIME NOT NULL DEFAULT NOW(),
    PRIMARY KEY (id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Create Table "contacts"
DROP TABLE IF EXISTS contacts;
CREATE TABLE IF NOT EXISTS contacts (
    contact_id SMALLINT UNSIGNED AUTO_INCREMENT,
    user_id SMALLINT UNSIGNED NOT NULL,
    email VARCHAR (50) UNIQUE NOT NULL,
    cell_phone VARCHAR (13) NOT NULL,
    country_code VARCHAR (3) NOT NULL,
    fix_phone CHAR (10),
    PRIMARY KEY (contact_id),
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Create Table "addresses"
DROP TABLE IF EXISTS addresses;
CREATE TABLE IF NOT EXISTS addresses (
    address_id SMALLINT UNSIGNED AUTO_INCREMENT,
    user_id SMALLINT UNSIGNED NOT NULL,
    street VARCHAR (60) NOT NULL,
    city VARCHAR (30) NOT NULL,
    CP CHAR (5) NOT NULL,
    country VARCHAR (30) NOT NULL,
    billing TINYINT NOT NULL,
    shipping TINYINT NOT NULL,
    PRIMARY KEY (address_id),
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```

-- Create Table "establishments"
DROP TABLE IF EXISTS establishments;
CREATE TABLE IF NOT EXISTS establishments (
    etab_id SMALLINT UNSIGNED AUTO_INCREMENT,
    user_id SMALLINT UNSIGNED NOT NULL,
    etab_name VARCHAR (20) NOT NULL,
    etab_color VARCHAR (15) NOT NULL DEFAULT 'black',
    etab_size TINYINT NOT NULL DEFAULT 30,
    etab_font VARCHAR (20) NOT NULL DEFAULT 'Arial',
    PRIMARY KEY (etab_id),
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Create Table "documents"
DROP TABLE IF EXISTS documents;
CREATE TABLE IF NOT EXISTS documents (
    doc_id SMALLINT UNSIGNED AUTO_INCREMENT,
    user_id SMALLINT UNSIGNED NOT NULL,
    file_path TINYBLOB NOT NULL,
    doc_order TINYINT UNSIGNED NOT NULL,
    is_banner TINYINT UNSIGNED NOT NULL DEFAULT 0,
    banner_link TINYBLOB NOT NULL,
    show_from TIME,
    show_to TIME,
    doc_title VARCHAR(20),
    title_color VARCHAR (15),
    PRIMARY KEY (doc_id),
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Create Table "qr_codes"
DROP TABLE IF EXISTS qr_codes;
CREATE TABLE IF NOT EXISTS qr_codes (
    qr_id SMALLINT UNSIGNED AUTO_INCREMENT,
    user_id SMALLINT UNSIGNED NOT NULL,
    qr_path TINYBLOB NOT NULL,
    qr_link TINYBLOB NOT NULL,
    PRIMARY KEY (qr_id),
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

```

-- Create Table "products"
DROP TABLE IF EXISTS products;
CREATE TABLE IF NOT EXISTS products (
    product_id SMALLINT UNSIGNED AUTO_INCREMENT,
    product_name VARCHAR(20) NOT NULL,
    product_path TINYBLOB NOT NULL,
    product_price DECIMAL (5,2) UNSIGNED NOT NULL,
    product_type VARCHAR (15) NOT NULL,
    validity VARCHAR (7),
    PRIMARY KEY (product_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Create Table "memberships_details"
DROP TABLE IF EXISTS memberships_details;
CREATE TABLE IF NOT EXISTS memberships_details (
    membership_id SMALLINT UNSIGNED AUTO_INCREMENT,
    user_id SMALLINT UNSIGNED NOT NULL,
    statut VARCHAR(15) NOT NULL,
    membership_start_date DATETIME,
    membership_end_date DATETIME,
    product_id SMALLINT UNSIGNED,
    PRIMARY KEY (membership_id),
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (product_id) REFERENCES products(product_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Create Table "orders"
DROP TABLE IF EXISTS orders;
CREATE TABLE IF NOT EXISTS orders (
    order_id SMALLINT UNSIGNED AUTO_INCREMENT,
    user_id SMALLINT UNSIGNED NOT NULL,
    order_date DATETIME NOT NULL DEFAULT NOW(),
    total_price DECIMAL (6,2) UNSIGNED NOT NULL,
    PRIMARY KEY (order_id),
    FOREIGN KEY (user_id) REFERENCES users(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Create Table "lines_order"
DROP TABLE IF EXISTS lines_order;
CREATE TABLE IF NOT EXISTS lines_order (
    order_id SMALLINT UNSIGNED AUTO_INCREMENT,
    product_id SMALLINT UNSIGNED NOT NULL,
    quantity TINYINT UNSIGNED NOT NULL,
    unit_price DECIMAL (5,2) UNSIGNED NOT NULL,
    FOREIGN KEY (order_id) REFERENCES orders(order_id) ON DELETE CASCADE,
    FOREIGN KEY (product_id) REFERENCES products(product_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

```
-- Create Table "statistics"
DROP TABLE IF EXISTS statistics;
CREATE TABLE IF NOT EXISTS statistics (
    stat_id SMALLINT UNSIGNED AUTO_INCREMENT,
    user_id SMALLINT UNSIGNED NOT NULL,
    stat_date DATE NOT NULL DEFAULT NOW(),
    stat_hour TINYINT UNSIGNED NOT NULL,
    nb_views TINYINT UNSIGNED NOT NULL,
    PRIMARY KEY (stat_id),
    FOREIGN KEY (user_id) REFERENCES users(id)
) ENGINE=Innodb DEFAULT CHARSET=utf8mb4;

SET FOREIGN_KEY_CHECKS = 1;

-- AUTRE MÉTHODE --
-- ALTER TABLE statistics
-- ADD
-- FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE;
```

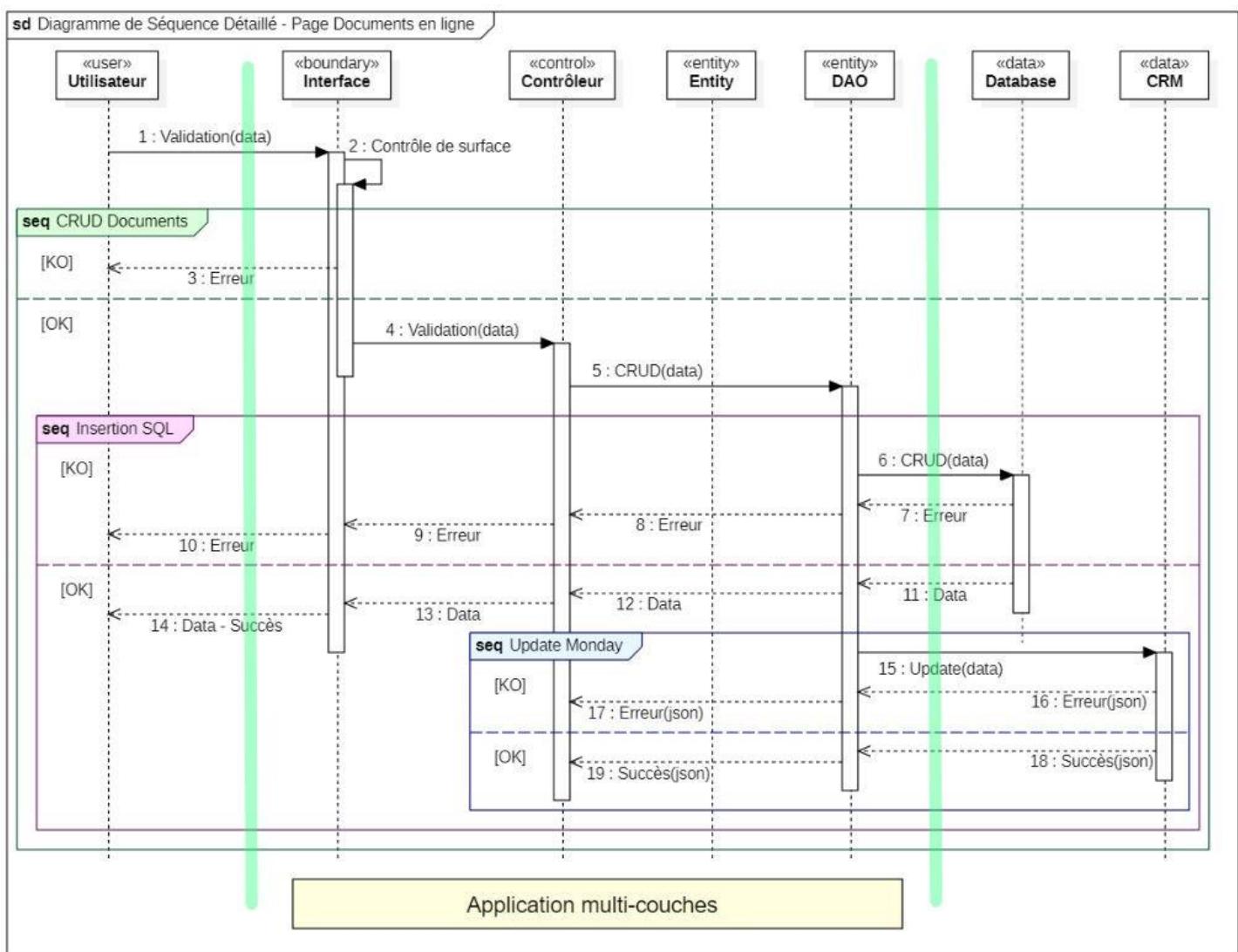
V. Conception du site

A. Diagramme de Séquence Détailé

Le **Diagramme de Séquence** est une solution de modélisation dynamique en langage **UML**, car il se concentre plus précisément sur les lignes de vie, les processus et les objets qui vivent simultanément, et les messages qu'ils échangent entre eux pour exercer une fonction avant la fin de la ligne de vie.

Il permet de montrer les interactions d'objets dans le cadre d'un **scénario** d'un Diagramme des cas d'Utilisation. On représente l'**acteur principal** à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

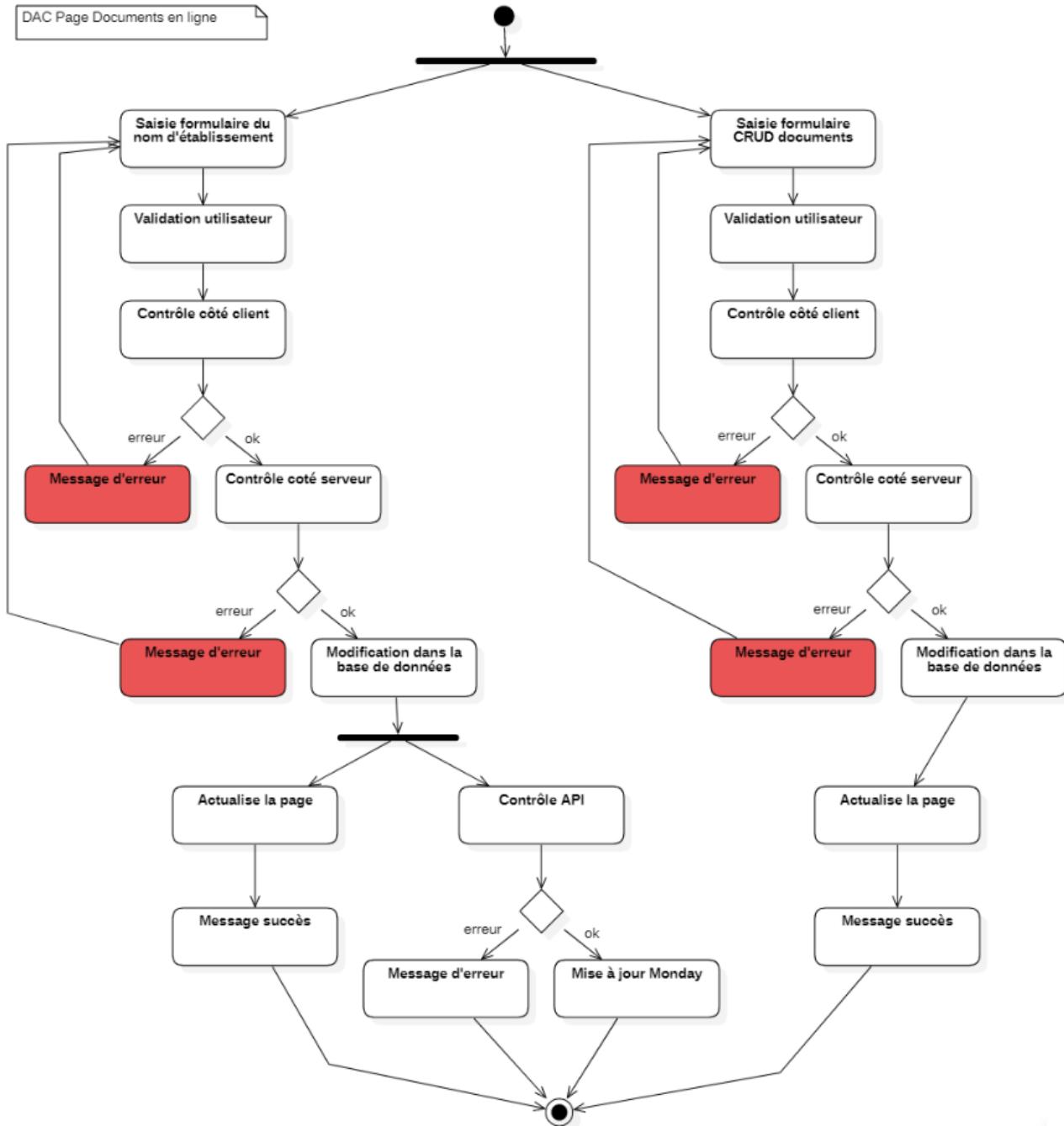
La dimension verticale du diagramme représente le **temps**, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent à l'aide de messages.



B. Diagramme d'Activité

Le diagramme d'activité représente graphiquement le déroulé d'un cas d'utilisation du point de vue de l'utilisateur. Les composants essentiels du DAC sont les actions et les transitions.

Il nous montre des cas alternatifs (un losange représente une condition) et donc des transitions avec des gardes (ok/erreur) en fonction des résultats des tests et des chemins parallèles. Il permet de faire la synthèse des différents scénarios du cas d'utilisation.



VI. Développement du site

A. Technologies utilisées

HTML :

HTML (HyperText Markup Language ou Langage de balisage hypertexte) est comme son nom l'indique un langage de balisage conçu pour représenter les pages web. Il permet également de structurer sémantiquement et logiquement et ainsi de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques.

CSS :

CSS (Cascading Style Sheets ou Feuilles de style en cascade) est un langage informatique qui décrit ma présentation des documents HTML. Il permet de mettre en forme les documents.

PHP :

Langage open-source, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP. PHP est un langage impératif orienté objet. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web. Il permet également le dialogue avec la base de données.

jQuery :

Bibliothèque JavaScript open-source et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. Elle permet de par son utilisation sur le front-end de manipuler le DOM (Document Object Model), de créer des événements, des effets visuels, la manipulation des feuilles CSS.

MySQL :

Un des systèmes les plus populaires de gestion de base de données SQL (Structured Query Language) Open Source. SQL est un langage utilisé pour créer, modifier et extraire des données de la base de données relationnelle, ainsi que pour contrôler l'accès des utilisateurs à la base de données.

GraphQL :

GraphQL est un langage de requêtes pour les API et un moteur d'exécution pour répondre à ces requêtes avec vos données existantes. Il est utilisé pour charger des données d'un serveur vers un client, c'est un moyen d'obtenir des données d'une API dans votre application. Développé par Facebook.

B. Environnement de développement

Visual Studio Code:

Visual Studio Code est un éditeur de code (IDE : Integrated Development Environment) redéfini et optimisé pour la création et le débogage d'applications web et cloud modernes. Visual Studio Code est gratuit et open source. Il peut être utilisé avec une variété de langages de programmation, notamment Java, JavaScript, PHP, Go, Node.js et C++. Il facilite grandement la production de code grâce à la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.

WordPress :

CMS (content management system ou système de gestion de contenu), gratuit, libre et open-source. Ce logiciel écrit en PHP repose sur une base de données MySQL. Les fonctionnalités de WordPress lui permettent de créer et gérer différents types de sites Web : blog, site de vente en ligne, site vitrine

phpMyAdmin :

Outil d'administration gratuit et open source pour MySQL et MariaDB. En tant qu'application web portable écrite principalement en PHP, il est devenu l'un des outils d'administration MySQL les plus populaires, notamment pour les services d'hébergement web.

OVH :

Société française de stockage informatique sur le cloud (cloud computing) qui propose des VPS, des serveurs dédiés et d'autres services web.

Monday.com :

Outil de gestion de projet (CRM : Customer Relationship Management) qui permet aux organisations ou équipes de gérer les tâches, d'exécuter des processus, des projets et des flux de travail dans un espace de travail numérique unique.

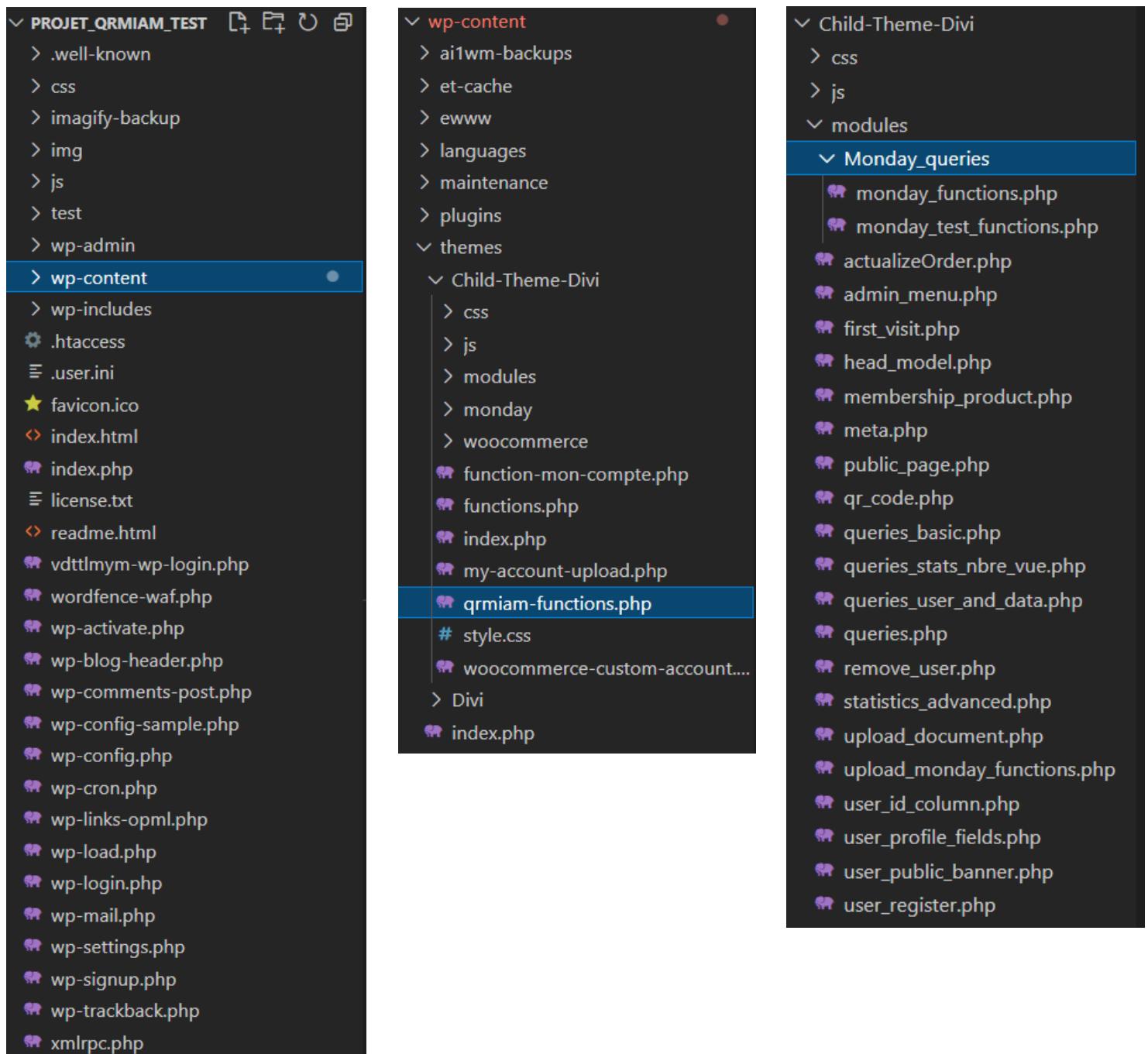
Le site existant a été créé avec WordPress qui est codé en PHP, j'ai donc dû me conformer à leur utilisation pour le développement des différentes fonctionnalités. Pour la gestion de sa base de données WordPress utilise MySQL et l'entreprise utilise OVH pour héberger son site web, ce dernier fonctionnant avec phpMyAdmin pour gérer sa base de données.

Pour coder j'ai utilisé l'IDE Visual Studio Code, c'est celui que j'utilise depuis mes débuts dans le développement et dont je suis très satisfait car il est très ergonomique, facile d'utilisation et me permet de coder dans à la fois en PHP et JavaScript. Il convenait donc très bien pour ce projet.

L'entreprise gère ces clients grâce au CRM Monday et son API utilise le langage GraphQL pour récupérer ou transférer des données depuis ou vers le CRM.

C. Architecture du site

Afin de pouvoir travailler en parallèle avec le développeur WordPress sans effacer par mégarde le travail de l'autre, nous ne travaillions pas directement dans le fichier « function.php » du thème enfant mais dans des templates chacun appelé dans function.php. L'utilisation de Git et des branches n'était pas possible dans ce cas car l'autre développeur ne savait pas l'utiliser. J'utilisais donc qrmiam-functions.php pour développer mes fonctionnalités et appeler les modules.



Ici j'ai décidé de vous présenter la page « Mes documents en ligne » et sa fonctionnalité de modification de l'ordre des documents. Cette fonctionnalité permet de réorganiser à tout moment l'ordre des documents uploadé en cliquant sur le chiffre correspondant à la nouvelle position.

D. Le style responsif

1. Page Web

Tout d'abord voici l'aperçu des pages dans les différents formats d'écran

a) Écran PC

Documents en ligne



Ordre :

Titre

Couleur du titre

Heure début ⏱

Heure fin ⏱

Mise à jour

Effacer



Menu O'PANAME

Ordre :

Titre

Couleur du titre

Heure début ⏱

Heure fin ⏱

Mise à jour

Effacer



Ordre :

Titre

Couleur du titre

Mise à jour

French ▾

b) Écran de tablette

Documents en ligne



Ordre : 1

Titre title

Couleur du titre

Heure début 17:50

Heure fin 00:13

Mise à jour **Effacer**



Menu O'PANAME

Ordre : 2

Titre Menu O'PANAME

Couleur du titre

Heure début ---

Heure fin ---

Mise à jour **Effacer**



Ordre : 3

Titre title

Couleur du titre

Heure début 17:00

Heure fin Français

c) Écran téléphone

Documents en ligne



Ordre : 1

Titre title

Couleur du titre

Heure début 17:50

Heure fin 00:13

Mise à jour

Effacer



Menu O'PANAME

Ordre : 2

Titre Menu O'PANAME

Couleur du titre

Heure début

Heure fin

Mise à jour

Effacer

2. Code HTML

```
<!-- UPDATE DOCUMENT -->
<div class="div-item-form">

<form method="post" accept-charset="utf-8" class="form-item">
    <div class="div-fields-item">
        <!-- Get document id -->
        <input type="hidden" name="document_id" value=<?php echo
            $document['id']; ?>>
        <!-- Get document position -->
        <input type="hidden" name="document_position" value=<?php
            echo $key_actual_doc_position; ?>>

        <!-- Order Document -->
        <div class="div-item-order" >
            <label>Ordre : </label>
            <select class="select-item-order" name="document_ordering"
            >
                <?php
                    orderOption( documents_order: $documents_order, document: $document);
                ?>
            </select>
        </div>

        <?php
        if ($is_premium) { ?>
        <!-- Title Document -->
        <div>
            <label>Titre</label>
            <input type="text" name="title" class="input-text"
                value=<?php echo $document['title']; ?>
                placeholder="title">
        </div>

        <!-- Title Color Document -->
        <div>
            <label>Couleur du titre</label>
            <input type="color" name="title_color" class="input-color"
                value=<?php echo $document['color']; ?>>
        </div>
    
```

```

<!-- Time to display -->
<div>
    <label>Heure début</label>
    <input type="time" name="cn_from" class="input-time"
        value=<?php echo $document['cn_from']; ?>">
</div>
<div>
    <label>Heure fin</label>
    <input type="time" name="cn_to" class="input-time"
        value=<?php echo $document['cn_to']; ?>">
</div>

    <?php } // if ($is_premium) ?>
</div>

<!--
-   BUTTON SUBMIT DOCUMENT  --
-->
<div class="div-btn-item">
    <!-- BUTTON UPDATE -->
    <div class="div-btn-item-update">
        <button type="submit" class="cn_btn gif_load_onClick
et_pb_button" name="btn_update_document">Mise à jour</
button>
    </div>
    <!-- BUTTON DELETE -->
    <div class="div-btn-item-delete">
        <input type="hidden" name="document_id" value=<?php echo
$document['id']; ?>">
        <button type="submit" class="cn_btn gif_load_onClick
et_pb_button" name="btn_delete_document">Effacer</button>
    </div>
</div>
</form>
</div> <!-- END Upload Document -->

```

3. Code CSS

J'ai utilisé les « Média Queries » afin d'affecter un style différent à chaque type d'écran (smartphone, tablette, écran de pc)

Le style initial ne suivait pas le modèle recommandé du « Mobile First » et il partait en premier de l'écran du pc. Le code css est contenu dans un fichier .php et ajouté grâce à une balise HTML afin de paramétrier du style dynamique.

a) Style Écran PC

```
<style type="text/css" media="screen">

#établissement_title {
    font-family: <?php echo $name_style ?>;
    font-size: <?php echo $name_size."px" ?>;
    color: <?php echo $user_établissement_color; ?>
}

.div-marg-10px {
    overflow: hidden;
    margin-bottom: 10px;
}

.update-blocks {
    width: 100%;
    background: #fff;
    padding: 20px;
    margin: 0 auto;
    border-radius: 10px;
}

.update-blocks label{
    font-size: 1.2em;
    color: black;
    display: inline-block;
    min-width: 338px;
    width: 60%;
}

.update-blocks label span{
    display: inline-block;
    width: max-content;
}

.input-text,
.input-number,
.input-color,
.select-style {
    display: inline-block;
    font-size: 1.1em;
    color: black;
    text-align: center;
    border: rgb(95, 95, 95) solid 2px !important;
    border-radius: 4px;
    height: 27px;
}
```

On peut voir ici du code PHP qui permet de fixer des valeurs de manière dynamique en utilisant des variables.

b) Style Tablette

```
/*
 * TABLETTE
 */
@media(min-width: 780px) and (max-width: 980px){

    .div-item-form {
        width: 65%;
        display: flex;
        align-content: center;
        margin-left: 10px;
    }

    .div-fields-item {
        width: 100%;
    }

    .div-btn-item {
        display: flex;
        flex-direction: row;
        justify-content: space-around;
        margin: 10px 0;
    }

    .div-item-img {
        display: table;
        width: 170px;
        min-height: 188px;
    }
}
```

c) Style Smartphone

```
*****  
* SMARTPHONE *  
*****  
  
@media(max-width: 779px){  
    .update-blocks label{  
        display: flex;  
        flex-wrap: wrap;  
        min-width: min-content;  
        width: 100%;  
    }  
    .update-blocks label span{  
        display: inline-block;  
        min-width: min-content;  
        width: auto;  
    }  
    .update-name input,  
    .select-style {  
        min-width: min-content;  
        width: 100%;  
        max-width: 250px;  
    }  
    .update-docs .input-text {  
        min-width: min-content;  
        width: 100%;  
        max-width: 310px;  
    }  
    .update-blocks .input-file {  
        font-size: 1.1em;  
        color: black;  
        display: inline-block;  
        min-width: min-content;  
        width: 100%;  
    }  
    .div-btn-submit{  
        text-align: center;  
    }
```

```
.div-item-form {  
    width: 100%;  
    max-width: 300px;  
    display: flex;  
    flex-direction: column;  
    align-content: center;  
}  
.div-fields-item {  
    width: 100%;  
    min-width: min-content;  
}  
.div-btn-item {  
    display: flex;  
    flex-direction: row;  
    flex-wrap: wrap;  
    justify-content: space-around;  
    width: 100%;  
    margin: 10px 0;  
}  
.div-item-img {  
    display: table;  
    width: 100%;  
    max-width: 300px;  
    min-height: 188px;  
}  
.select-item-order,  
.input-color {  
    width: 26%;  
    min-width: 40px;  
    text-align-last: center;  
    font-weight: 600;  
}  
.input-time {  
    width: 26%;  
    min-width: 75px;  
    text-align-last: center;  
    font-weight: 600;  
}
```

E. Fonctionnalité changement d'ordre

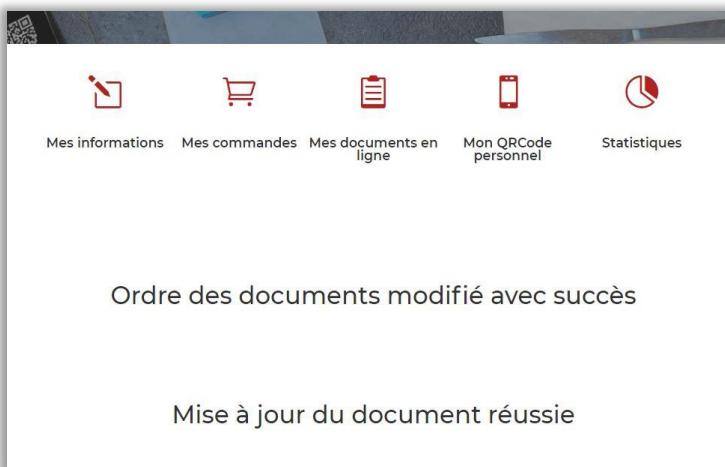
1. Page Web

1- Click sur le champ « select » pour voir les positions

The screenshot shows a web interface for managing document orders. At the top, there's a preview of a cocktail menu. Below it is a form with fields for 'Titre' (Title) set to 'title', 'Couleur du titre' (Color title) set to blue, 'Heure début' (Start time) set to 17:50, and 'Heure fin' (End time) set to 00:13. There are buttons for 'Mise à jour' (Update) and 'Effacer' (Delete). A dropdown menu labeled 'Ordre:' is open, showing options 1 through 10. The option '2' is highlighted with a cursor. To the left of the form is a thumbnail of the document titled 'Menu O'PANAME' with the instruction 'Appuyez pour ouvrir' (Press to open).

This screenshot shows the same web interface after an update. The dropdown menu 'Ordre:' now has '2' at the top and '1' highlighted with a cursor. The other options (3-10) are still visible. The rest of the form and the document thumbnail remain the same.

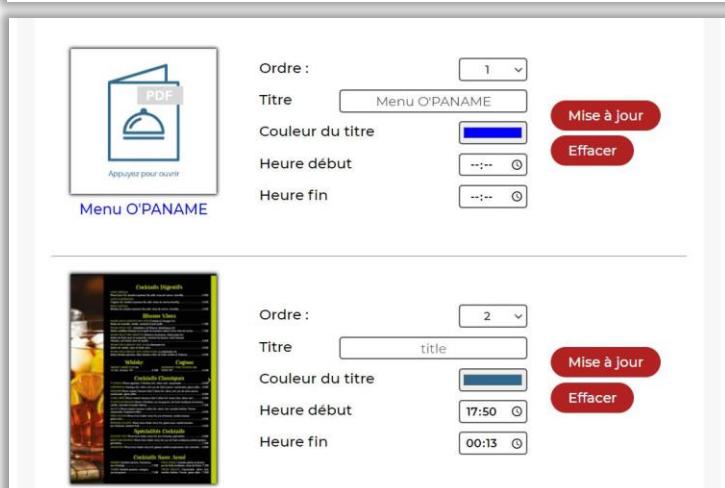
3- la nouvelle position est mise à jour :



(Déclenche la soumission du formulaire)

- la page est actualisée

- un message de confirmation apparaît



- les documents sont réorganisés comme souhaité

2. Code jQuery

Ce code permet simplement la modification de l'ordre des documents lorsqu'on clique sur la nouvelle position. Il active la soumission du formulaire approprié (bannière ou document) au moment du clic sur l'option choisit.

```
// Actualize the documents and banners's order on change of the value (submit the
// form)
$(".select-item-order").change(function () {
    var selectName = $(this).attr('name');
    var closestForm = $(this).closest('form');

    // There are 2 submit button in each form so we must select one to submit the
    // form
    if(selectName == "banner_ordering") {
        closestForm.find('button[name=btn_update_banner]').click();
    } else if (selectName == "document_ordering") {
        closestForm.find('button[name=btn_update_document]').click();
    }
})
```

3. Code PHP

a) Fonctions pour actualiser l'ordre

orderOption() permet d'insérer dans le `<select>` les balises `<options>` qui afficheront la liste des positions des documents (et par conséquent leur nombre) et sélectionnera la position correspondante à chaque item.

```
/*
-   FUNCTION TO ORDER DOCUMENTS AND BANNER
- */

// Displays select's <option> to choose the documents's order
function orderOption($documents_order, $document) {
    foreach ($documents_order as $key => $itemId) {
        $selectedOption = '<option value="'.$key.'" selected>' . ($key+1) . '</option>';
        $notSelectedOption = '<option value="'.$key.'>' . ($key+1) . '</option>';
        // var_dump($documents_order[$key]);
        echo ($itemId == $document['id'])? $selectedOption : $notSelectedOption;
    }
}
```

\$documents_order est un tableau ordinal créé à partir des données des documents. Les documents sont récupérés dans l'ordre établis dans la BD grâce à la colonne « ordering ». C'est dans ce tableau que l'ordre des documents va s'organiser.

```
/*
-   UPDATE DOCUMENT SUBMITED
- */

// Save the id's document list in a table to set the order, will be used to manage the
order
$documents_order = [];
foreach ($list_user_documents as $document){
    array_push(&array: $documents_order, values[0]: $document['id']);
}
```

`get_user_documents_By_Id()` récupère les documents de la personne connectée dans le bon ordre.

```
$list_user_documents = get_user_documents_By_Id( user_id: get_current_user_id() );
```

```
*****
*                                     FUNCTION get_user_documents_By_Id()
*****/
```

```
/*
* Récupère les infos des documents dans la table cn_user_doc en fonction de l'id du client
* et si ce n'est pas une bannière
*
* Used in: upload_document()    /   cn_public_page()    /   user_banner()
* @param [type] $user_id
* @return void
*/
function get_user_documents_By_Id($user_id) {
    global $wpdb;
    $cn_user_doc = $wpdb->prefix . 'cn_user_doc';
    $cn_user_doc_sql = "select * from `".$cn_user_doc."` WHERE `user_id`=$user_id AND `banner` !='yes'
    ORDER BY `ordering` ASC";

    return $cn_user_doc_details = iWhileFetch( sql: $cn_user_doc_sql );
} // get_user_documents_By_Id()
```

```
*****
*                                     FUNCTION iWhileFetch()
*****/
```

```
/*
* Requête générale sur la BD et conversion du résultat json en objet PHP
*
* @param [type] $sql
* @return array type
*/
function iWhileFetch($sql) {
    global $wpdb;
    $record = $wpdb->get_results( query: $sql ); // return array object
    $record = json_decode( json_encode( value: $record ), associative: true );

    return $record;
} // iWhileFetch()
```

actualizeOrder() gère la réorganisation des données afin que chaque document garde bien en mémoire la bonne position. C'est ici qu'on utilise le tableau **\$documents_order** pour gérer l'ordre :

- 1- Je retire le document concerné du tableau, ce qui déplace d'un cran tous les autres documents car les clés ne sont pas préservées
- 2- J'extraie tous les documents à partir de la nouvelle position souhaitée dans un autre tableau **\$sliced**, il y a maintenant 2 tableaux
- 3- J'insère le document concerné qui sera donc à la bonne position
- 4- J'insère le contenu du tableau **\$sliced** à la suite. Et voilà, tous mes documents sont à la bonne position
- 5- J'enregistre dans la BD les nouvelles valeurs de « ordering » qui correspondent aux clés de **\$documents_order**
- 6- Gestion des messages de confirmation ou d'erreur

(Cette fonction gère aussi l'ordre des bannières)

```
// $documents_order must be global with & to be used in orderOption()
function actualizeOrder(&$documents_order, $option_selected, $document_id,
$key_actual_doc_position) {
    array_splice( &array: $documents_order, [offset: $key_actual_doc_position, Length: 1];
    $sliced = array_slice( array: $documents_order, [offset: $option_selected]);
    array_splice( &array: $documents_order, [offset: $option_selected]);
    array_push( &array: $documents_order, [values[0]: $document_id]);
    foreach ($sliced as $item){
        array_push( &array: $documents_order, [values[0]: $item ]);
    }

    global $wpdb;
    $table_user_doce = $wpdb->prefix . 'cn_user_doce';

    foreach ($documents_order as $key => $id) {
        $result_data = array(
            'ordering'=>$key
        );
        $response_ordering = iUpdateArrayMultiConds(
            [table: $table_user_doce,
            postData: $result_data,
            conditions: array('id' => $id)];
        $response_ordering = json_decode( json: $response_ordering);
    }

    // Display update confirmation
    if (isset($_POST['btn_update_document'])) {
        if ($response_ordering->success == 'success') {
            ?> <div class="umsg"><h2>Ordre des documents modifié avec succès</h2></div> <?php
        } else {
            ?> <div class="umsg"><h2>Il y a eu un problème, modification de l'ordre des
            documents non effectuée!</h2></div> <?php
        }
    } elseif (isset($_POST['btn_update_banner'])) {
        if ($response_ordering->success == 'success') {
            ?> <div class="umsg"><h2>Ordre des bannières modifié avec succès</h2></div> <?php
        } else {
            ?> <div class="umsg"><h2>Il y a eu un problème, modification de l'ordre des
            bannières non effectuée!</h2></div> <?php
        }
    }
} // actualizeOrder
```

iUpdateArrayMultiConds() permet de mettre à jour plusieurs colonne d'une table de la BD en fonction de plusieurs conditions, c'est une fonction générale qui fonctionne aussi bien avec des chaînes de caractères, des dates ou des entiers.

```
/*
 *          FUNCTION iUpdateArrayMultiConds()
 */
function iUpdateArrayMultiConds($table, $postData = array(), $conditions = array(), $html_spl='No') {
    global $wpdb;
    $valuesToSet = '';

    // VALUES TO SET
    foreach($postData as $key=>$value) {
        if($html_spl == 'Yes') {
            $value = htmlspecialchars( string: $value);
        }

        if($value == NULL && $value != 0){
            $valuesToSet .= "`$key` = NULL, ";
        } else {
            if (is_numeric( value: $value)){
                $valuesToSet .= "`$key` = $value, ";
            } else {
                $valuesToSet .= "`$key` = '$value', ";
            }
        }
    }

    // Pour enlever la «, » à la fin de «`$k` = NULL, »
    $valuesToSet = substr( string: $valuesToSet, offset: 0, length: strlen( string: $valuesToSet) - 2);
}
```

(Suite de la fonction à la page suivante)

```

// CONDITIONS
$conds = '';
if(count($conditions) == 1) {
    foreach($conditions as $k => $v) {
        $v = htmlspecialchars($v);
        if (is_numeric($v) || strtotime($v)){
            // if (is_int($v) || date_parse($v)){
            $conds .= "$k = '$v'";
        } else {
            $conds .= "$k = '$v'";
        }
    }
} else {
    $i = 0;
    foreach($conditions as $k => $v) {
        $v = htmlspecialchars($v);
        if ($i == 0) {
            if (is_numeric($v) || strtotime($v)){
                $conds .= "$k = '$v'";
            } elseif (is_string($v)){
                $conds .= "$k = '$v'";
            }
        } else {
            if (is_numeric($v) || strtotime($v)){
                $conds .= " AND $k = '$v'";
            } elseif (is_string($v)) {
                $conds .= " AND $k = '$v'";
            }
        }
        $i++;
    }
}

// Requête SQL
$update = "UPDATE `{$table}` SET {$valuesToSet} WHERE {$conds}";
$rs = $wpdb->query($update);

if($wpdb->last_error) {
    $error=$wpdb->last_error;
    $last_query=$wpdb->last_query;
    $success='warning';
    $msg='Something was wrong';
}
else {
    $success='success';
    $msg='updated successfully';
}

$response = array(
    'success' => $success,
    'msg' => $msg,
    'error' => $error,
    'last_query' => $last_query
);

return json_encode($response);
} // iUpdateArrayMultiConds()

```

F. Sécurité : Les Failles XSS et SQL

Le Cross-Site Scripting (XSS) est un type de vulnérabilité de sécurité que l'on trouve généralement dans les applications web. Les attaques XSS permettent aux attaquants d'injecter des scripts côté client dans les pages Web consultées par d'autres utilisateurs. Une vulnérabilité de type XSS peut être utilisée par les attaquants pour contourner les contrôles d'accès. Les XSS exécutés sur des sites web représentaient environ 84 % de toutes les vulnérabilités de sécurité documentées par Symantec jusqu'en 2007¹ !

L'injection SQL permet d'injecter dans la requête SQL en cours un morceau de code MySQL malicieux pouvant modifier la base de données.

Sur les pages où je travaillais durant mon stage il y avait très peu de champs ou l'injection XSS était possible. Pour ceux où il y avait un risque j'ai utilisé la fonction PHP `mysqli_real_escape_string()` qui permet d'éviter à la fois les injection XSS et SQL

```
// Update establishment name
if($_POST['input_etablissement_name']) {

    $new_etablissement_name = $_POST['input_etablissement_name'];

    $con = mysqli_connect( hostname: "localhost:3306", username: "username", password: "mdp", database: "Dbname"
    $new_etablissement_name = mysqli_real_escape_string( mysql: $con, string: $new_etablissement_name);
```

J'ai aussi utilisé aussi `htmlspecialchars()` pour filtrer les caractères spéciaux dans les fonctions créant du SQL.

```
// VALUES TO SET
foreach($postData as $key=>$value) {
    $value = htmlspecialchars( string: $value);
```

FILTRE :

`filter_input()` récupère une variable externe spécifique par son nom et la filtre.

Le filtre `FILTER_SANITIZE_STRING` supprime ou code les caractères indésirables. Ce filtre supprime les données qui sont potentiellement nuisibles pour votre application. Il est utilisé pour éliminer les balises et supprimer ou coder les caractères indésirables. Protège contre les injections XSS

```
// Update establishment name
if($_POST['input_etablissement_name']) {
    // $new_etablissement_name = $_POST['input_etablissement_name'];
    $new_etablissement_name = filter_input( type: INPUT_POST, var_name: "input_etablissement_name",
    filter: FILTER_SANITIZE_STRING);
```

¹ ["Symantec Internet Security Threat Report: Trends for July–December 2007 \(Executive Summary\)"](#)

PASSWORD :

Je n'ai pas eu à sécuriser le mot de passe car WordPress le hash tout seul, mais il faut savoir qu'il est possible de protéger son mot de passe en PHP grâce la fonction ***password_hash()***.

```
$password = filter_input(INPUT_POST, "password", FILTER_DEFAULT);
$hash = password_hash($password, PASSWORD_BCRYPT);
```

REQUÊTE PRÉPARÉE :

Contre les injection SQL on peut aussi utiliser des « requêtes préparées » qui s'utilisent avec les commandes ***prepare*** et ***execute*** avec PHP.

On utilise des ? ensuite j'exécute ma requête en donnant des valeurs au paramètres.

(Extrait de code d'une autre application personnelle utilisant NodeJS)

```
var item = data[i]
// insert les données des lignes 1 par 1 à chaque boucle
var sql = `INSERT INTO ${BDchoisie}.${nomTable} SET ?`;

connexion.query(sql, item, function (err, rows, fields) {
    if (!err) { // Pas d'erreur

    } // if Pas d'erreur

    else { // Erreur !!!
        console.log('\nErreur d\'exécution de la requête !' + err);
    }
}); // query
```

G. Difficultés rencontrées

Tout d'abord j'ai dû analyser, décortiquer, m'approprier un fichier de code vraiment très compliqué à lire.

Les indentations n'étaient pas respectées, les variables et les fonctions n'étaient pas efficacement nommées, et dans un fichier de plus de 2000 lignes ça a vraiment été une épreuve pour moi de comprendre le code.

Il arrivait parfois que certaines variables ou même des fonctions n'était pas utilisées, ce qui rendait d'autant plus difficile de m'y retrouver.

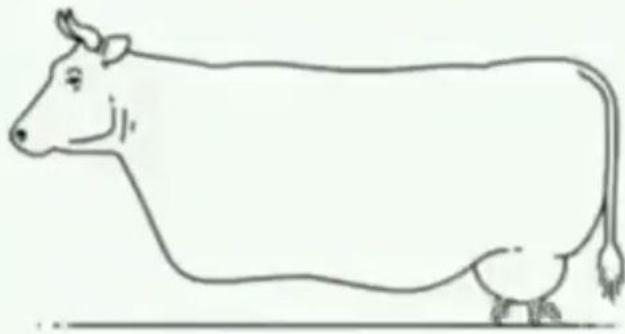
Je suis de nature perfectionniste mais j'ai dû me résigner à ne pas toucher certaines choses de peur de ne pas voir certaines connexions et de rendre le site non opérationnel. Mes 2 mois de stage ne me permettaient pas de refaire le site entièrement et il valait mieux qu'il soit opérationnel à la date voulue, quitte à prendre le temps de refaire bien les choses plus tard.

Pour développer en toute sécurité hors du site en ligne, j'ai voulu cloner le site WordPress mais je n'ai pas du tout réussi à cause d'erreurs lors des tentatives de copies de la base de données de WordPress. Nous avons dû développer directement sur le site en ligne. Les restaurants étant encore fermés et le site non utilisé, cela ne posait pas de problème en cas d'erreurs.

J'ai d'ailleurs eu beaucoup de mal avec WordPress en général, à me repérer dans l'architecture du site et des plugins pour trouver les fichiers que je voulais modifier. J'ai eu du mal à savoir quel **hook** utiliser en fonction des besoins.

OVH contient des sécurités au niveau de la base de données qui m'ont empêché d'obtenir les supers priviléges et d'utiliser les « events » dans MySQL pour la vérification automatique de la validité d'un abonnement après un certain temps. J'ai dû contourner le problème avec l'accord de l'entreprise par un système moins efficace mais temporaire. J'ai découvert, un peu tardivement dans le stage, la possibilité d'utiliser CRON pour programmer des tâches et j'ai manqué de temps pour le mettre en place.

**1st rule of programming:
If it works don't touch it**



VII. GIT

Git est un système de contrôle de version distribué, gratuit et open source, conçu pour traiter tous les projets, des plus petits aux plus grands, avec rapidité et efficacité.

Il permet de sauvegarder des versions différentes de son application tout au long de son développement et aussi de travailler efficacement en équipe en l'utilisant comme archive distante et grâce à son système de branches.

GitHub est un service d'hébergement de dépôts Git, mais il ajoute de nombreuses fonctionnalités qui lui sont propres. Alors que Git est un outil en ligne de commande, GitHub fournit une interface graphique basée sur le Web. Il fournit également un contrôle d'accès et plusieurs fonctions de collaboration, comme un wiki et des outils de gestion des tâches de base pour chaque projet.

Commandes	Actions
Git init	Initialise GIT sur le projet et crée un nouveau dépôt
Git add	Ajoute à l'index les fichiers versionnés et modifiés
Git commit	Stock l'index avec un message qui décrit les changements
Git push	Envoi le commit sur le dépôt du serveur distant (pc->git)
Git pull	Permet de récupérer la dernière version du serveur distant
Git log	Historique complet de toute l'activité sur le dépôt
Git clone	Permet de créer une copie du dépôt local ou de récupérer un projet sur un serveur distant

J'ai utilisé Git pour sauvegarder mon travail et l'ai stocké sur Github pour bénéficier de son interface utilisateur, il suffit de rentrer ces quelques lignes de commandes dans le terminal :

1ère sauvegarde :

Créer un fichier caché « .git »

`git init`

Ajouter les fichiers dans la zone de transit

`git add wp-content\themes\Child-Theme-Divi\`

Valider le dépôt dans le « local repository »

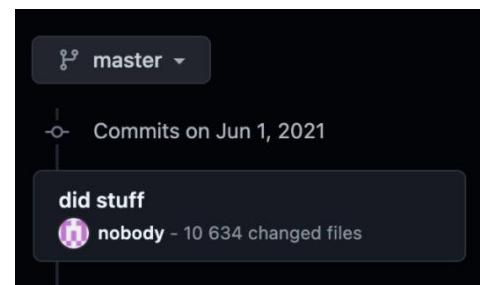
`git commit -m "1ère sauvegarde de mon code"`

Définir un dépôt distant

`git remote add origin https://github.com/myUsername/test.git`

Ajouter les modifications sur le dépôt distant

`git push origin master`



Sauvegarde suivante :

`git add .`

`git commit -m "sauvegarde de mon code"`

`git push origin master`

VIII. Recherche Anglophone

Lors de mon projet de stage j'ai fait énormément de recherches dont le plus grand nombre était en anglais. L'une des plus significatives a été celle me permettant d'apprendre le fonctionnement de l'API de Monday. On trouve la documentation à cette adresse : <https://monday.com/developers/v2#mutations-section>

The screenshot shows the Monday.com Developers API documentation for the 'Change simple column value' mutation. The left sidebar lists navigation items like Introduction, Authentication, Rate Limits, Using GraphQL, Querying monday.com, and Mutating monday.com. The main content area starts with a brief description of what the mutation does: changing the value of a specific column in an item. It then details how different column types require different parameters. A note mentions combining simple and regular values. Below this, it says you can query the board data after the mutation runs. A table shows the 'change_simple_column_value' mutation with its arguments: 'item_id' (Int) and 'column_id' (!String). To the right, a code example in GraphQL shows how to use this mutation with specific values for board_id, item_id, column_id, and status.

Field	Arguments
change_simple_column_value <small>Item</small>	item_id <small>Int</small> The item's unique identifier. column_id <small>!String</small> The column's unique identifier.

```
mutation {  
  change_simple_column_value (board_id: 20178755, item_id: 200819371, column_id: "status", value: "1") {  
    id  
  }  
}
```

Modifier la valeur simple d'une colonne

Portée requise : tableaux:écriture

Les données de chaque élément sont stockées dans des colonnes. La mutation « modifier la valeur simple d'une colonne » vous permet de modifier la valeur d'une colonne dans un élément spécifique (ligne), avec une chaîne de caractères.

Chaque colonne a un certain type, et les différents types de colonnes attendent des paramètres différents pour mettre à jour leurs valeurs.

Lorsque vous envoyez des données à une colonne particulière, utilisez une chaîne de caractères. Pour vérifier comment envoyer les données pour chaque colonne, veuillez consulter la section « modifier des valeurs simples des colonnes ».

Vous pouvez également utiliser les valeurs simples dans les mutations créer_element, créer_sous_element et modifier_multiple_valeurs_d_une_colonne.

Vous pouvez combiner des valeurs simples et régulières dans votre mutation. Ceci est utile lorsque vous souhaitez mettre à jour des colonnes qui ne prennent pas en charge les valeurs simples de colonnes.

Une fois la mutation exécutée, vous pouvez requêter sur les données du tableau, comme indiqué dans la section « Requête des colonnes ».

IX. Conclusion

J'ai commencé à m'initier seul au développement pendant le 1^{er} confinement l'année dernière et je suis vraiment content de voir ma progression depuis le début de ma formation à M2i. Pendant ce stage j'ai pu confirmer mes capacités et je suis très positif pour la suite de mon avenir professionnel. Je me suis aussi rendu compte du chemin qu'il me reste à parcourir pour devenir un « bon développeur », d'autant plus que les routes sont nombreuses, mais je pense sincèrement, et depuis longtemps, avoir les qualités requises pour le devenir.

Ce stage m'a forcé à faire énormément de recherches, de lecture de documentation (tout ce que j'adore) et de débogage (ça j'adore vraiment ! c'est comme des énigmes à résoudre). Résultat, j'ai l'impression de pouvoir réparer n'importe quel bug !

Certaines des choses qui m'attirent dans le développement sont la rigueur et la minutie qu'il faut avoir lorsqu'on code une application, les « bonnes pratiques » à respecter me paraissent couler de source pour une production de qualité.

Un ami développeur m'a dit qu'une des plus grandes difficultés du métier de développeur était la reprise d'un code d'une autre personne, je pense donc que c'était très bénéfique pour moi d'avoir été confronté à un code de cette difficulté très tôt.

J'ai quand même vraiment eu très peur de ne pas avoir les capacités suffisantes pour franchir cette étape obligatoire dès mes premiers jours. Mais je l'ai fait !

6 hours of debugging can save you 5 minutes of reading documentation



J'ai eu beaucoup de chance de tomber sur Neil, Bertrand et Henry comme collaborateur. C'était une équipe bienveillante, intelligente, où la communication était facile et efficace. Neil et Bertrand m'ont vraiment donné une place importante et m'ont fait confiance pour les aider à mener à bien leur projet. Henry a pris le temps de me donner les conseils et les directions à prendre pour me sortir d'impasses ; parfois c'était l'inverse et cela m'a tout autant appris.

J'espère vraiment pouvoir me lancer rapidement dans ce nouveau monde qui, je sens, n'est pas prêt de me lasser.

X. Annexes

A. Statistiques

1. Vues Page Statistiques

a) Date spécifique

Vos cartes et menus ont été vues 1 fois aujourd'hui et 184 fois au total.

Statistiques Avancées

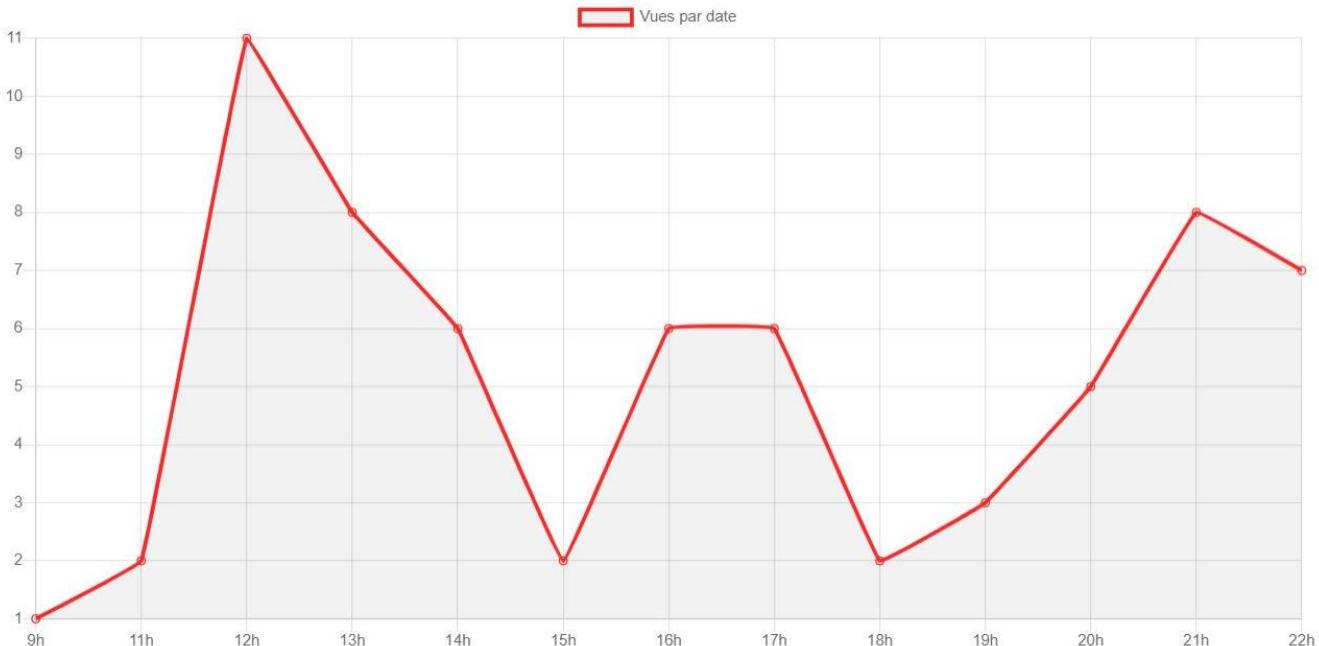
Depuis le début Les 30 derniers jours Mois en cours

Visualiser les statistiques par: Date Spécifique Période

Du

Rechercher un utilisateur:

Statistiques de la.loge.celestins du 04/07/2020 :



b) Période

Vos cartes et menus ont été vues 1 fois aujourd'hui et 184 fois au total.

Statistiques Avancées

Depuis le débutLes 30 derniers joursMois en cours

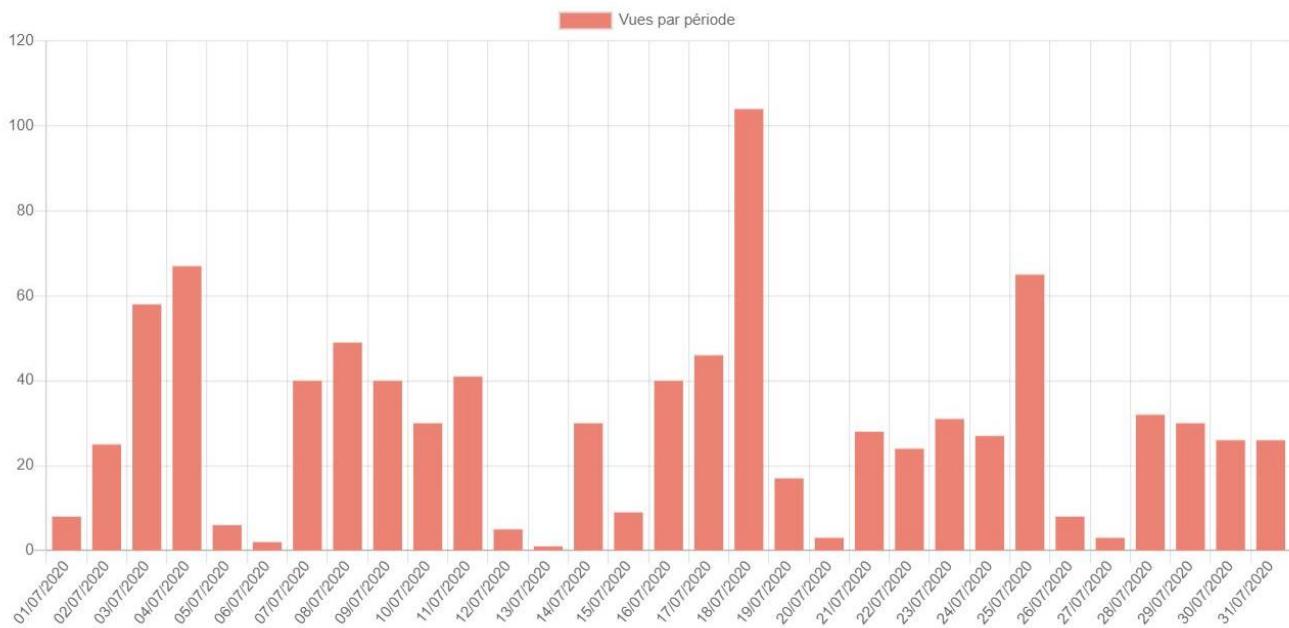
Visualiser les statistiques par: Date Spécifique Période

Du

Au

Rechercher un utilisateur:

Statistiques de la.loge.celestins du 01/07/2020 au 31/07/2020 :



2 choix de vues possible pour la moyenne de vue horaire, géré grâce à du jQuery

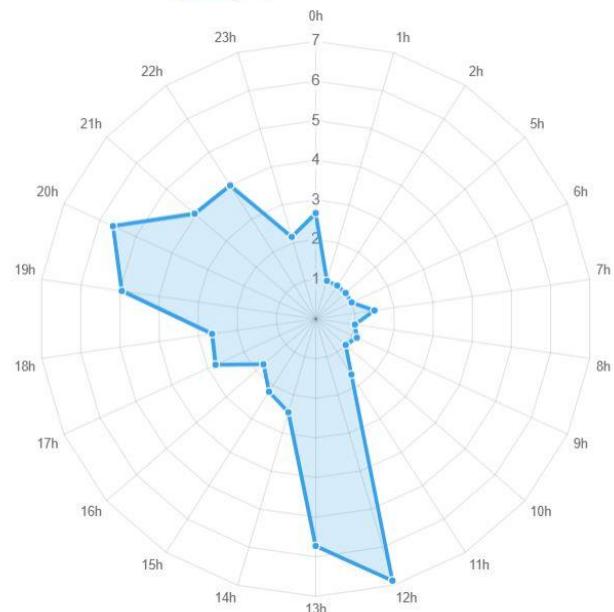
Moyennes des vues par heures sur la période:

Vue Horloge

Vue Courbe



Moyenne des nombres de vue



Moyennes des vues par heures sur la période:

Vue Horloge

Vue Courbe



Vues par date



2. Code

```
/*
 *          FUNCTION show_stats_view() *
 */
function show_stats_view($user_id, $start_date, $end_date, $canvas_type) {

    $stat_nbre_vue = get_stat_nbre_vue_By_user_id_And_Date_hourly( user_id: $user_id, year: $start_date, month: $month, day: $day);
    // echo print_r($stat_nbre_vue);
    if($stat_nbre_vue[0]['nbre_vue'] != 0) {

        for ($i=0; $i < sizeof( value: $stat_nbre_vue); $i++) {

            if(!$stat_nbre_vue[$i]['nbre_vue'] OR !$stat_nbre_vue[$i]['heure']) {
                //echo "0";
            }

            if($end_date) {
                $date = $stat_nbre_vue[$i]['date'];
                list($year, $month, $day) = explode( separator: '-', string: $date);
                $formatedDate = $day . "/" . $month . "/" . $year;
                // $formatedDate = "<span class=\"xChart\>" . $day . "/" . $month . "/" . $year . "</span>";
                $x[$i] = $formatedDate;
                $y[$i] = $stat_nbre_vue[$i]['nbre_vue'];
            } else {
                $x[$i] = $stat_nbre_vue[$i]['heure'].":h";
                $y[$i] = $stat_nbre_vue[$i]['nbre_vue'];
            }
        }
    }
}

?>

<canvas id="chartDate"></canvas>
<script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>

<!--
- show graphic for specific date (by hour)
-->
<?php if (!$end_date) { ?>
<script>
    var ctx = document.getElementById("chartDate");
    var chartDate = new Chart(ctx, {
        type: 'line',
        data: {
            labels: <?php echo json_encode( value: $x); ?>,
            datasets: [{
                label: 'Vues par date',
                data: <?php echo json_encode( value: $y); ?>,
                borderColor: 'rgba(255, 40, 40, 1)',
                lineTension: 0.05,
                backgroundColor: 'rgba(0, 0, 0, 0.05)'
            }]
        });
    </script>
```

```

<!--
- Show graphic for period (by day)  -
-->

<?php } else { ?>
<script>
    var ctxChartDate = document.getElementById("chartDate");
    var chartDate = new Chart(ctxChartDate, {
        // type: 'line',
        type: 'bar',
        data: {
            labels: <?php echo json_encode( value: $x); ?>,
            datasets: [{
                label: 'Vues par période',
                data: <?php echo json_encode( value: $y); ?>,
                lineTension: 0.05,
                backgroundColor: '#EB8273'
            }]
        },
        options: {
            scales: {
                yAxes: [{
                    display: true,
                    ticks: {
                        beginAtZero: true,
                        // min: 0,
                        // precision: 0,
                        // stepSize: 2,
                    }
                }]
            }
        }
    });
</script>
<?php
    show_average_per_hour( user_id: $user_id, start_date: $start_date, end_date: $end_date, canvas_ty
}
} // if($stat_nbre_vue[0]['nbre_vue'] != 0)
else {
    echo "Aucune vue ce jour là.";
}
} // function show_stats_view()

```

```
/*
 *      RADAR CHART      *
 */
var ctxChartAvgHorloge = document.getElementById("chartAvgHorloge");
var chartAvgHorloge = new Chart(ctxChartAvgHorloge, {
    type: 'radar',
    data: {
        labels: <?php echo json_encode( value: $x); ?>,
        datasets: [{
            label: 'Moyenne des nombres de vue',
            data: <?php echo json_encode( value: $y); ?>,
            fill: true,
            backgroundColor: 'rgba(54, 162, 235, 0.2)',
            borderColor: 'rgb(54, 162, 235)',
            pointBackgroundColor: 'rgb(54, 162, 235)',
            pointBorderColor: '#fff',
            pointHoverBackgroundColor: '#fff',
            pointHoverBorderColor: 'rgb(54, 162, 235)'
        }]
    },
    options: {
        elements: {
            line: {
                borderWidth: 3
            }
        },
        scale: {
            ticks: {
                beginAtZero: true,
                // min: 0,
                // precision: 0,
                // stepSize: 2,
            }
        }
    }
});
```

B. Monday.com

1. Vue Application Monday

Commercial	Date	URL	Etablissement	Statut	Lieu	Co
Notice d'utilisation de QR...						
Tarif PDF						
WHAT IS UP MY FRIENDS!						
Hello world!1					66 chemin vert, epone,France	
New name	May 10, 20...	https://qrmiam.fr/pr...	Café, salon de thé	Actif	6 Mail des Tilleuls, Gentilly	
My new complete item	May 10, 20...	https://qrmiam.fr/pr...	Casa de Tom	Actif	6 Mail des Tilleuls, Gentilly	
DevTest	May 22, 03...	https://qrmiam.fr/pr...		Premium	..	
DevTest5	May 25, 03...	https://qrmiam.fr/pr...	Dev café ultra	Premium	6 Mail Des Tilleuls, Bat. A, Apt A...	

2. Code

```
/*
 *          FUNCTION get_users_data_forMonday_By_userLogin() *
 *****/
function get_users_data_forMonday_By_userLogin($user_login) {
    global $wpdb;
    $users = $wpdb->prefix . 'users';
    //$username="victor_test";

    $users_sql = "SELECT ID, user_registered, user_email FROM `{$users}` WHERE
    `user_login`=\"$user_login\"";
    //print_r($etablissement_color=iWhileFetch($users_sql));

    $users_data_raw = iWhileFetch( sql: $users_sql );
    // print_r($user_id_raw);
    return $users_data = $users_data_raw[0];

} // get_users_data_forMonday_By_userLogin()
```

```

function insert_user_registered($user_id) {
    /**
     * GET USER DATA
     */
    $data_user = get_user_by( 'id', $user_id );
    $user_login = $data_user->user_login;

    // url
    $user_login_url = str_replace(" ", "%20", $user_login);
    $url = "https://qrmiam.fr/profile/?username=".$user_login_url;

    // Get data from users table
    $users_table_data = get_users_data_forMonday_By_userLogin($user_login);

    $registered_date = $users_table_data['user_registered'];
    $email = $users_table_data['user_email'];

    // Get data from usermeta table
    $usermeta_table_data = get_user_meta($user_id);

    // Edit Statut
    $membership_product_id = $usermeta_table_data['membership_product_id'][0];
    $list_statut = array('1105' => 2, '236951' => 158, '1107' => 4, '236950' => 8, '238609' => 3, '1000' => 0, '1500' => 1, '2000' => 10, '3000' => 1);
    $statut = $list_statut[$membership_product_id];

    $membership_product_date = $usermeta_table_data['membership_product_date'][0];

    /**
     * DO ACTION
     */
    $responseContent = create_new_user($user_login, $user_id, $registered_date, $url, $statut, $membership_product_date, $email);

    $monday_item_id = $responseContent['data']['create_item']['id'];
    update_user_meta($user_id, 'monday_item_id', $monday_item_id);
}

// END insert_user_registered()

```

```

/*
 *          FUNCTIONS create_new_user()
 *
 * Insert new item (user) in Monday after user registration
 */
function create_new_user($user_login, $user_id, $registered_date, $url, $statut,
$membership_product_date, $email) {

    $token = 'eyJhbGciOiJIUzI1NiJ9.
eyJ0aWQiOjQ4Mzc3MDM0LCJ1aWQiOjEyNjU5Mzg5LCJpYWQiOiIyMDIwLTA2LTA4VDEwOjQwOjA5LjAwMFoiLCJwZXIiOiJtZTp3c
m10ZSIzImFjdGlkIjo1Njc0Mjk0LCJyZ24iOj1c2UxIn0.xNzGee_xepUBxc73pA1m7JaYT_wlBKXszoHa0WsqAyI';
$apiUrl = 'https://api.monday.com/v2';
$headers = ['Content-Type: application/json', 'Authorization: ' . $token];

$query = 'mutation {
    create_item (
        board_id:609995067,
        item_name:"'.$user_login.'"
        column_values:{
            "user_id": "'.$user_id.'",
            "date": "'.$registered_date.'",
            "lien_internet": "'.$url.' '$url.'",
            "statut8": {"index": '.$statut.'},
            "date5": "'.$membership_product_date.'",
            "e_mail": "'.$email.' '.$email.'"
        }
    ) {
        id
    }
}';

$data = @file_get_contents( filename: $apiUrl, use_include_path: false, context: stream_context_create([
    'http' => [
        'method' => 'POST',
        'header' => $headers,
        'content' => json_encode( value: [ 'query' => $query]),
    ]
]));
}

return $responseContent = json_decode( json: $data, associative: true);
// echo json_encode($responseContent);
}

```

C. Jointure SQL

(Fonction de maintenance sur la base de données)

```
/*
 *          FUNCTION update_all_users_monday() *
 */
// NO USED ANY MORE, ONLY FOR DEVELOPMENT
function update_all_users_monday() {
    global $wpdb;
    $usermeta = $wpdb->prefix . 'usermeta';
    $users = $wpdb->prefix . 'users';

    $sql = "SELECT
        u.ID, u.user_login, m.meta_value
        FROM $users u
        INNER JOIN $usermeta m
        ON u.ID = m.user_id
        WHERE m.meta_key = 'membership_product_date'
    ";
    $list_user = iWhileFetch( $sql: $sql);
    // print_r($list_user);

    foreach ($list_user as $user) {
        $user_id = $user['ID'];
        $user_login = $user['user_login'];
        $membership_product_date = $user['meta_value'];

        if ($user_id == 436) {
            $user_monday_raw_data_id = get_item_id_by_name_min ( user_login: $user_login);
            $user_monday_id = $user_monday_raw_data_id["data"]["items_by_column_values"][0]["id"];
            // echo $user_monday_id."<br>";
            update_id_date_monday( user_monday_id: $user_monday_id, user_id: $user_id, membership_product_date:
            update_user_meta( user_id: $user_id, meta_key: 'monday_item_id', meta_value: $user_monday_id);
        }
    }
}
```