

Rapport de projet

Meylan's site

CRÉÉ LE : VENDREDI 9 FEVRIER 2024
CHATTON ADRIEN

Début du module le : 02.02.2022

Fin du Module le :

Version du rapport : v2

Source image : <https://www.futura-sciences.com/tech/actualites/sauvegarde-lignes-code-informatique-ont-desormais-leur-sanctuaire-84330/>

E — Ecole des Métiers
— M Berufsfachschule
— F Fribourg



Table des matières

1 INTRODUCTION 4

2 ANALYSE..... 4

2.1 PRÉSENTATION DU PROJET 4

2.1.1 HOME 4

2.1.2 COLLECTIONS 5

2.1.2.1 « Line »..... 6

2.1.2.1.1 Montres 7

2.1.3 LES LANGUES..... 9

2.2 USES CASE 9

2.3 MAQUETTES 9

2.4 DIAGRAMME ACTIVITÉ 10

2.5 DIAGRAMME DE SÉQUENCES SYSTÈMES 10

2.6 SCHÉMA ER 11

3 CONCEPTION 12

3.1 DIAGRAMMES DE CLASSE 12

3.1.1 CLIENT 12

3.1.2 SERVEUR 13

3.2 SCHÉMA RELATIONNEL 13

3.3 DIAGRAMME SÉQUENCE INTERACTIONS..... 14

3.4 CONCEPTION DES TESTS 14

4 IMPLÉMENTATION..... 18

4.1 DESCENTE DE CODE 18

4.1.1 INTERFACE	19
4.1.2 CHARGEMENT DES INFOS	19
4.1.2.1 Client.....	19
4.1.2.2 Le serveur	21
4.1.3 ENVOI DES MODIFICATIONS	22
4.1.3.1 Client.....	22
4.1.3.2 Serveur.....	23
4.2 PROBLÈMES RENCONTRÉS	27
4.2.1 BASE DE DONNÉE	27
4.2.2 SESSION	28
4.3 TESTS FONCTIONNELS	28
4.4 HÉBERGEMENT.....	32
 5 SYNTHÈSE	 32
 5.1 DIFFÉRENCES ENTRE PLANNING ET RÉALISATION	 32
5.2 CONCLUSION	32

1 Introduction

Pour ce projet, nous allons développer un site web qui fera appel à un backend, les deux seront développés par Orlando et moi.

Ce site web sera une boutique en ligne de vente de montre, plus précisément les montres de l'entreprise appartenant à « Claude Meylan ».

Nous travaillons sur un gros projet et avons donc divisé nos tâches, je vais m'occuper de mettre en place un système ressemblant à un cms et de gestion de stock. En dehors de ce module nous implémenterons aussi un module de paiement.

Donc un utilisateur disposant des droits de modifications du site, aura accès à une nouvelle page. Dans cette dernière nous pourrons modifier la plupart des textes des pages du site, les images des pages ainsi que rajouté des collections de montres, et enfin ajouté des montres. Pour une montre j'entends donc le nom de la montre, les images de cette dernière ainsi que ses caractéristiques.

Le site sera aussi traductible dans deux langues, donc tous les textes seront traduits et stocké une fois en anglais et une fois en français.

2 Analyse

2.1 Présentation du projet

Pour rentrer plus dans les détails après cette introduction, nous allons explorer page par page les possibilités.

2.1.1 Home

Sur la page home, en tant que compte disposant des droits de modifications, nous pourrons modifier l'image du background de l'accueil, ainsi que le texte qui précède la navigation pour la page histoire.



HERITAGE & MODERNITY

In the depths of the Swiss Jura and more precisely in L' Abbaye, a typical village in the luxury watch making Valley (the "Vallée de Joux") where great winters gave birth to great ideas and achievements in time keeping, the family company CLAUDE MEYLAN SA maintains proudly the watch making know-how inherited from their ancestors.

OUR HISTORY



Le texte qui suit le titre ainsi que l'image, seront récupérés directement depuis la page histoire elle-même. Donc si nous modifions le texte et l'image dans Histoire, il sera modifié ici aussi. Le système est pareil pour le paragraphe sur les collections.

Aucune autre partie de cette page n'est modifiable directement !

2.1.2 Collections

Pour la page collection cela se complexifie un peu, pour les modifications simples nous avons l'image de fond ainsi que le titre et texte du premier paragraphe.



THE NEW INTERPRETATION
OF OUR HERITAGE

Entering our world, you will discover a various and extensive collection marked CLAUDE MEYLAN, Vallée de Joux. Each model is unique and dedicated to "true" time keeping. With a classic, yet contemporary style, colour and shape combined in great harmony to ensure

Pour la suite de la page, encore une fois rien ne sera modifiable « directement ». En expliquant mieux, la page collections, sert à recenser toutes les collections de montres. Nous allons simplifier en appelant une collection de montre « line ».

Exemple d'une « line » sur la page collections :



LAC

The Lake of Joux forms integral part of our life. It is there since millenniums and contributes to the beauty of our valley. Its presence releases an infinite serenity, an infinite variation of colours and harmony.

[VIEW THE COLLECTION](#) >

Comme pour la page Home, l'image, le titre et le paragraphe de texte est récupérer depuis la page « line » directement. Bien évidemment dynamiquement pour chaque line.

2.1.2.1 « Line »

Ici nous attaquons le gros du sujet, les modifications possibles sont :

- ❖ Image background
- ❖ Titre
- ❖ Texte qui apparait sur la page collections
- ❖ Texte d'introduction
- ❖ Titre du paragraphe et texte du paragraphe



Since The Prémontrés affixed their prints, L'ABBAYE du Lac de Joux is a typical and characteristic village of the Valley. Throughout its history, L'ABBAYE became and remains one of the cradles of the high watch making industry. It is in its centre; in a house of the heart of the village going back to 1684 that CLAUDE MEYLAN is established.

L'ABBAYE 3263-PB
43MM GOLD PLATED CASE

CHF 4'100

THE NEW INTERPRETATION
OF OLD TALENTS

Since The Prémontrés affixed their prints, L'ABBAYE du Lac de Joux is a typical and characteristic village of the Valley. Throughout its history, L'ABBAYE became and remains one of the cradles of the high watch making industry. It is in its centre; in a house of the heart of the village going back to 1684 that CLAUDE MEYLAN is established.

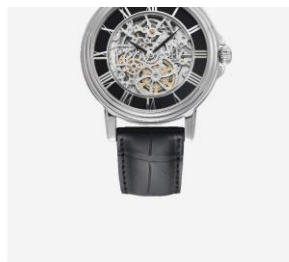
De plus, l'utilisateur avec les droits de modifications aura la possibilité d'ajouter une « line ».

Le reste de la page affichera les montres qui sont contenues dans cette collection, toujours le même principe, nous allons récupérer les informations des montres pour les afficher :



L'ABBAYE 3263-B
43MM STAINLESS STEEL

CHF 4'100



L'ABBAYE 3263
43MM STAINLESS STEEL

CHF 4'100



L'ABBAYE 3263-PM
43MM GOLD PLATED CASE

CHF 4'100



2.1.2.1.1 Montres

Pour les montres, nous aurons la possibilité de modifier :

- ❖ L'image de la montre
- ❖ Son nom
- ❖ Taille
- ❖ Epaisseur
- ❖ Mouvement
- ❖ Descriptif
- ❖ Prix
- ❖ Image shadow

❖ image variante



TORTUE RAINBOW

6047-NP

TAILLE
42 MM

ÉPAISSEUR
21 MM

MOUVEMENT
165 CM

INITIALES*

MD ☒

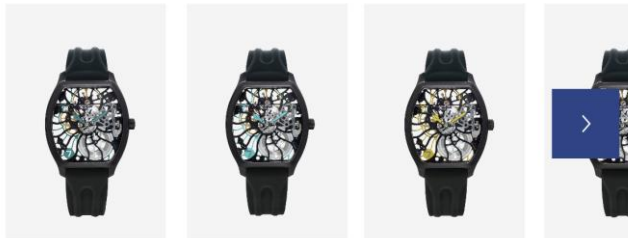
Cette montre en or gris 750/1000 marie la pureté de la nacre blanche à l'éclat des diamants. La lunette sertie de 64 diamants taille rond et sa boîte au diamètre de 30 millimètres épousent à la perfection le poignet. Selon le souhait de son propriétaire, la montre est personnalisable par une gravure sur le fond plein.

4'850 CHF

BUY THIS WATCH >

Une des dernières particularités sont les variances, les variantes sont des montres semblables qui seront affichés ici :

VARIANTES



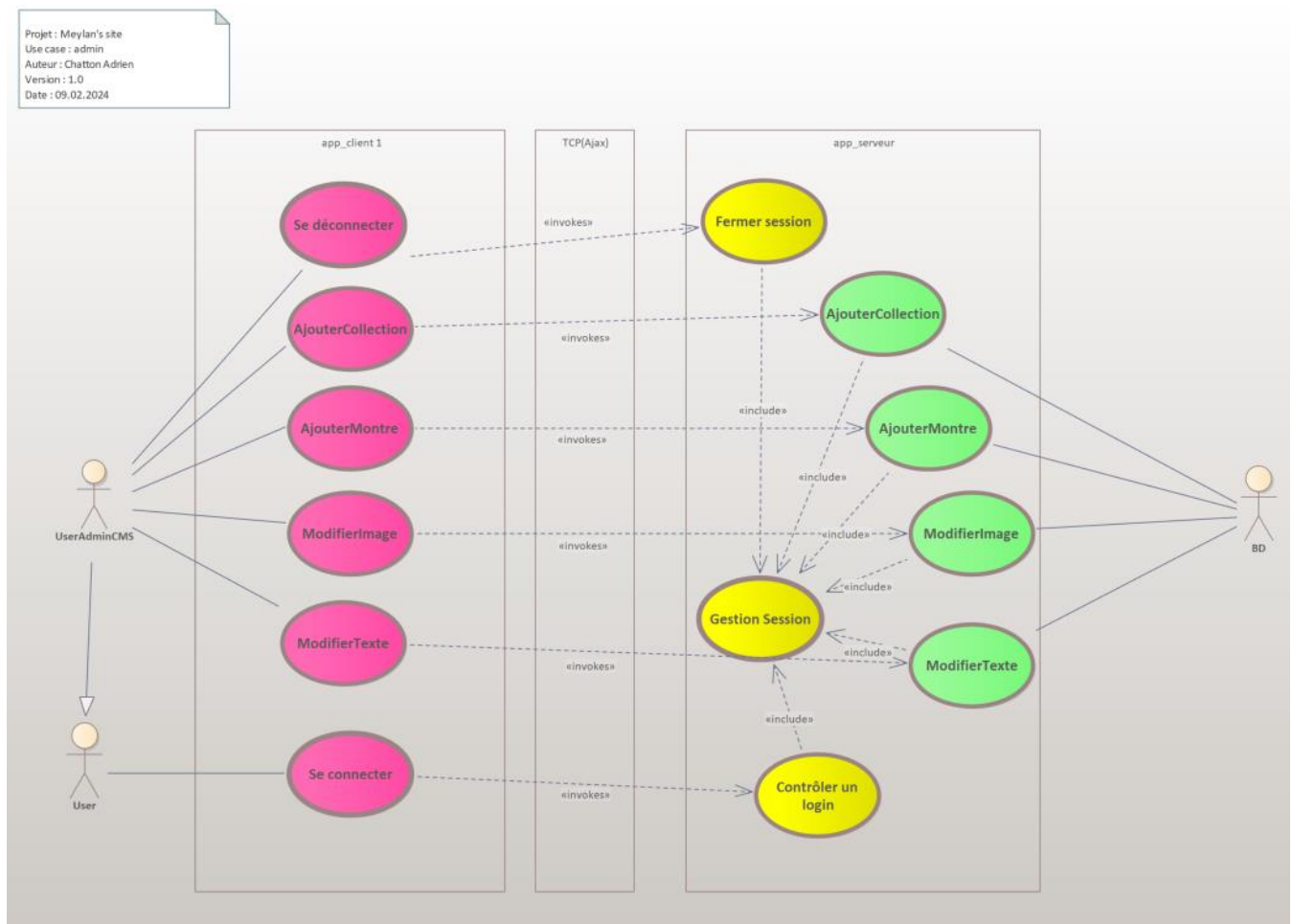
Vu que les variantes sont propres à chaque montre, nous avons un champ variante dans notre base de données qui contient le numéro de la variante.

Et pour finir la possibilité d'ajouté des montres.

2.1.3 Les langues

Pour tous les textes, les modifications sont faisable en deux langues, cela permet un site complètement traductible.

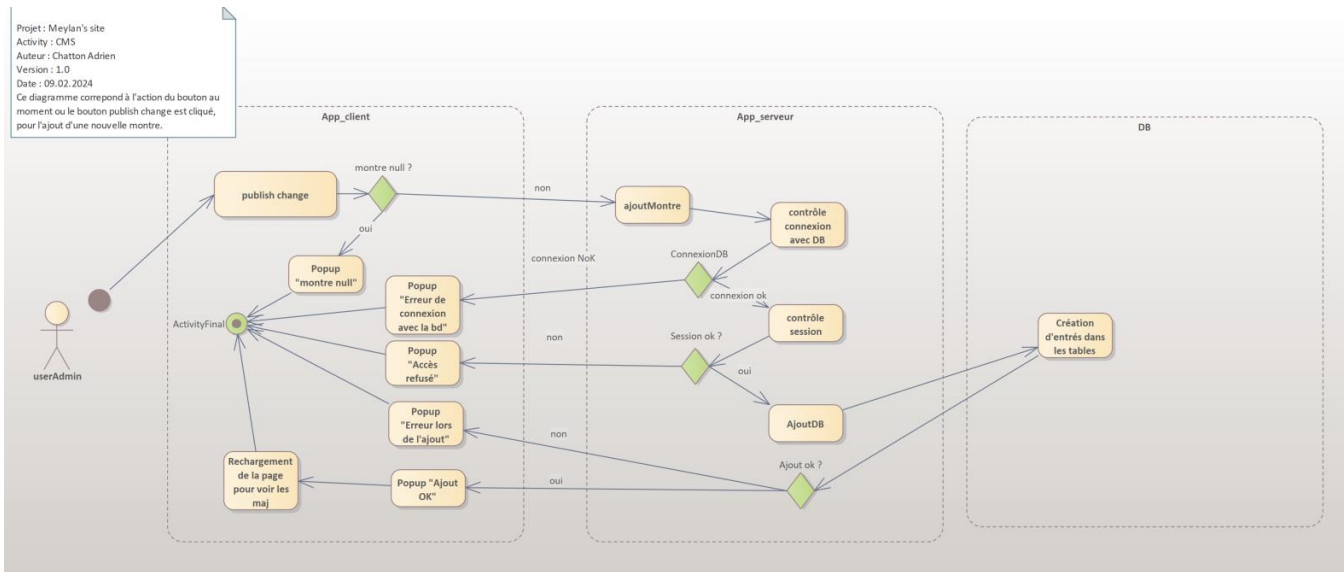
2.2 Uses Case



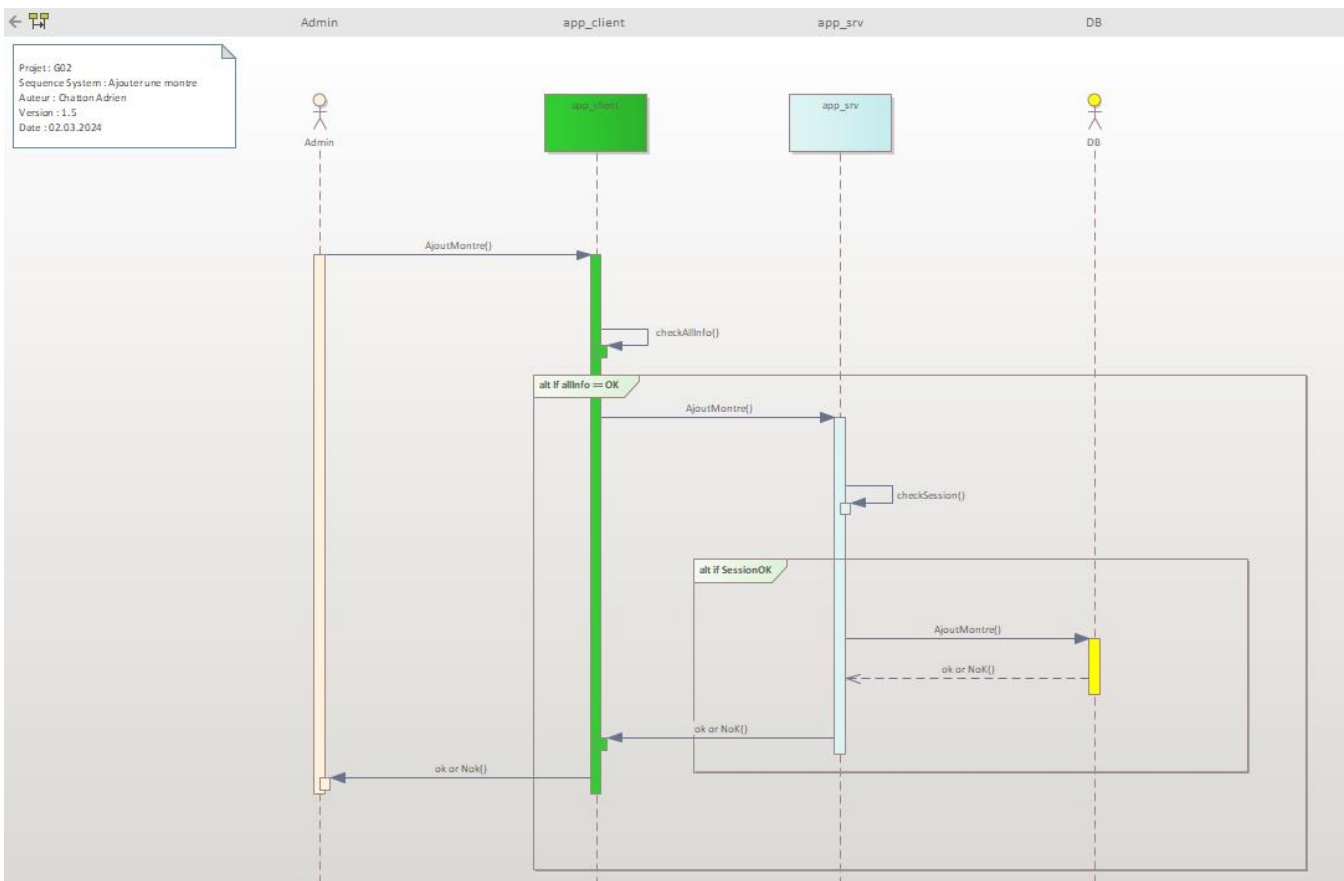
2.3 Maquettes

<https://www.figma.com/proto/ri8lQWvquoA3OBpcsI386T/Meylan?page-id=116%3A6&type=design&node-id=379-136&viewport=-1873%2C288%2C0.33&t=K1fdEb540ipOXZJu-1&scaling=scale-down&starting-point-node-id=372%3A476>

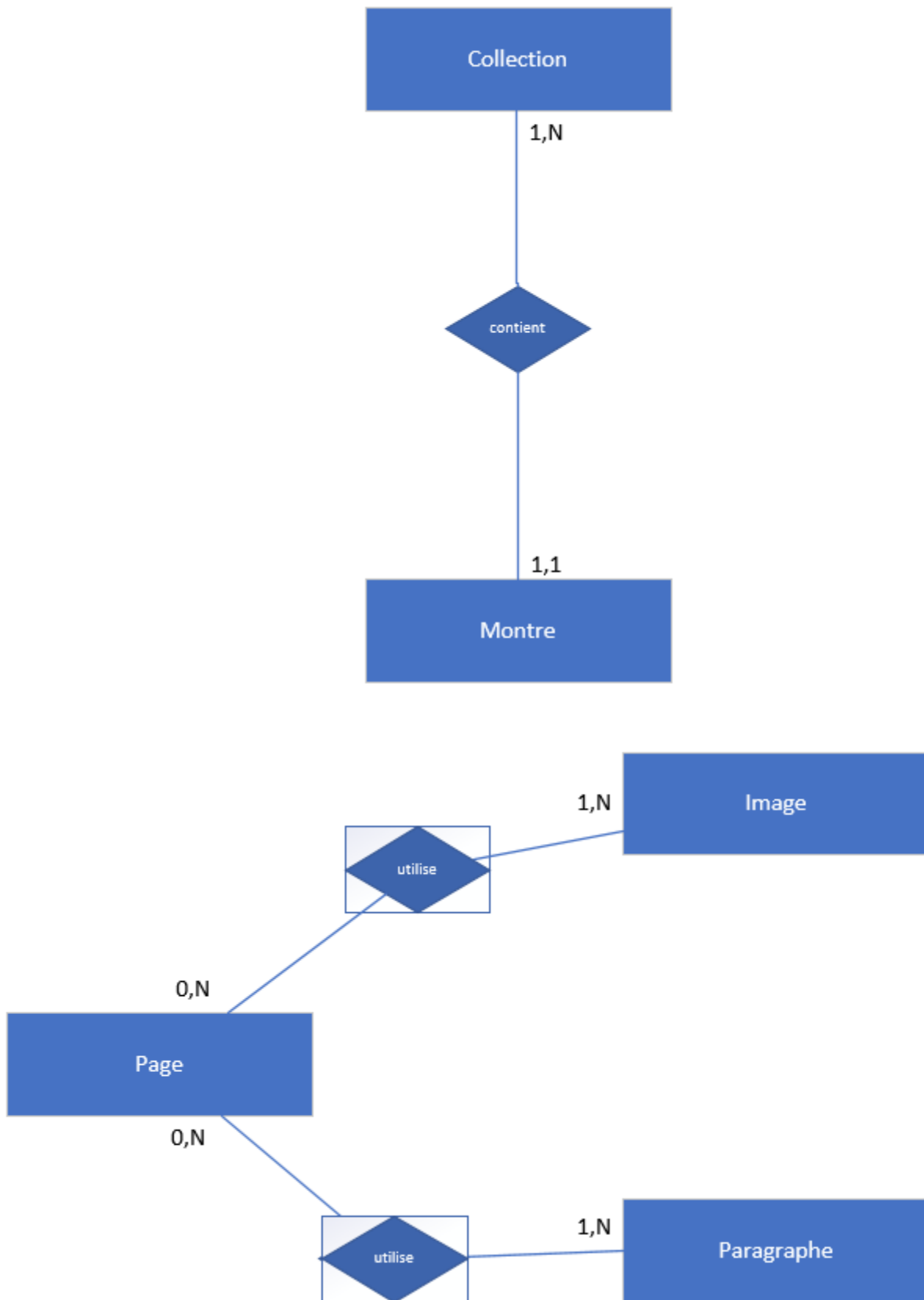
2.4 Diagramme activité



2.5 Diagramme de séquences systèmes



2.6 Schéma ER

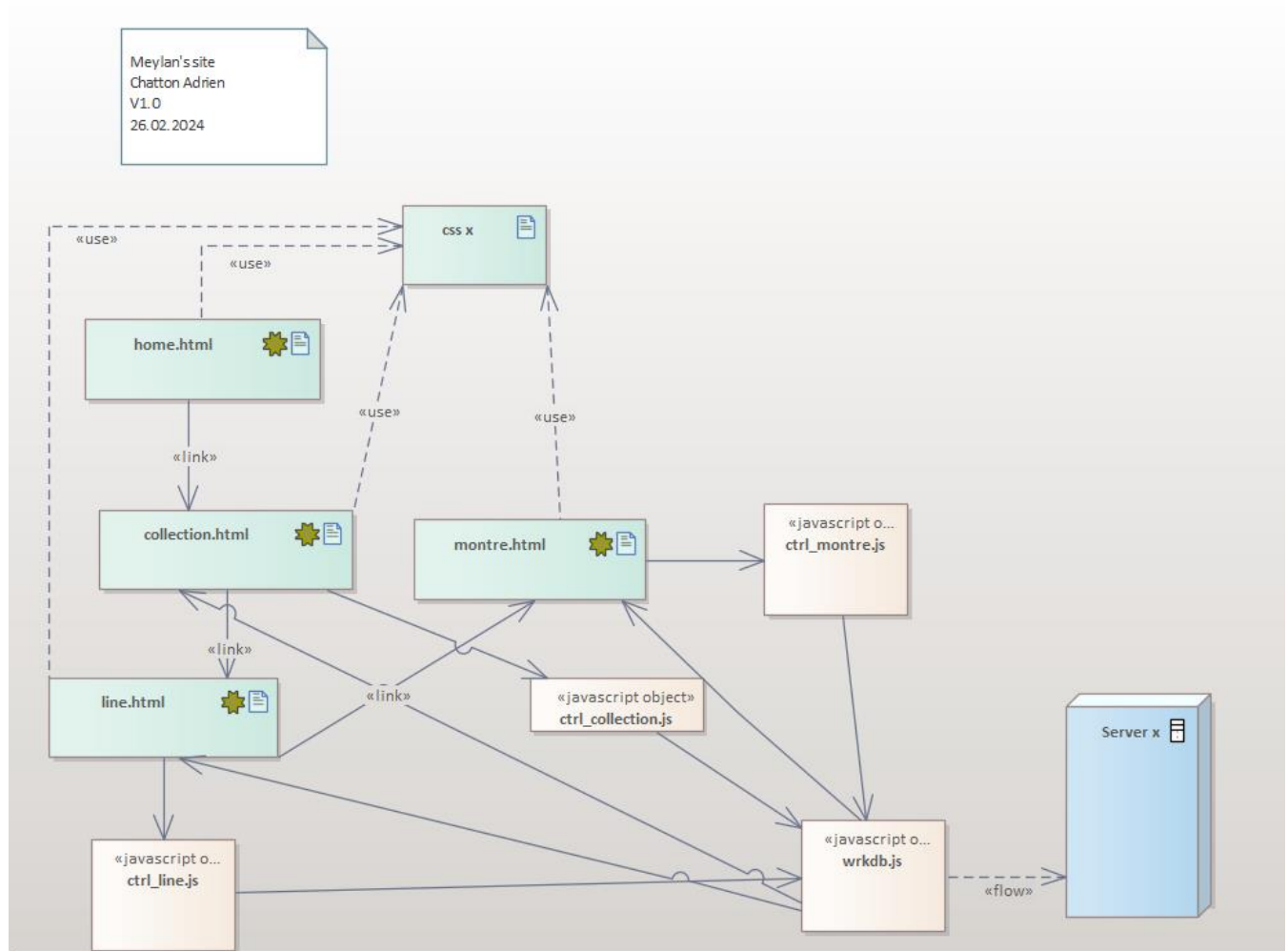


3 Conception

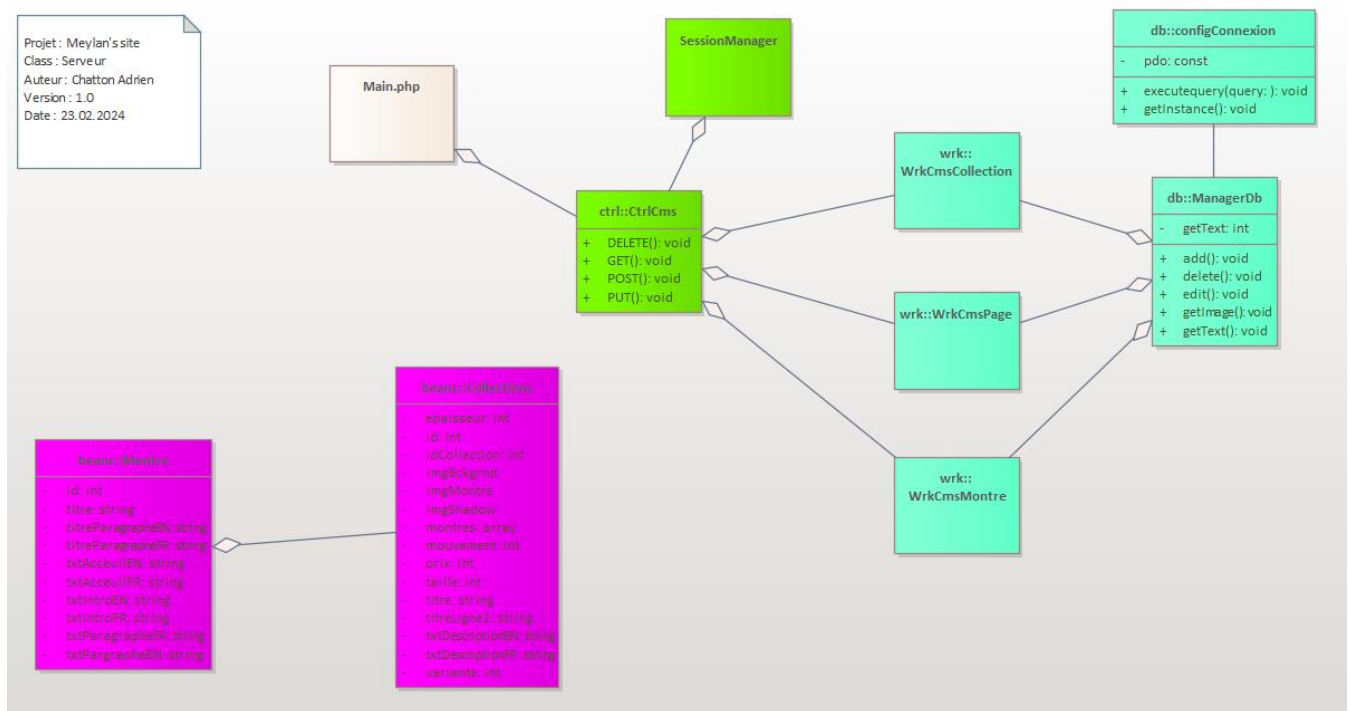
3.1 Diagrammes de classe

3.1.1 Client

Pour le diagramme d'application client, j'ai simplifié. Concrètement nous utiliserons vue.js, un framework. Ce qui permettra de gérer les routes ainsi que de créer des composant qui sont utilisées dans les pages.

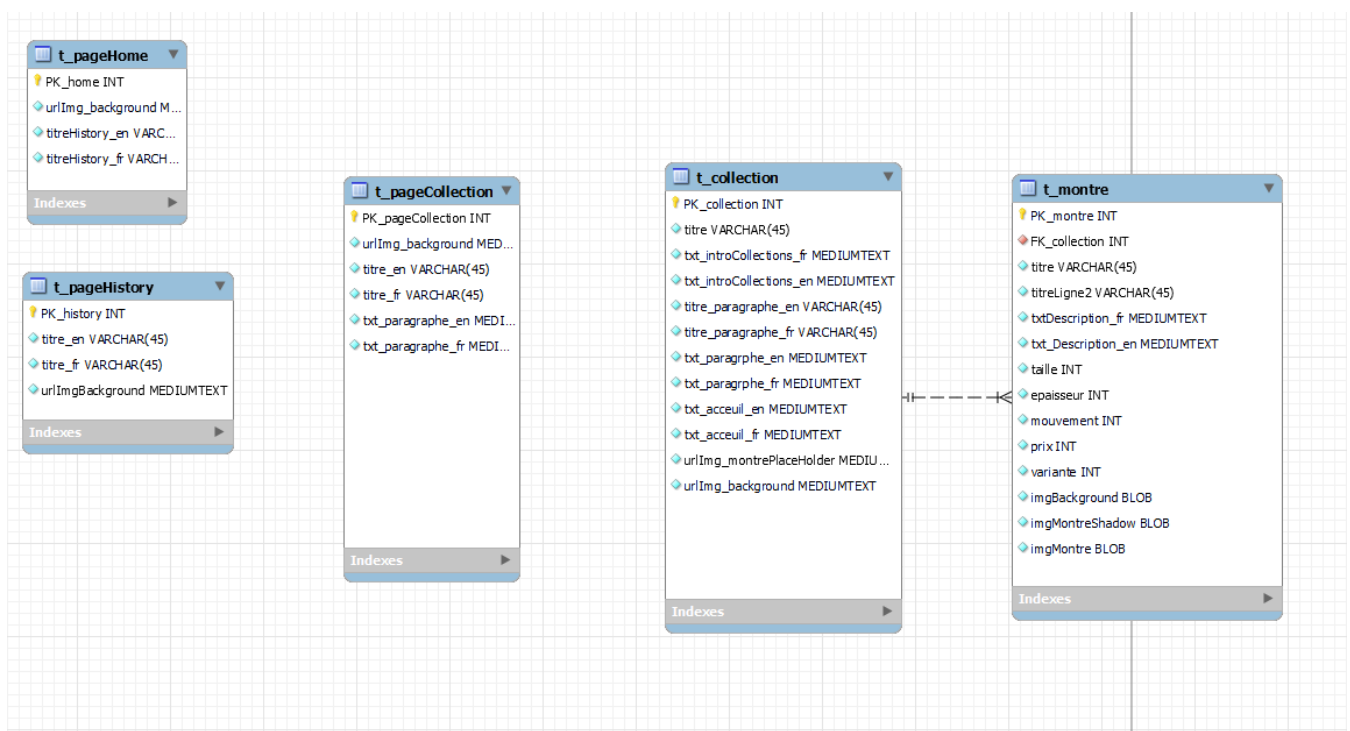


3.1.2 Serveur



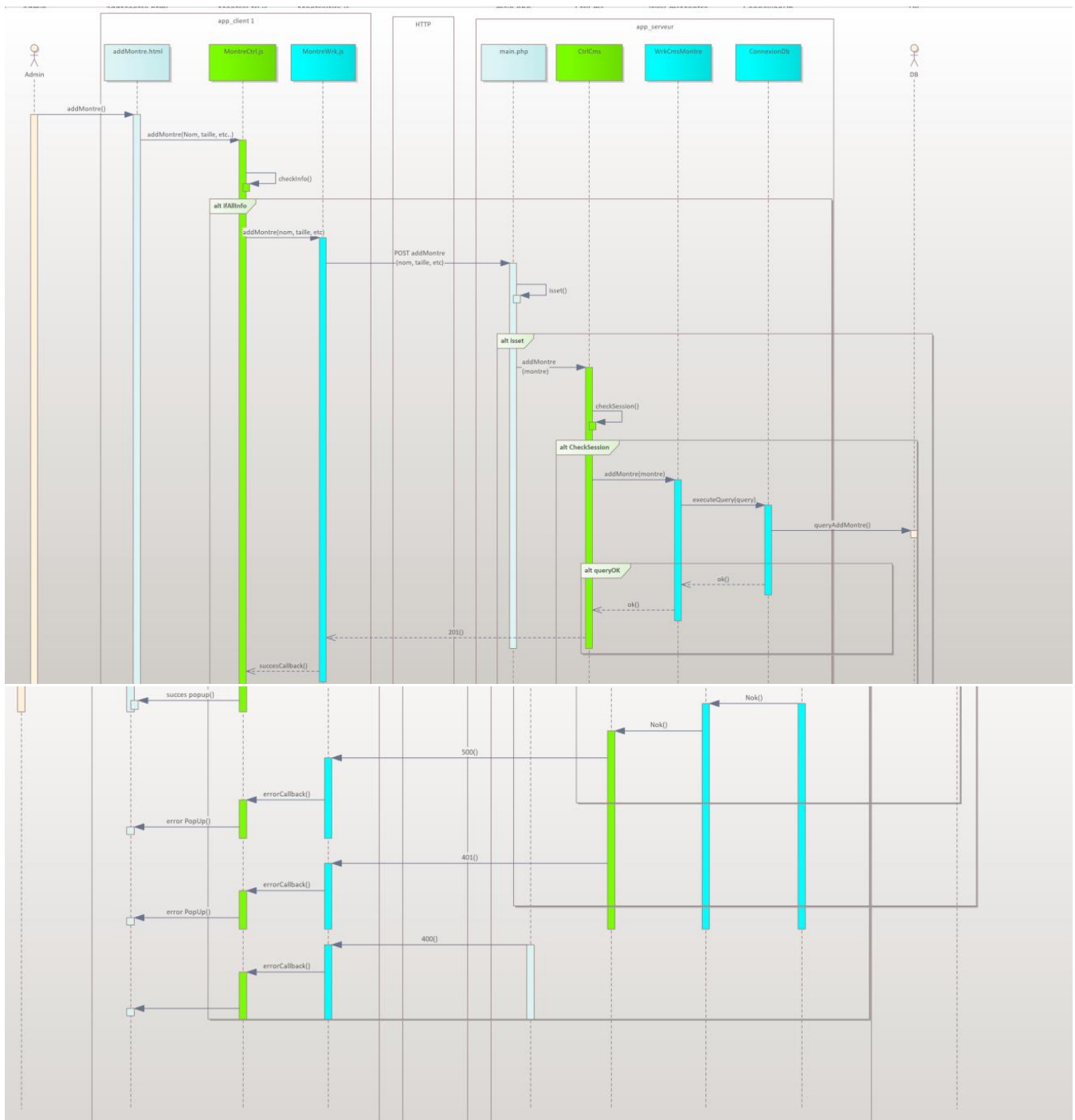
3.2 Schéma relationnel

Ici je n'utilise pas le fonctionnement des pages mis en place dans le diagramme ER, cela simplifie la gestion pour un unique site web et le temps a disposition ne me permet pas d'aller plus loin sans dépasser les délais de retour.



3.3 Diagramme séquence interactions

Désolé, impossible de plus dézoomer sur ce schéma.



3.4 Conception des tests

N° du test

Description

Résultat attendu

1	Chargement des informations dans la page (texte et image)	Les informations affichées sont celles contenues dans la bd
	Résultat obtenu	Conclusion

N° du test	Description	Résultat attendu
2	Changement de langue	Les textes se modifie, et la langue est gardé pour toutes les pages
	Résultat obtenu	Conclusion

N° du test	Description	Résultat attendu
3	Modification d'un texte ou d'une image	Le texte et l'image sont remplacé par les nouveaux dans la db
	Résultat obtenu	Conclusion

--	--	--

N° du test	Description	Résultat attendu
4	Ajout d'une collection	Création d'une entre collection dans la db
	Résultat obtenu	Conclusion

N° du test	Description	Résultat attendu
5	Ajout d'une montre	La montre est ajoutée dans la db
	Résultat obtenu	Conclusion

N° du test	Description	Résultat attendu
6	Affichage des variantes de la montre sur la page montre	Nous avons toutes les montres d'une collection avec la même id de variante sur la page
	Résultat obtenu	Conclusion

N° du test	Description	Résultat attendu
7	Suppression d'une collection	La montre ne se trouve plus dans la db
	Résultat obtenu	Conclusion

N° du test	Description	Résultat attendu
8	Refus d'ajout, suppression et modification si pas connecté	Le serveur répond avec un code 401

	Résultat obtenu	Conclusion

4 Implémentation

4.1 Descente de code

Ici la descente de code pour la modification de la page contact :

4.1.1 Interface

localhost:5173/adminCContact

Home

Histoire

Contact

Collections

Montres

Press

Contact

EN-TÊTE

Image

Choisir un fichier Aucun fichier choisi

ADRESSE

Titre Adresse

ADRESS EXEMPLE

Rue & NO

CESTLARUE

Commune et num postal

test

Pays

pays test

Map

Choisir un fichier Aucun fichier choisi

NEWSLETTER

Image

Choisir un fichier Aucun fichier choisi

Enregistrer

Nous voyons donc notre interface, qui charge avec les infos actuelles de la page.

4.1.2 Chargement des infos

4.1.2.1 Client

Ici j'utilise une fonctionnalité de vue qui permet d'effectuer des actions au moments où la page est chargée, j'utilise également « data » qui me permet d'afficher des variables dans le HTML directement, comme ici par exemple où je récupère ma variable adresse pour préremplir le formulaire avec :

```
<label for="adresse-title">Titre Adresse</label>  
<input class="ml-6" type="text" name="address" v-model="address" />
```

```
<script>
import ServiceHttp from "../../services/ServiceHttp";

export default {
  data() {
    return {
      address: "",
      rue: "",
      commune: "",
      pays: "",
      urlImg_entete: null,
      url_map: null,
      urlImg_news: null,
      serviceHttp: new ServiceHttp(),
    };
  },
  mounted() {
    // Charger les collections au moment du montage du composant
    this.serviceHttp.chargerContact(
      this.succesChargerContact.bind(this),
      this.errorChargerContact.bind(this)
    );
  },
}
```

Dans mon succès callback je vais donc set les data avec celles de la réponse de mon serveur :

```
methods: {
  succesChargerContact(response) {
    this.address = response.address;
    this.rue = response.rue;
    this.commune = response.commune;
    this.pays = response.pays;
  },
  errorChargerContact() {
    alert("Erreur de chargement !");
  },
}
```

Ensuite dans mon service http :

```
chargerContact(successCallback, errorCallback) {
  $.ajax({
    type: "GET",
    contentType: "application/x-www-form-urlencoded",
    url: this.URL,
    data: {
      action: "contact",
    },
  },
```



```
    dataType: "json",
    success: successCallback,
    error: errorCallback,
  });
}
```

4.1.2.2 Le serveur

Ici la réception de la requête

```
if ($_SERVER['REQUEST_METHOD'] == 'GET' && isset($_GET['action'])) {

    switch ($_GET['action']) {
        //Retourne les informations de la page contact :
        case 'contact':
            $ctrlCms->getContact();
            break;
    }
}
```

Ensuite dans mon ctrl :

```
public function getContact()
{
    $jsonContact = $this->wrkCmsPage->getContact();
    if ($jsonContact) {
        http_response_code(200);
        echo $jsonContact;
    } else {
        http_response_code(500);
        echo json_encode('Erreur de chargement de contact');
    }
}
```

Dans mon worker j'utilise un beans contact qui me permet ensuite d'être sérialiser en JSON :

```
public function getContact()
{
    $contact = $this->connexion->selectQuery('SELECT * FROM `t_contact` WHERE 1',
    []);
    if ($contact) {
        extract($contact[0]);
        $contact = new Contact(
            $address,
            $rue,
            $commune,
            $pays
        );
        $contact->setUrlImgEntete($urlImg_entete);
        $contact->setUrlImgNews($urlImg_news);
        $contact->setUrlMap($url_map);
        $contact = $contact->toJson();
        return json_encode($contact, JSON_PRETTY_PRINT);
    }
}
```

```
    } else {  
        return false;  
    }  
  
}
```

4.1.3 Envoi des modifications

4.1.3.1 Client

Du coté client, je récupère les informations de mon formulaire pour les envoyer :

Avec juste une particularité pour mes images :

```
handleFileUpload(field, event) {  
    // Récupérer le fichier sélectionné  
    this[field] = event.target.files[0];  
},  
handleSubmit() {  
    const formData = new FormData();  
    formData.append("action", "editContact");  
    formData.append("address", this.address);  
    formData.append("rue", this.rue);  
    formData.append("commune", this.commune);  
    formData.append("pays", this.pays);  
    formData.append("urlImg_entete", this.urlImg_entete);  
    formData.append("url_map", this.url_map);  
    formData.append("urlImg_news", this.urlImg_news);  
    this.serviceHttp.editContact(  
        formData,  
        this.successEdit.bind(this),  
        this.errorEdit.bind(this)  
    );  
};
```

Pour l'instant les succès et error callback affiche uniquement des alerte, mais cela sera modifié par la suite :

```
successEdit() {  
    alert("Page modifier avec succès !");  
},  
errorEdit() {  
    alert("Echéc de la modification de la page !");  
},
```

J'envoie donc la requête avec mon service http :

```
editContact(formData, successCallback, errorCallback) {
```

```
$.ajax({
  type: "POST",
  url: this.URL,
  data: formData,
  contentType: false,
  processData: false,
  success: successCallback,
  error: errorCallback,
});
}
```

4.1.3.2 Serveur

Quand je reçois la requête je contrôle que mes données sont intactes, principalement si les images ont été envoyé correctement.

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    switch ($_POST['action']) {
        case 'editContact':
            if (isset($_POST['address'], $_POST['rue'], $_POST['commune'],
$_POST['pays'])) {
                $contact = new Contact(
                    $_POST['address'],
                    $_POST['rue'],
                    $_POST['commune'],
                    $_POST['pays'],
                );
                if (isset($_FILES['urlImg_entete'])) {
                    if ($_FILES['urlImg_entete']['error'] === UPLOAD_ERR_OK) {
                        $contact->setUrlImgEntete($_FILES['urlImg_entete']);
                    } else {
                        http_response_code(400);
                        echo json_encode('Erreur de chargement de : urlImg_entete');
                    }
                }
                if (isset($_FILES['url_map'])) {
                    if ($_FILES['url_map']['error'] === UPLOAD_ERR_OK) {
                        $contact->setUrlMap($_FILES['url_map']);
                    } else {
                        http_response_code(400);
                        echo json_encode('Erreur de chargement de : urlImg_entete');
                    }
                }
                if (isset($_FILES['urlImg_news'])) {
                    if ($_FILES['urlImg_news']['error'] === UPLOAD_ERR_OK) {
                        $contact->setUrlImgNews($_FILES['urlImg_news']);
                    } else {
                        http_response_code(400);
                    }
                }
            }
        }
    }
```

```
        echo json_encode('Erreur de chargement de : urlImg_news');
    }
}
$ctrlCms->editContact($contact);

} else {
    http_response_code(400);
    echo json_encode('Erreur de paramètre dans la requête !');
}

break;
```

Dans mon Ctrl, je vais contrôler que l'utilisateur connecté a bien le droit d'effectuer cette modification :

```
public function editContact($contact)
{
    if ($this->sessionManager->getCms()) {
        if ($this->wrkCmsPage->editContact($contact)) {
            http_response_code(204);

        } else {
            http_response_code(500);
            echo json_encode('Erreur de la modification de la page contact');
        }

    } else {
        http_response_code(401);
        echo json_encode('Accès refusé');
    }
}
```

Dans mon wrk, je vais commencer par me connecter en ftp pour pouvoir transférer mes images, si ces dernières sont différentes de null. Ensuite si l'image a été modifiée je récupère l'url de l'ancienne image et je la supprime.

Puis finalement je remets tous dans la db

```
public function editContact($contact)
{

    $ftp_server = "jc4pes.ftp.infomaniak.com";
    $ftp_username = " ";
    $ftp_password = "";
    $remote_dir = "/sites/node.claudemeylan.ch.claudemeylan.ch/assets/contact";
```

```
$conn_id = ftp_connect($ftp_server);
$login_result = ftp_login($conn_id, $ftp_username, $ftp_password);
ftp_pasv($conn_id, true);

if ($conn_id && $login_result) {
    // Changement de répertoire sur le serveur FTP
    ftp_chdir($conn_id, $remote_dir);

    if ($contact->getUrlImgEntete() != null) {
        if (
            getimagesize($contact->getUrlImgEntete()['tmp_name']) != false &&
            ftp_put($conn_id, basename($contact->getUrlImgEntete()['name']),
$contact->getUrlImgEntete()['tmp_name'], FTP_BINARY)
        ) {

            $contact->setUrlImgEntete($contact->getUrlImgEntete()['name']);
            //Suppresion de l'ancienne image
            $ancienUrlImgEntete = $this->connexion->selectSingleQuery('SELECT
urlImg_entete FROM `t_contact` WHERE 1', []);
            if (!empty($ancienUrlImgEntete)) {
                $ancienUrlImgEntete = $ancienUrlImgEntete['urlImg_entete'];
                if (ftp_delete($conn_id, $ancienUrlImgEntete)) {
                } else {
                    return false;
                }
            }

        } else {
            return false;
        }
    } else {
        //Si pas de nouvelle image, alors récupérer la valeur dans la db pour
la set
        $urlImgEntete = $this->connexion->selectSingleQuery('SELECT
urlImg_entete FROM `t_contact` WHERE 1', []);
        $urlImgEntete = $urlImgEntete['urlImg_entete'];
        $contact->setUrlImgEntete($urlImgEntete);
    }

    if ($contact->getUrlImgNews() != null) {
        if (
            getimagesize($contact->getUrlImgNews()['tmp_name']) != false &&
            ftp_put($conn_id, basename($contact->getUrlImgNews()['name']),
$contact->getUrlImgNews()['tmp_name'], FTP_BINARY)
        ) {
            //Suppresion de l'ancienne image

            $contact->setUrlImgNews($contact->getUrlImgNews()['name']);
            $ancienUrl = $this->connexion->selectSingleQuery('SELECT
urlImg_news FROM `t_contact` WHERE 1', []);
```

```
        if (!empty($ancienUrl)) {
            $ancienUrl = $ancienUrl['urlImg_news'];
            if (ftp_delete($conn_id, $ancienUrl)) {
            } else {
                return false;
            }
        }

        } else {
            return false;
        }
    } else {
        //Si pas de nouvelle image, alors récupérer la valeur dans la db pour
la set
        $urlImgNews = $this->connexion->selectSingleQuery('SELECT urlImg_news
FROM `t_contact` WHERE 1', []);
        $urlImgNews = $urlImgNews['urlImg_news'];
        $contact->setUrlImgNews($urlImgNews);
    }

    if ($contact->getUrlMap() != null) {
        if (
            getimagesize($contact->getUrlMap()['tmp_name']) !== false &&
            ftp_put($conn_id, basename($contact->getUrlMap()['name']),
$contact->getUrlMap()['tmp_name'], FTP_BINARY)
        ) {
            //Suppression de l'ancienne image
            $contact->setUrlMap($contact->getUrlMap()['name']);
            $ancienUrlMap = $this->connexion->selectSingleQuery('SELECT
url_map FROM `t_contact` WHERE 1', []);
            if (!empty($ancienUrlMap)) {
                $ancienUrlMap = $ancienUrlMap['url_map'];
                if (ftp_delete($conn_id, $ancienUrlMap)) {
                } else {
                    return false;
                }
            }
        } else {
            return false;
        }
    } else {
        //Si pas de nouvelle image, alors récupérer la valeur dans la db pour
la set
        $urlMap = $this->connexion->selectSingleQuery('SELECT url_map FROM
`t_contact` WHERE 1', []);
        $urlMap = $urlMap['url_map'];
        $contact->setUrlMap($urlMap);
    }

    ftp_close($conn_id);
```



```
        if (
            $this->connexion->executeQuery(
                'UPDATE `t_contact` SET
`urlImg_entete`=?,`address`=?,`rue`=?,`commune`=?,`pays`=?,`url_map`=?,`urlImg_news`=?
WHERE PK_contact=1',
                [
                    $contact->getUrlImgEntete(),
                    $contact->getAddress(),
                    $contact->getRue(),
                    $contact->getCommune(),
                    $contact->getPays(),
                    $contact->getUrlMap(),
                    $contact->getUrlImgNews()
                ]
            )
        ) {
            return true;
        }
    } else {
        return false;
    }
}
```

4.2 Problèmes rencontrés

Les problèmes rencontrés ont plus été l'apprentissage de nouvelle pratique, telle que l'utilisation de vue.js, ou de trouver une manière de comment gérer les images. Mais avec un peu de réflexion et de persévérance rien d'insurmontable.

4.2.1 Base de donnée

Mon plus gros problème a été durant le déploiement en prod de la base de données, impossible de se connecter, j'ai trouvé dans les configurations du serveur infomaniak que le port 3306 n'était pas ouvert par défaut...

Ports ouverts supplémentaires

<div>OUVRIR UN PORT</div> <div>RECHERCHER...</div>				
TYPE ▾	PORT ▾	IP / HÔTE ▾	DIRECTION ▾	DESCRIPTION ▾
TCP	3306	*	Entrée	test sql 1
TCP	3306	*	Sortie	test sql

A régler pour la prod

4.2.2 Session

Du fait que j'ai héberger mon front end et mon backend sur deux servers différents, j'ai été confronté à un problème de session, les sessionid n'était plus garder même avec le champ xhr. J'ai trouvé dans mes recherches un moyen de contourner ce problème en autorisant les cookies_samesite :

```
public function __construct()
{
    ini_set('session.cookie_secure', true);
    ini_set('session.cookie_samesite', 'None');
    if (session_status() === PHP_SESSION_NONE) {
        session_start();
    }
}
```

4.3 Tests fonctionnels

N° du test	Description	Résultat attendu
1	Chargement des informations dans la page (texte et image)	Les informations affichées sont celles contenues dans la bd
	Résultat obtenu	Conclusion

	OK	Les informations sont correctement récupérer
--	----	--

N° du test	Description	Résultat attendu
2	Changement de langue	Les textes se modifie, et la langue est gardé pour toutes les pages
	Résultat obtenu	Conclusion
	Pas encore implémenté	Fonctionne avec postman, juste pas encore mis en place pour le client

N° du test	Description	Résultat attendu
3	Modification d'un texte ou d'une image	Le texte et l'image sont remplacé par les nouveaux dans la db
	Résultat obtenu	Conclusion
	L'image précédente est supprimé et le texte remplacé dans la db	Fonctionne parfaitement

N° du test	Description	Résultat attendu
4	Ajout d'une collection	Création d'une entre collection dans la db
	Résultat obtenu	Conclusion
	Nous pouvons ajouter une collection	RAS

N° du test	Description	Résultat attendu
5	Ajout d'une montre	La montre est ajoutée dans la db
	Résultat obtenu	Conclusion
	Les montres sont ajoutables depuis le client	RAS

N° du test	Description	Résultat attendu
6	Affichage des variantes de la montre sur la page montre	Nous avons toutes les montres d'une collection avec la même id de variante sur la page

	Résultat obtenu	Conclusion
	Les variantes des montes, si existantes s'affiche	Pourrait améliorer le système en cas de non variante

N° du test	Description	Résultat attendu
7	Suppression d'une collection	La montre ne se trouve plus dans la db
	Résultat obtenu	Conclusion
	Une collection peut être supprimé, cela supprime également les images associé	Reste la mise en place de quoi faire des montres dans une collection qui va être supprimé

N° du test	Description	Résultat attendu
8	Refus d'ajout, suppression et modification si pas connecté	Le serveur répond avec un code 401
	Résultat obtenu	Conclusion
	En effet le serveur nous retourne un code 401	La session est bien gérer !

4.4 Hébergement

Notre projet étant perso, notre site est hébergé chez infomaniak, voici l'url de preview :
<https://jc4pesbides.preview.infomaniak.website>

Pour le backend, j'ai eu de la difficulté et peu de temps pour prendre en main l'interface d'infomaniak, je l'ai donc héberger ici :

<https://chattona.emf-informatique.ch/151/main.php>

5 Synthèse

5.1 Différences entre planning et réalisation

En raison de notre gros projet, et des difficultés rencontrées, nous n'avons pas du tout pu respecter le planning prévu. Nous nous y attendions de ne pas pouvoir finir dans les délais, mais les problèmes ont encore plus retardé ce dernier.

5.2 Conclusion

En conclusion, ce module m'a beaucoup plus, de plus le fait de pouvoir allier ce module avec un vrai projet pratique m'a motivé. De plus php est un langage que j'apprécie utiliser et dont je comprends l'utilité.