Janvier – Septembre 2024

Aurélien Plazzotta

Projet de validation de formation

RNCP niveau 6 Developeur web et web mobile

LANCEMENT D’UN SITE WEB POUR UBC CREATION

Création d’un site web dynamique en PHP, JavaScript, SCSS et Docker

|  |  |
| --- | --- |
| **Identifiants du document** | |
| Date de création | 2024-03-26 |
| Dernière mise à jour | 2024-08-28 |
| Status | Travail en cours |
| Version | 0.0.3 |
| License | ISC |

Table des matières

[Présentation de la structure 4](#_Toc175754390)

[I. Présentation de l’entreprise 4](#_Toc175754391)

[i. Origine 4](#_Toc175754392)

[ii. But de l’entreprise 4](#_Toc175754393)

[iii. Bureau et organisation 4](#_Toc175754394)

[iv. La concurrence 5](#_Toc175754395)

[v. Rôle du projet orienté web 6](#_Toc175754396)

[vi. Existant et fréquentation des prospects et clients 6](#_Toc175754397)

[II. Objectifs du site 7](#_Toc175754398)

[III. Cible du site 7](#_Toc175754399)

[i. Population type 7](#_Toc175754400)

[ii. Segmentation de la cible 8](#_Toc175754401)

[iii. Objectifs quantitatifs 8](#_Toc175754402)

[iv. Périmètre du projet 8](#_Toc175754403)

[v. Existant et hébergement du site web 8](#_Toc175754404)

[Spécifications techniques du site 9](#_Toc175754405)

[I. Contraintes techniques 9](#_Toc175754406)

[II. Inspiration et charte graphique 9](#_Toc175754407)

[Analyse et conception 10](#_Toc175754408)

[1. Pourquoi avoir choisi SysML ? 10](#_Toc175754409)

[2. Diagramme d’analyse 12](#_Toc175754410)

[III. Story map 0](#_Toc175754411)

[IV. User story 0](#_Toc175754412)

[V. Diagramme de définition de bloc 2](#_Toc175754413)

[VI. Modèle entité-relation 0](#_Toc175754414)

[VII. Diagramme d’activité 1](#_Toc175754415)

[Front-end 0](#_Toc175754416)

[I. CSS 0](#_Toc175754417)

[i. Conception adaptative (responsive design) 0](#_Toc175754418)

[ii. Media queries 0](#_Toc175754419)

[iii. Bootstrap 1](#_Toc175754420)

[iv. SCSS 2](#_Toc175754421)

[II. JavaScript 2](#_Toc175754422)

[v. Référencement web - SEO : *(viz. Search Engine Optimization)* 2](#_Toc175754423)

[vi. Navigation au clavier 3](#_Toc175754424)

[Back-end 7](#_Toc175754425)

[I. Administration de la base de données avec Adminer 7](#_Toc175754426)

[II. Patron de conception : MVC 7](#_Toc175754427)

[i. Rôle du contrôleur 7](#_Toc175754428)

[III. Mise en place du projet sous Symfony 8](#_Toc175754429)

[i. Installation du cadriciel 8](#_Toc175754430)

[ii. Création d’un projet 8](#_Toc175754431)

[iii. Installation du gestionnaire de dépendances PHP 8](#_Toc175754432)

[iv. Installation des dépendences 9](#_Toc175754433)

[v. Support du protocole TLS 9](#_Toc175754434)

[IV. MVC (Modèle Vue Contrôleur) 10](#_Toc175754435)

[vi. CRUD 10](#_Toc175754436)

[vii. Implémentation 11](#_Toc175754437)

[viii. Restriction des accès 18](#_Toc175754438)

[V. Sécurité du code 19](#_Toc175754439)

[i. La faille de sécurité XSS 19](#_Toc175754440)

[ii. La faille de sécurité CSRF 19](#_Toc175754441)

[ix. User 20](#_Toc175754442)

[x. Internationalisation 21](#_Toc175754443)

[VI. Tests unitaires 23](#_Toc175754444)

[i. Désinstallation préalable 23](#_Toc175754445)

[ii. Installation des dépendances 23](#_Toc175754446)

[iii. Fichier de configuration 23](#_Toc175754447)

[iv. Execution des tests 24](#_Toc175754448)

[v. Test fonctionnel 26](#_Toc175754449)

[Exploitation 28](#_Toc175754450)

[I. Containerisation avec Docker 28](#_Toc175754451)

[i. Pourquoi Docker ? 28](#_Toc175754452)

[ii. Rôle et bénéfices 28](#_Toc175754453)

[iii. Manifeste 30](#_Toc175754454)

[iv. DockerFile 32](#_Toc175754455)

[II. Système de gestion de contrôle de version 33](#_Toc175754456)

[i. Configuration de git 33](#_Toc175754457)

[ii. Lexikon des commandes git 34](#_Toc175754458)

[iii. Editeur de texte et extensions 36](#_Toc175754459)

[iv. Migration de la base de données 37](#_Toc175754460)

[v. Sauvegarde de la base de données : 38](#_Toc175754461)

[III. Législation 42](#_Toc175754462)

[vi. RGPD 42](#_Toc175754463)

[IV. Conclusion 44](#_Toc175754464)

[vii. Outils informatiques utilisés 45](#_Toc175754465)

[viii. Bibliographie 46](#_Toc175754466)

[V. License 47](#_Toc175754467)

[Annexes 48](#_Toc175754468)

# Présentation de la structure

## Présentation de l’entreprise

### Origine

L’entreprise existe sous la forme d’une SAS., soit Société à Action Simplifiée, créée et gouvernée par Monsieur Carlos de Sousa depuis 2013, localisée à 1795 rue de la Source 45160 OLIVET et immatriculée en préfecture au registre du commerce et des sociétés du Loiret.

L’entreprise est constituée d’une douzaine de salariés et génère un chiffre d’affaires annuel d’environ 1,2 millions d’euros.

### But de l’entreprise

L’entreprise propose des prestations dans deux domaines d’activités stratégiques :

1. Construction
2. Rénovation
3. Les projets de constructions font l’object des services suivants :
   1. Construction de maisons individuelles
   2. Maçonnerie générale tous corps d’état
   3. Gros oeuvre
   4. Aménagement d’intérieur et extérieur
   5. Isolation
   6. Aire de jeu
4. Les projets de rénovation couvrent les services associés aux :
   1. Habitations
   2. Aires de jeux existantes
   3. Salles de conférence
   4. Salles des fêtes
   5. Salles de réunion
   6. Théâtre

### Bureau et organisation

L’entreprise constituée de différentes couches de management et coeurs de métiers :

1. Direction : UBC Création est dirigée par une unique personne, sus-mentionnée dans le pagraphe précédent, savoir Carlos de Sousa, qui agit en qualité de chef d’entreprise.
2. Administation/ Comptabilité : Madame Emilie de Sousa assume le rôle de Responsable administrative et comptable.
3. Supervision d’équipe : Monsieur Jérôme DRULAC et Matthieu BRICOURT management chacun une équipe de 5 spécialistes
4. Les artisans : enduiseurs, plaquistes, couvreurs-zinceurs, menuisiers agenceurs, peintres.

#### Assurer le renouvellement d’un bassin de jeunes recrues :

Des candidats en contrat d’apprentissage en provenance du lycée professionnel Gaudier Brzeska à Saint Jean de Braye (Loiret), effectuent parfois un cursus au sein d’UBC CREATION et peuvent faire l’objet d’une embauche à l’issue de leur examen.

Cette coopération de 22 semaines permet à des jeunes de 15 à 19 ans issus des filières bac professionnel :

1.Aménagement et finition du bâtiment

2. Technicien du bâtiment : organisation réalisation du gros oeuvre

3. Technicien constructeur bois

de découvrir le métier dans des conditions favorables à leur épanouissement grâce à l’encadrement vigilant d’experts métiers reconnus dans la région Centre-Val de Loire.

L’entreprise maintient quant à elle, une communication institutionnelle avec les instances de l’éducation nationale et *in fine*, renforcer sa visibilité sur le marché local et maintenir une image d’entreprise engagée dans le renouvellement des compétences.

#### Piste exploratoire pour le perfectionnement des futures recrues :

Une amélioration des compétences des candidats à une possible issue d’un bac pro peut s’effectuer sous la forme d’une poursuite d’études via le cursus suivant :

1. Brevet de Technicien Supérieur Bâtiment

La formation est également dispensée par le lycée Gaudier Brzeska et fait l’objet d’un enseignement général de 10h/semaine et d’un corpus professionnel de 22h/semaine, sur deux années.

### La concurrence

Voici un bref récapitulatif des concurrents directs, indirects et de substitution d’UBC Création.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nom | Date de création | Nombre de salariés | Chiffres d’affaires (en milliers €) | Localité | Domaine de compétition |
| Jmml Renovation | 2020 | 5 | 870 | Saint Jean de la Ruelle | Rénovation, peinture, isolation |
| AD Construction | 2003 | 6 | 319 | Saint Jean de la Ruelle (45) | Maçonnerie |
| Queiros | 2000 | 3 | 143 | Saint Denis en Val (45) | Ravalement |
| Orelec | 1999 | 2 | 101 | Orléans (45) | Maçonnerie, rénovation |
| Jeux Kompan | 1991 | 3 | 205 | Châteauroux (28) | Aires de jeux |
| Altrad | 1985 | 25-30 | 16 144 | Béziers (34) | Aires de jeux |
| Husson | 2015 | 5-8 | 540 | Veigne (37) | Aires de jeux |

*Source : infogreffe.fr, societe.com, pappers.fr*

Le marché est atomisé et se compose d’acteurs nombreux et prêts à parcourir jusqu’à 75 km (en moyenne) pour effectuer des chantiers dans les départements limitrophes au Loiret : Loir-et-Cher, Eure-et-Loire, Essone, Seine et Marne, Yonne, Cher, Nievre.

De plus, nombre de concurrents bénéficient du soutien publicitaire et logistique d’un réseau national. Environ 90% des entreprises observées bénéficient d’ores et déjà d’un site web, de type vitrine, conduisant à deux constatations :

1. L’existence de vitrines commerciales en ligne cause une situation fragile pour UBC Création, qui manque d’apparaître dans les résultats de recherche pour les prospects utilisant internet comme point d’entrée dans leur démarche de sélection d’un prestataires de services.
2. Le contenu des sites web mentionnés au point précédent ne sont finalement que des «démonstrations statiques » (dit site vitrine) et de par leur absence totale de fonctionnalités et de dynamisme, présente une opportunité pour UBC Création pour se démarquer instamment et durablement grâce à la mise en ligne d’un site professionnel aux mécaniques riches et complexes.

### Rôle du projet orienté web

Le projet de site web s’inscrit dans une stratégie de conquête de parts de marché sur la région Centre-Val de Loire sur les prestations courantes et à forte marge commerciale pour le ravalement de façade, la petite grosse maçonnerie, réaménagement de salles de spectacle et la fabrication et pose d’aires de jeux.

Exposer un portfiolo en ligne permet, outre le gain en publicité et *in extenso* en visibilité, de consolider son image d’entreprise fiable et capable d’assumer des chantiers complexes afin de convaincre des acteurs exigents comme les communautés de communes.

Le troisième rôle visé par la création et le lancement d’un site web vise à fidéaliser nos clients durablement en les convainquant de faire appel à nos services sur des thématiques connexes et de féderer une communauté active de rapproteurs d’affaires.

### Existant et fréquentation des prospects et clients

Il n’existe pas d’existant associé à un actuel ou précédent site web. UBC Création ne possède ni n’héberge aucune vitrine web associé à ses activités.

Les communications entre prospects/clients et l’entreprise s’effectuent habituellement par voie téléphonique. Les SMS servent de canal secondaire et les courriers électroniques sont employés pour les communications plus formelles et/ou administratives.

Le bouche à-oreille favorable suite à la réalisation réussie des chantiers incarne le moyen priviligié de l’entreprise pour renforcer son portefeuille de clients et renouveller son carnet de commandes.

Le trafic prévisionnel est très faible et correspond aux dimensions fort modestes de l’entreprise, savoir plusieurs dizaines de visites par mois sont attendus.

Les offres premiers prix d’hébergeurs populaires répondront sans difficultés aux besoins de termes de bande passante et puissance computationnelle des serveurs nécessaires au maintien en conditions opérationnelles dudit site web.

L’entreprise ne dispose d’aucune statistique relative au taux de rebond et au taux de conversion en raison de l’absence d’un site web antérieur au présent projet.

Aucune page ou technologie n’est à récupérer pour assurer le succès du lancement du site. Seules des photographies d’anciens chantiers seront incorportés au catalogue visuel du projet.

## Objectifs du site

Le site web poursuit deux principaux enjeux :

Le projet s’apparente à une vitrine destinée à assurer la **promotion de l’entreprise** et à l’implémentation d’une charte graphique spécifique propice à la construction et au développement d’une identité visuelle aisément reconnaissable par les prospects et clients de l’entreprise.

Des fonctionnalités d’ajout, de lecture, modification et suppression de fiches techniques, photographies et messages assureront le renouvellement et la **maintenance et du portfolio, et du catalogue de chantiers** d’UBC Création, afin de communiquer le succès des prestations et le gage de sérieux de l’organisation.

Un **système d**’**actualités** pourrait voir le jour, et assurerait une première phase de communication institutionnelle, comme communiquée précédemment dans le présent dossier, à l’intention de futurs candidats à l’apprentissage dans la spécialité du bâtiment.

## Cible du site

### Population type

La population ciblée par le projet est entièrement constituée de particuliers (*i.e.* personnes physiques) propriétaires de leur logement ainsi que de communautés de communautés (*i.e.* personnes morales) en charge de projets d’urbanisme.

Leurs caractéristiques sociologiques sont hétérogènes bien que leur inclination technologique varie assez peu. Une majorité est à l’aise avec la navigation web et les outils de communication associés (notamment les champs de formulaire et les emails). Il s’agit de personnes âgées de 25 à 80 ans, de nationalité française, disposant d’un ordinateur personnel et d’une connexion internet stable et d’un débit suffisant disponible au quotidien.

### Segmentation de la cible

Il n’existe aucune classification, priorisation, découpage ou subdivision quelconque visant la poursuite de tactiques commerciales particulières à un segment de la population cible.

Chaque prospect et client est traité avec respect et considération quel que soit son *ethnè*, âge ou historique socio-culturel.

Les mêmes considérations sociétales s’appliquent aux futurs clients et potentiels apporteurs d’affaires.

### Objectifs quantitatifs

Le trafic attendu est considéré très faible au lancement, modeste durant les six premiers mois et vraisemblablement modéré durant les deux premières années. Le traffic potentiel est estimé à plusieurs centaines de visites par an.

Le taux de transformation est peu stratégique en l’état en raison d’un réseau personnel étendu de l’équipe dirigeante associé à une bonne réputation de l’entreprise.

Le choix de l’hébergeur est donc facilité. Les besoins en puissance du micro-processeur, la quantité de mémoire vive et de bande passante délivrées par le serveur web à louer sont donc très modestes.

### Périmètre du projet

Le site web est prévu pour être établi en langue française uniquement lors de son lancement en-ligne.

L’ajout de l’anglais ne fait l’objet d’aucun avantage comparatif, les locuteurs anglophones ne traitant nullement avec ce type d’organisation. Cette tendance est considérée négligeable et le besoin de traduction du site web en langue étrangère est *in fine*, inexistant.

Le **support pour périphériques mobiles** (*i.e.* smartphones et tablettes tactiles) est jugé *recommandé* et non urgent.

L’implémentation d’une solution logicielle pour web orienté bureau et PC portables s’effectue après le lancement de la version du site web ciblant les dispositifs mobiles.

### Existant et hébergement du site web

Il n’y a pas d’existant. Cette particularité du cahier des charges et des prestations adoptées est traitée en détail dans la sous-rubrique « Contraintes techniques » du présent document.

# Spécifications techniques du site

## Contraintes techniques

Le nom de domaine retenu par les dirigeants de l’association est : https//www.ubccreation.fr

L’hébergeur de données sélectionné est **planethoster**, disponible sur *www.planethoster.com*, reconnu pour sa fiabilité et ses prix abordables suite à des expériences concluantes liées à d’autres activités de ces mêmes membres.

Planethoster assure à la fois l’hébergement des données et le rôle du registrar, soit le fournisseur de noms de domaines en location.

L’offre intitulée « Hébergement web illimité » à 6 € TTC / mois, soit 72 € par an, a été retenue afin d’assurer les besoins d’hébergement du projet pour une période (initialement définie) de 1 an. Elle présente les caractéristiques suivantes :

Dans l’hypothèse où le trafic viendrait à augmenter durablement et de manière significative, le dirigeant de l’entreprise accepte la possibilité de souscrire à l’offre dite « Forfait Pro » facturée 2 € TTC supplémentaire par mois.

Le nouveau coût potentiel d’hébergement s’élèverait donc à 96 € TTC / an pour bénéficier d’un matériel plus performant :

* Environnement semi-dédié : Hébergement des données sur un serveur mutualisé avec très peu de comptes
* Serveur web Lightspeed engine en lieu et place d’ Apache
* Technologies LSCache et LiteMage natives

## Inspiration et charte graphique

La charte graphique présente de grandes libertés. La couleur noir est à proscrire (à l’exception de la police de caractères). Le orange peut être encouragé pour les éléments de navigation. Des dégradés peuvent être employés pour adoucier l’apparance globale et réduire la fatigue nerveuse des visiteurs grâce à des dégradés plus réduits.

# Analyse et conception

## Pourquoi avoir choisi SysML ?

Le langage de modélisation système SysML (*i.e.* System Modeling Language) a été préféré à son plus ancien homologue UML (i.e. Unified Modeling Language). En effet, SysML présente des atouts indéniables :

* Plus d’offres d’emplois : en raison de son caractère simplifié (et non moins riche !), plus proche des besoins actuels des organisations. La charge cognitive pour apprendre le langage est réduite.
* Courbe d’apprentissage plus rapide : son nombre réduit de diagrammes accélère sa prise en main par les programmeurs souhaitant s’orienter vers une carrière d’Analyse-Programmeur, Analyse fonctionnel ou Architecte Logiciel, grâce à son nombre réduit de diagrammes, soit 9 en lieu et place des 14 présents en UML ! Cela encourage son adoption au sein des organisations et réduit le risque encouru par ses employeurs dans l’hypothèse d’une pénurie de talents.
* Communication plus efficiente : les interlocuteurs non-techniciens impliqués dans le déroulement des premières phases du projet logiciel :
  + Phase exploratoire : Investigation de nouvelles opportunités, exploration de la faisabilité d’intégration d’une technologie et évaluation de l’adéquation entre les pre-concepts et les besoins des utilisateurs.
  + Concept : Identification des parties prenantes, et leurs besoins, évaluation de concepts alternatifs, recommandation de solutions possibles.
  + Développement (ne pas confondre avec le développement « programmatique » du futur logiciel) : production de calendriers détaillés, identification et management du risque et des opportunités métiers (y a –t-il un nouvel outil en interne, en sus du projet actuel et connecté à ce dernier, qui bénéficierait aux équipes et dont le rôle et ses possibles bienfaits sont insoupçonnés ?).
  + présents lors des réunions entre associées à l’analyse, l’expression et/ou l’architecture d’une solution logicielle existante ou nouvelle, A REVOIR] ont plus de facilité à comprendre SysML en raison d’une grande réutilisabilité de ses symboles.
* Modéliser des projets non logiciels : SysML permet de modéliser des voitures, réfrigérateurs, aires de jeux, et tout autres équipements dénués d’électronique (bien que les 3 examples cités peuvent faire se voir intégrer une couche de composants électroniques et d’un firmware associé).
* Moins monolitique, plus adaptable : SysML permet, contrairement à sa contre-partie UML, offre l’immense avantage de modéliser des systèmes adoptant des languages non-orientés objets.

En effet, le paradigme orienté objet étant sur le déclin depuis plus d’une décennie, à la fois en France, et surtout à l’international, présente des caractérisques de moins en moins séduisantes pour les nouvelles générations de programmeurs :

1. Le caractère monolitique et compliqué des hiérarchies de classes (aujourd’hui évitées de plus en plus), avec les mixin et enfin les traits (voir PHP et Rust)
2. Les structures de données muables sont remplacées au profits d’équivalents immuable (*e.g.* DateTimeImmutable)
3. Les iterateurs : remplacement l’usage des larges collections d’objets. Evalués superficiellement (*viz*. uniquement lors de leur appel), ces requièrent l’allocation d’un bloc mémoire nettement plus petit grâce à l’évaluation de chaque valeur uniquement lorsque ces dernières sont appelées.
4. Les fonctions anonymes : offrent une meilleure clarté d’une code tout en évitant la verbosité d’une function standard et sa possible encapsulation qui nécessite la compréhension d’un contexte supplémentaire. Ces dernières sont non-modélisables avec un language de modélisation entièrement orienté objet.

En conclusion, le SysML s’avère assez délicat à mettre en oeuvre à l’heure de la conception du projet dû à une refonte complète du langage par le OMG (*i.e.* Object Management Group) qui publie actuellement la version 2.0.0 Beta 2.

La littérature associée est donc limitée et les documentations peu nombreuses et assez indigestes, en raison du caractère sans cesse changeant de la syntaxe, qu’il s’agisse de la notation textuelle ou (nouvellement ajoutée) graphique du langage.

## Diagramme d’analyse

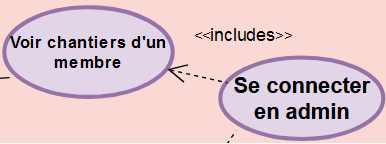
Ce diagramme presente le plus haut niveau d’abstraction logique entre la réflexion menant à l’identification des besoins et leur conditions requises à leur bonne exécution entre les différents acteurs et l’implémentation de la solution logicielle finale déployée sur le serveur de production.

Il s’agit de l’unique diagramme en SysML (et UML) permettant de modéliser une phase d’analyse. Il présente l’avantage d’être aisé à comprendre par tous les interlocuteurs du projet, quelque soit leur historique socio-culturel.

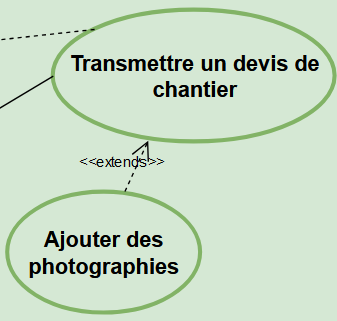
Ce diagramme d’analyse est essentiel pour s’assurer de la bonne cohésion entre les besoins des utilisateurs et le produit final. Il peut et doit être mis à jour au fil de l’évolution des besoins afin d’assurer la continuité du projet et la délivrance d’une valeur ajoutée aux opérationnels métiers toujours renouvellée.

*Lexikon* :

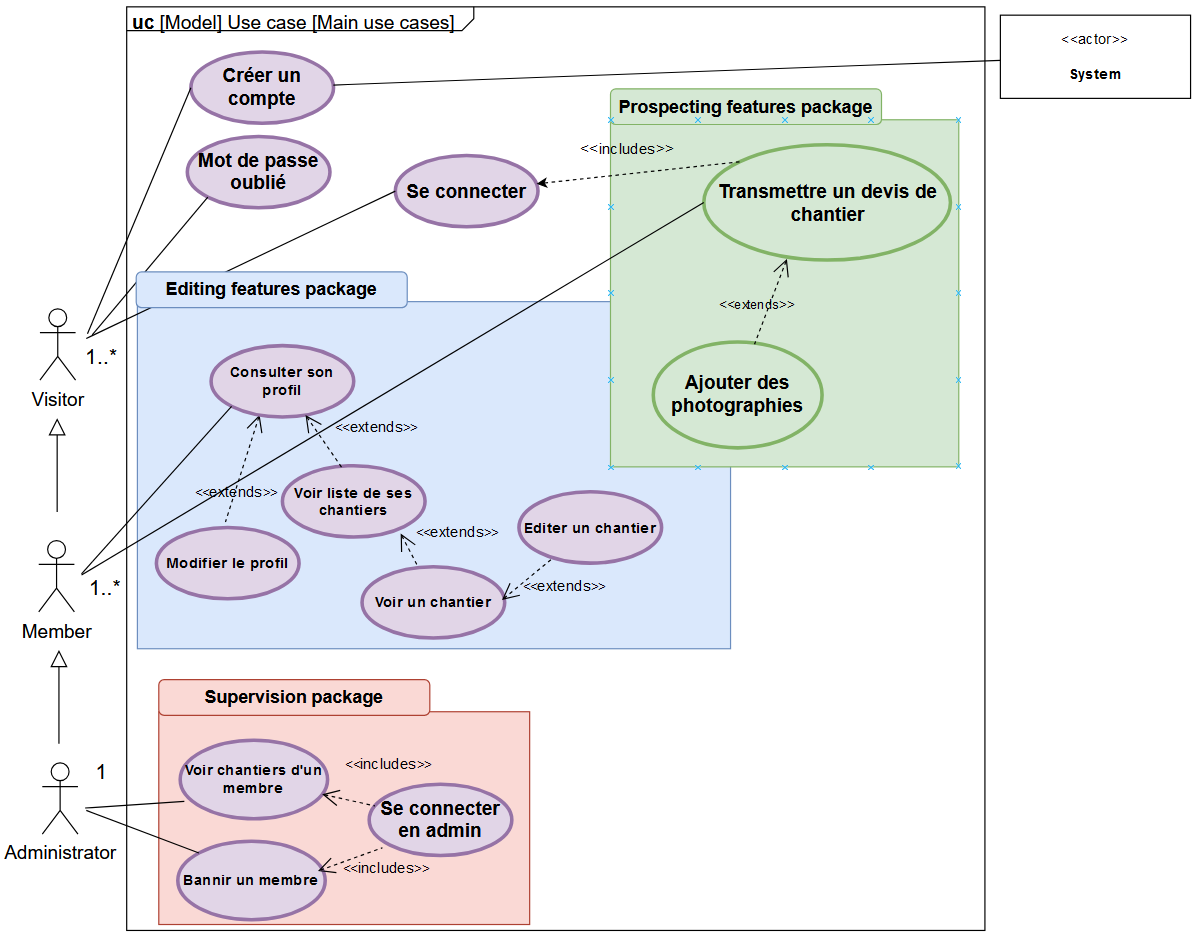
* **includes** : notation qui n’implique pas une relation de dépendance, mais plutôt que lorsque le scénario cible est invoqué, le scénario noté par la relation d’inclusion est lui aussi exécuté.



Voir les chantiers d’un membre requiert d’être connecté en administrateur.

* **extends** : relation qui informe que le scénario peut être optionellement exécuté après que le scénario cible, se suffisant à lui-même, est exécuté.

Ajouter des photographies de chantier est un scénario optionel rendu possible après la transmission d’un devis.



*Figure 1.3* – Diagramme de cas d’utilisation

## Story map

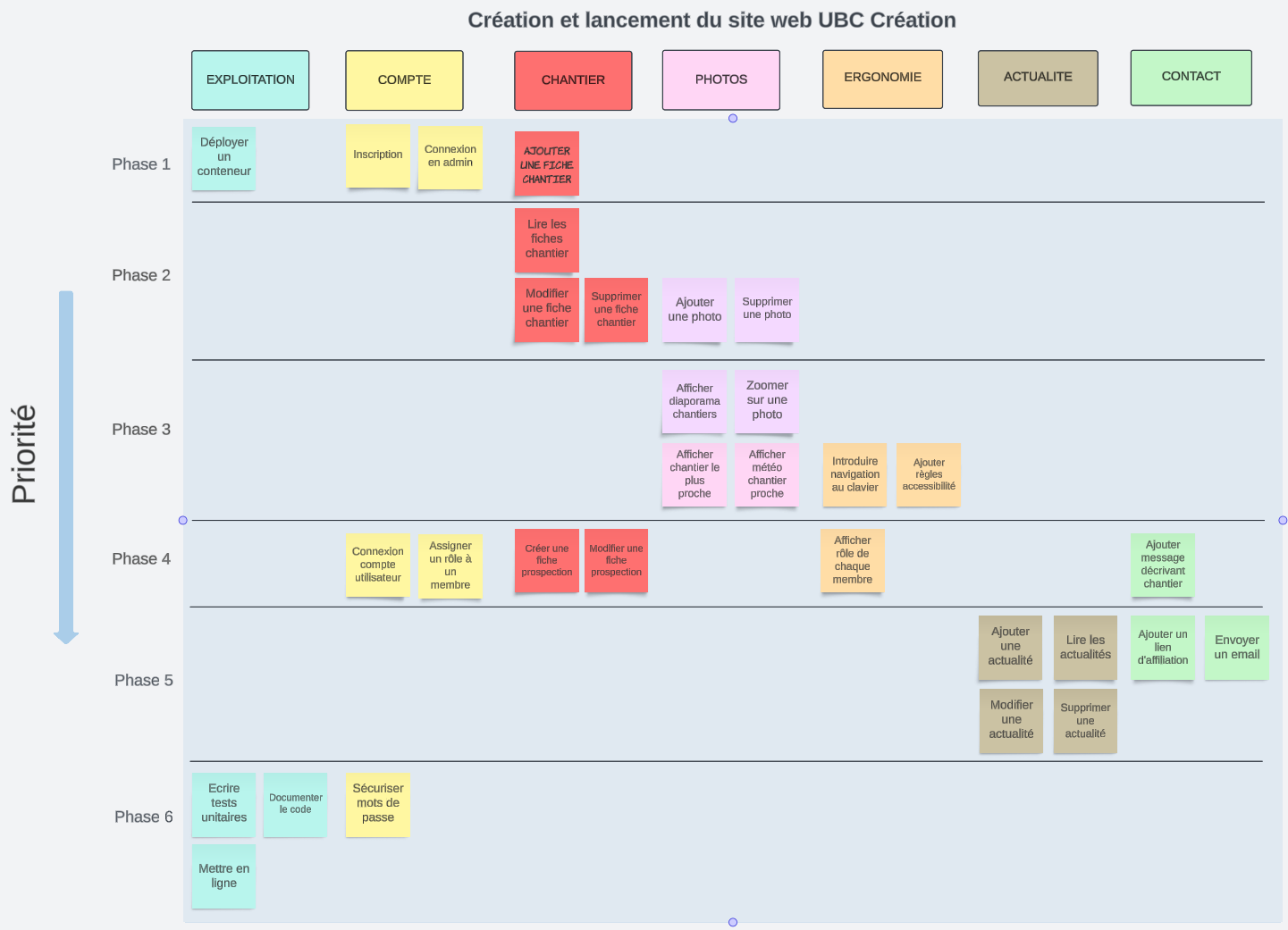
La « cartographie d’histoires utilisateur », ou *story map*, est un document composé des élements fonctionnels d’un projet de génie logiciel. Elle fait partie intégration des projets adoptant la philosophie embrassant le manifeste Agile et notamment sa méthode de management dite Scrum, elle même calquée sur la supervision d’une équipe de rugby.

Ce document ne requiert aucune aptitude technique et peut être lu et compris par toutes les parties prenantes du dit projet.

Son but est de **confirmer avec le client que les besoins de l’outil à développer ou à améliorer ont été bien compris** par l’équipe en charge du susdit développement.

La story map permet également à cette même équipe de décomposer le travail en de nombreuses sous-tâches afin de réduire la complexité technique et faciliter l’établissement d’un calendrier prévisionnel d’itérations logicielles.

Enfin ce document offre un troisième atout : celui de faciliter le coût en jours/homme pour un projet ou sous-projet, et ainsi déterminer le montant hors taxes à facturer au client.



## User story

Une « histoire utilisateur », ou *user story*, est l’un des composants de la story map. La user story représente une fonctionnalité à implémenter, voire un couloir fonctionnel, pour une user story dite « épique », laquelle requiert le découpage en sous-tâches, soit plusieurs user story.

Le but de ce découpage est de réduire et les délais d’implémentation, et la complexité technique, afin de pouvoir intégrer dans un *sprint* (période comprise entre deux semaines et deux mois avec une propension pour le premier).

La user story permet de maximiser la valeur utile délivrée au client tout en compartimentalisant le niveau de risque dans le cas où le besoin du client a changé ou lorsque sa spécification fonctionnelle est mésinterprétée par l’équipe en charge de son implémentation.

Voici un example de user story intégré au projet :

USER

STORY

* ID : 001
* Titre S’inscrire en créant un compte utilisateur
* Type : fonctionnel

###### Description

En tant que visiteur (*i.e.* futur utilisateur), je souhaite publier une fiche prospection afin de formaliser les besoins de mon chantier afin de permettre à l’entreprise d’établir un devis juste et des délais précis.

###### Règles métier

1. Créer un compte validé pour publier une description de mon chantier.
2. Seul un unique compte peut être créé par courriel.
3. Un compte utilisateur n’est validé qu’après un clic sur un clien de confirmation envoyé par courriel.
4. La création du compte et la gestion de ses données doit être conforme au Règlement sur la Gestion et Protection des Données européen (*i.e.* RGPD).
5. Saisir au moins huit caractères, dont un chiffre, un caractère spécial et un caractère alphabétique en capitale pour la définition du mot de passe.
6. Le lien de confirmation expire après 15 minutes.
7. Le lien de confirmation est unique et est désactivé après usage ou expiration.
8. Un code « captcha » est généré durant l’affichage du formulaire d’inscription et doit être saisi correctement pour confirmer la création du compte, sous réserve de la susmentionnée validation par courriel.

###### Tests d’acceptation

* Scénario n°1 : Lorsque je saisis un nom d’utilisateur déjà présent dans la base de données et finalise la création du compte, je reçois un message d’avertissemen, empêchant la confirmation de création.
* Scénario n°2 : Lorsque je saisis une adresse électronique déja existante dans la base de données utilisateurs, je reçois un message d’avertissement, lequel suspend également la confirmation de création de compte.
* Scénario n°3 : Lorsque je saisis un mot de passe d’une longueur de caractères inférieures à 8, sans caractère spécial et/ou sans majuscule ni chiffre, un message d’avertissement est affiché et empêche la validation du formulaire.
* Scénario n°4 : Je saisis un courriel erroné, c’est-à-dire qui existe mais ne m’appartiens pas, puis valide la création du compte.

Je ne peux valider la création du compte en raison de mon incapacité à cliquer sur le lien de confirmation (le courriel étant envoyé à une adresse dont je ne connais pas le mot de passe).

Avant l’expiration du lien sous quinze minutes, je saisis de nouveau le même nom d’utilisateur avec cette fois-ci le courriel désiré (*i.e.* m’appartenant bel et bien). Un message d’avertissement apparaît pour signaler que le nom d’utilisateur est déjà pris.

Ce dernier est de nouveau libre lorsque le lien de confirmation expire.

* Scénario n°5 : Je saisis un code « captcha » différent de celui aléatoirement généré. Un message d’avertissement est affiché, un nouveau code est généré et les champs de formulaire conservent leur données à l’exception du mot de passe.

*Nota bene :*

Les règles métier 3, 4 et 5 peuvent être assignées à une autre user story orientée vers la validation de compte, faisant de la présente user story une « épique ».

Le découpage est utile en équipe pour faciliter l’implémentation en parallèle ou dans des délai d’intégration courts.

## Diagramme de définition de bloc

Type de diagramme le plus courant en SysML, le diagramme de définition de bloc permet d’afficher différents genres de modèles d’élements et ses relations afin d’établir la structure d’un système.

Les élements modélisables, dit de définition, peuvent être l’un des genres suivants :

(*e.g.* interfaces, spécifications de flux, contraintes de blockes, types de valeurs, acteurs bloc)

* interfaces
* spécifications de flux
* contraintes de bloc
* types de valeurs
* acteurs
* bloc

Son (plus proche) équivalent en UML est le diagramme de classes.

*Lexikon* :



* Référence unidirectionnelle : un élément pout accéder à un autre élement de définition.

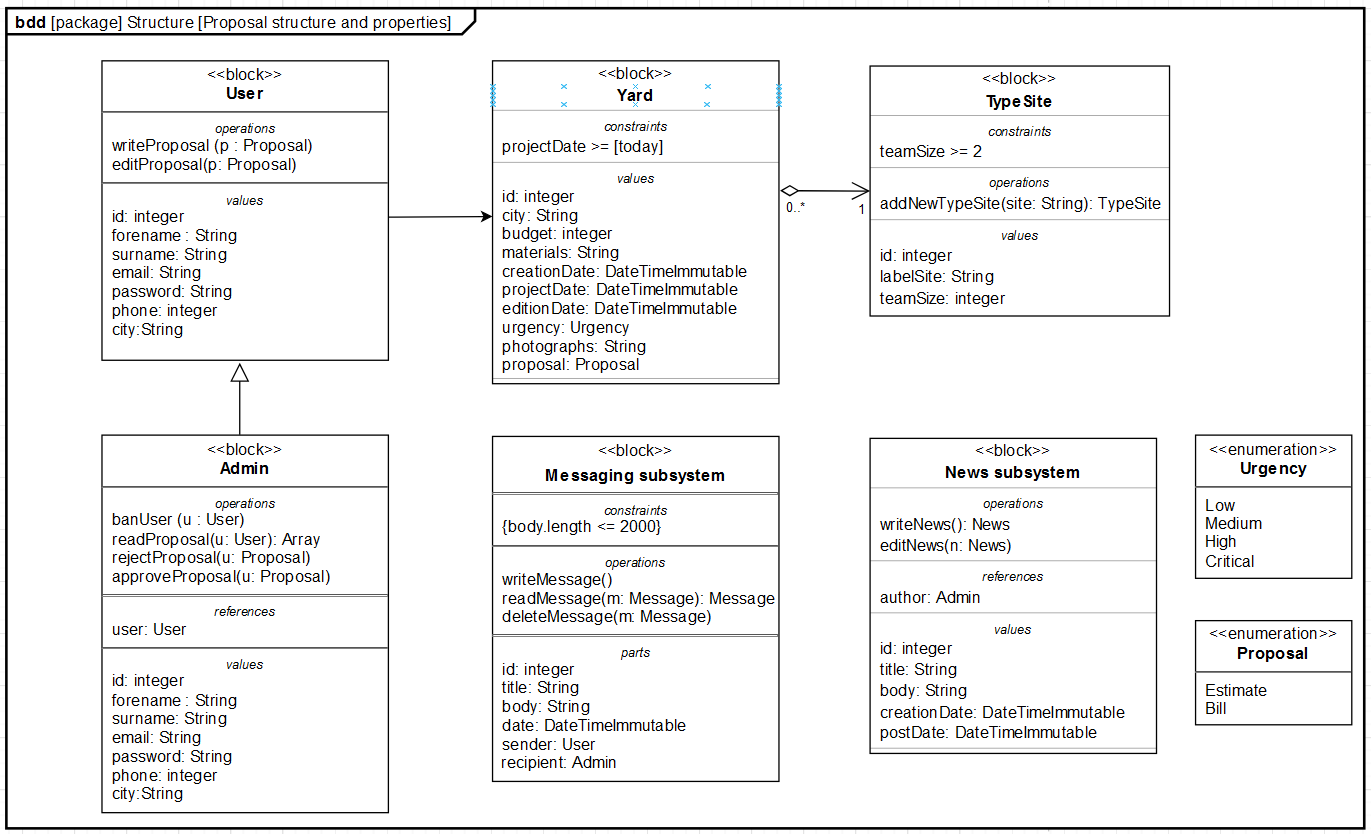


* Généralisation : un élément de définition hérite de toutes les propriétés et méthodes d’un élément parent.



* Composition : décomposition structurelle. La relation inclut les cardinalités. 0 ou n chantier est constitué d’une et une seule instance de type de chantier. La cardinalité minimum 0 est retenue afin de permettre la création de la table Chantier avant l’insertion de la table Type de chantier et éviter ainsi une violation d’intégration référentielle.

*Nota bene* : les couleurs sont inversées entre UML et SysML concernant l’association composite !



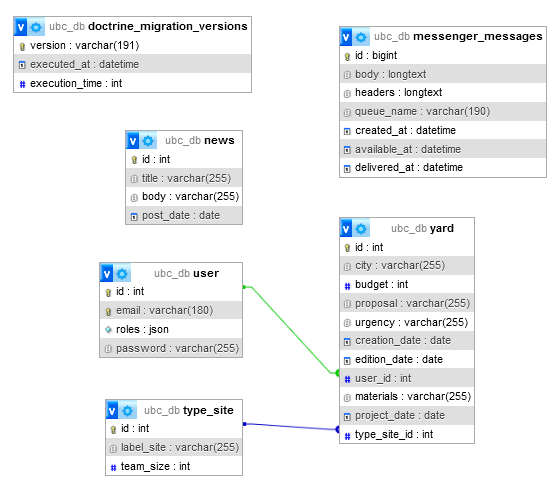
*Figure 1.3* – Diagramme de définition de bloc

## Modèle entité-relation

Le modèle entité-relation est généré par l’outil *Designer* du panneau d’information PHPMyAdmin.

Bien qu’ayant un rôle assez proche du diagramme de classes UML, du diagramme de définition de bloc SysML ou du modèle conceptuel de données issu de la méthodologie MERISE, le modèle entité-relation offre une une vue supplémentaire sur la cartographie des entités logiques constituant le modèle de données.

Voicu la vue d’ensemble du domaine métiermodélisé du projet ubccreation.fr ses relations entre différents tableaux de la base de données *via* les clés étrangères. Ces derniers sont des attributs permettant d’effectuer de communiquer avec une autre table grâce aux jointures.



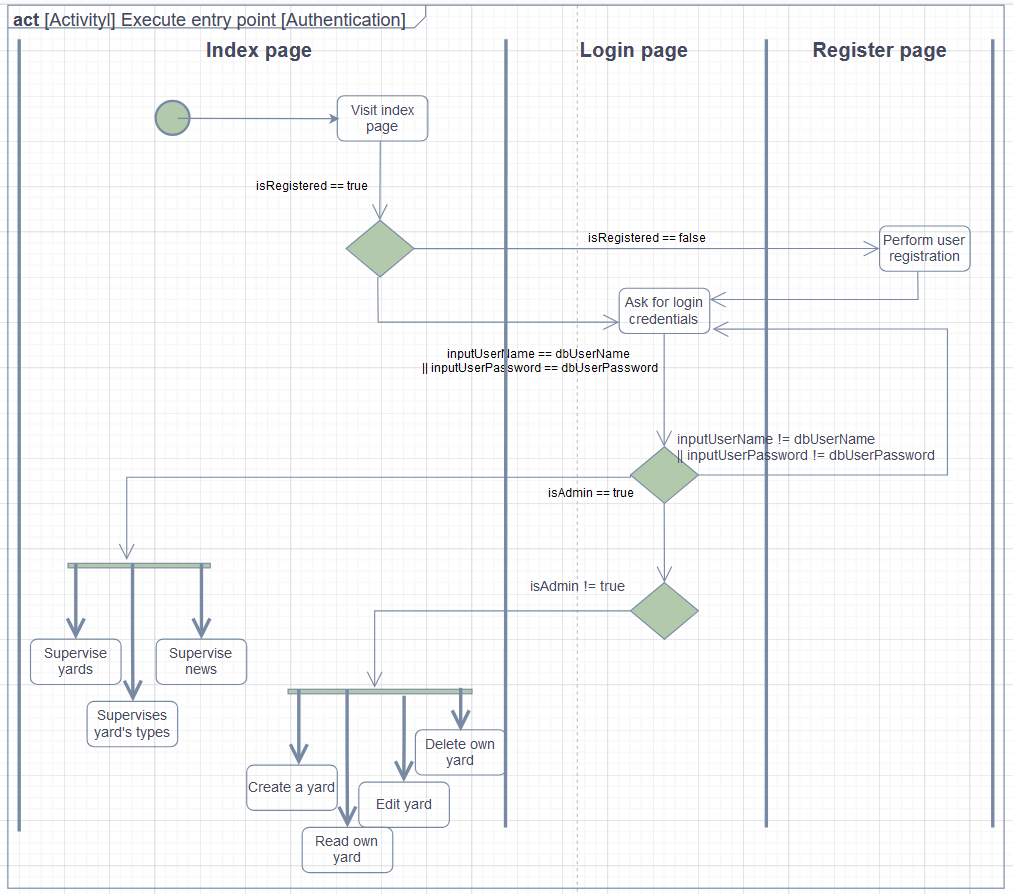
*Figure 1.5* – Diagramme de définition de bloc

## Diagramme d’activité

Outil d’analyse utile pour communiquer avec les parties prenantes et capturer le comportement attendu du système et de ses acteurs. Permet aussi la communication entre différents membres d’une même équipe technique pour assurer la cohérence entre les composants d’un sous-système.

Contrairement aux diagrammes structurels *(i.e*.définition de bloc, bloc interne, empaquetage et diagramme paramétrique), le diagrame d’activité est de type comportemental. Il intègre le passage du temps dans sa définition du comportement d’une ou plusieurs mécanique(s).

Le diagramme n’est pas lié au cycle de vie de l’application et peut être modélisé très tôt pour communiquer le périmètre d’un pan fonctionnel.



*Figure 1.4* – Diagramme d’activité

# Front-end

## CSS

Cascading Sheet Style, le CSS est un langage de présentation des données qui permet de styliser les contenus d’une page web en définissant des ensembles de règles qui peuvent s’appliquer conditionnellement à certains éléments de la page. Un ensemble de règles (*i.e.* ruleset) respecte la syntaxe suivante :

1\*n<selector> {

*// Declaration block*

property: value;

}

### Conception adaptative (responsive design)

Les résolutions des périphériques de sortie des utilisateurs peuvent être variables. Les visiteurs du site web peuvent user d’un ordinateur de bureau/portable, une tablette tactile ou un téléphone mobile. C’est pourquoi il est important (et même nécessaire si l’on souhaite éviter de générer une mauvaise impression qui potentiellement sont source d’opporunités commerciales manquées), d’adapter le contenu éditorial des pages en fonction de la résolution disponible.

Le cadriciel CSS Bootstrap, initialement une feuille de style diffusée gratuitement en source ouverte sous la license MIT en 2010, est intégré au projet Symfony pour offrir un style adaptable facile à implémenter.

### Media queries

Les media queries permettent d’appliquer conditionnellement un style CSS grâce aux règles @import et @media pour personnaliser l’apparence d’une page selon les résolutions et orientations du dispositif de l’utilisateur.

Comme mentionné plus haut, c’est le mécanisme qui permet d’implémenter le responsive design et font l’objet d’une spécification technique de niveau 5 publiée en décembre 2021 par le World Wide Web consortium : https://www.w3.org/TR/mediaqueries-5/.

En voici un simple extrait, bien que sujet à une couche d’abstraction insérée par Bootstrap :

@media (min-width: 1250px) and (orientation: portrait) {

  / \* Here goes the conditionally applied style \*/

}

@media (width <= 600px) {}

En effet, Bootstrap, inclut automatiquement ces media-queries via du sucre syntaxique dans les classes container et container-fluid.

### Bootstrap

#### Installation

Un projet implémenté en PHP pur (*viz.* "vanilla " PHP) aurait simplement nécessité l’ajout de deux lignes de code dans les en-tête d’une page HTML (ou du squelette de modèle élémentaire, soit templates/base.html.twig, selon qu’un moteur de template comme Twig ou Volt soit utilisé) :

<link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" *rel*="stylesheet" *integrity*="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" *crossorigin*="anonymous">

        <script *src*="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" *integrity*="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" *crossorigin*="anonymous"></script>

Figure 3.1 – Intégration de Bootstrap en "Vanilla"

Néanmoins, avec Symfony, son installation est encore simplifiée. Il suffit de saisir dans le terminal : composer require twbs/bootstrap

#### Configuration

Dans le fichier assets/styles/app.scss, le cadriciel Bootstrap et une feuille de style personnalisée sont tous deux importés. Le fichier inclut également toutes les variables primaires de Bootstrap qui sont personnalisées par le programmeur, selon la syntaxe : $<variable> : <propriété> ;

$primary: greenyellow;

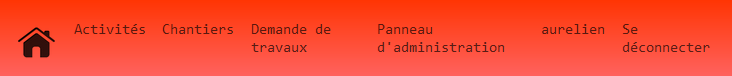
$card-bg: **rgb**(190, 169, 169);

$input-bg: #f7a18c;

@import '../../vendor/twbs/bootstrap/scss/bootstrap';

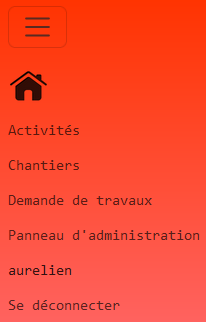
@import './custom.scss';

Voici un example démonstratif impliquant la barre de navigation.

Figure 3.1 - Large résolution : largeur **supérieure** à 600 pixels

Lorsque le moteur de rendu du navigateur web (*i.e.* Gecko pour Mozilla Firefox, V8 pour Chromium, Google Chrome, Microsoft Edge ou Webkit pour Apple Safari) détecte que le viewport (largeur effective en pixels, propriétée de l’objet document, lui-même noeud du DOM enfant de window), est inférieure à 600 pixels, la barre de navigation est réduite et un bouton cliquable apparaît pour "étendre"(*i.e.* expand) et "enrouler " (*i.e.collapse*).

* Barre déroulée :
* Barre enroulée :



### SCSS

Le pré-processeur CSS SASS est utilisé pour le projet afin de bénéficier de fonctionnalités étendues pour industrialiser et accélérer la maintenance et l’évolution des feuilles de styles CSS. Il gère l’extension .scss, deuxième génération de pré-processeur CSS, supplantées depuis par la 3e génération Less et la 4e nommée Stylus (extension .styl). Il intègre des mécaniques comme :

* gestion des variables
* imbrication
* partials et imports
* mixins (pour réutiliser une feuille de style)
* extensions/héritage (pour partager des jeux de propriétés).

Chaque mise à jour de la feuille .scss requiert une re-compilation afin d’être transpilée en .css pour être évaluée par le moteur de rendu du navigateur web : php bin/console asset-map :compile

#### Imbrication de CSS dans un template

Dans le but d’affiner la reconstruction du DOM (*i.e.* Document Object Model) et obtenir d’un template un style particulier qui a préscéance sur la feuille de style globale, une construction d’un élément HTML "à la volée" (*i.e.* inline) a été écrite comme suit:

{% apply **inline\_css** %}

    <style>

        .custom-modal {

            height: 100vh;

            position: relative**!important**;

            display: flex;

            align-items: flex-start;

            justify-content: center;

        }

    </style>

{% endapply %}

#### Bonnes pratiques

* Emploi des unités "vh" (*i.e.* viewport) et rejet des % pour assurer la cohérence avec différentes résolutions d’écran des périphériques utilisateur (pose les bases du responsive design).
* Usage de la paire propriété / valeur: `overflow : visible/overflow;` pour garantir un affichage contrôlé des contenus éditoriaux dépassant la fenêtre active du navigateur *(i.e.* objet JavaScript Document).
* Amélioration progressive grâce aux requêtes de fonctionnalité  pour styliser de manière conditionnelle des déclarations de bloc :
* @supports (display: grid) {
* .cgu-disclaimer {
* display: grid;
* grid-template-columns: 1fr 1fr 1fr;
* gap: 0.9rem;
* }
* .cgu-disclaimer ul {
* margin: unset;
* }
* }

Les deux blocs de déclarations ne sont évalués que si le navigateur supporte la valeur grid de la propriété display.

La valeur unset est équivalente à inherit pour les propriétés héritées et initial pour celles non héritées et évite une possible confusion.

## JavaScript

JavaScript est un langage de programmation originellement implémenté selon le paradigme orienté objet par prototype et interpreté. Depuis la publication du comité EcmaScript en 2015 (dit ES 6), il est dorénavant compilé juste-à-temps et orienté objet par classe.

JavaScript était depuis sa création réservé au front-end, c’est-à-dire exécuté coté client (par le moteur de rendu du navigateur web). La situation a changé lorsque Ryan Dhal a créé NodeJs en 2009, permettant d’éxécuter des applications web côté serveur écrites en JS.

En 2024, des successeurs comme Deno ou Bun gagnent progressivement en parts de marché et permettent d’exécuter également des bases de code en TypeScript, sur-ensemble de JS maintenu par Microsoft et Alphabet.

#### JavaScript asynchrone

Le JavaScript asynchrone en usant des promesses (*viz.* Promise type) est le choix retenu comme méthode moderne.  
L’ancienne méthode AJAX et de son API XMLHttpRequest est rejetée car elle est inutilement verbeuse. De plus, elle occasionne des fuites mémoire dûes à l’appel des classes objet Observable qui requièrent de mettre fin à la souscription de l’instance d’objet après son dernier appel et complique la maintenance.

const **readYardsLocation** = async function () {

   try {

      const response = await **fetch**('./yards.json');

      const locations = await response.**json**();

      if (response.ok) {

### Référencement web - SEO : *(viz. Search Engine Optimization)*

Le référencement web garantit l’affichage du site web dans les résultats de recherche des moteurs comme Bing, Google, DuckDuckGo, etc. et détermine la visibilité du projet sur internet. En effet, un site web merveilleusement implémenté avec les meilleures prestations du marché aux tarifs les plus bas ne génerera aucune visite, et in extension, ne produira aucun chiffre d’affaires, si personne ne connaît l’existence du site.

C’est pourquoi user de bonnes pratiques de codage permet d’améliorer nettement le référencement organique (*i.e.* contenu éditorial ni payé par des annonceurs, ni influencé par des moteurs de recherche ; et jugé le plus pertinent dans leurs algorithmes par ces derniers )dudit site et assurer son succès par le biais de nombreuses visites ciblées de la part de prospects.

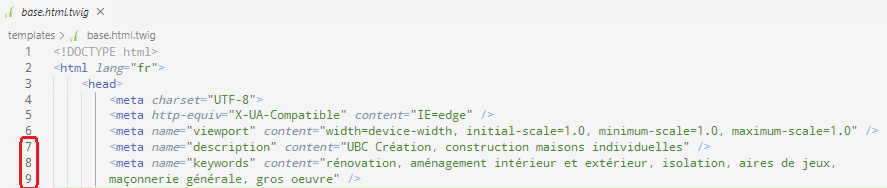
Deux balises HTML dans l’entête de requête améliorent le référencement dans le template élementaire template/base.html.twig

Figure 1.6 – Balises HTML d’en-tête liées aux référencement

#### Balises HTML sémantiques

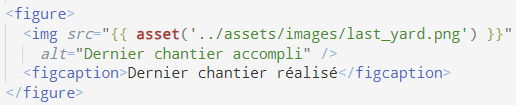
* <meta name="description": résumé court et précis du content de la page. Les moteurs de recherchent emploient cette valeur d’attribut pour contrôler l’apparence de la page web dans les résultats.
* <meta name= "keywords" : mot pertinents par rapport au content ;chaque valeur est séparée par une virgule.
* <header> : détermine un en-tête de contenu, généralement un titre et un sous-titre
* <nav> : contient le code associé exclusivement à la barre de navigation
* <section> : section de code indépendante dans la page
* <article> : portion du contenu éditorial relatif à un sujet spécifique
* <aside> : contenu auxilliaire au contenu princiapel d’une section, généralement une sous-section "pour aller plus loin " ou associé à une publicité.
* <footer> : pied de page contenant généralement les mentions légales, une cartographie du site (*viz.* sitemap) ou moyens de contact ou d’offres à pourvoir. Peut inclure les partenaires de l’entreprise.
* <search> :
* <canvas> :
* <audio> :
* <video>

figure / figcaption

### Navigation au clavier

Des balises HTML *tabindex* sont employées dans les templates afin de déterminer un ordre relatif au focus séquentiel de navigation. En des termes plus triviaux, appuyer répétitivement sur la touche *TAB* garantit l’application d’un focus sur des éléments déterminés dans un ordre jugé intuitif pour l’utilisateur.

Ce simple mécanisme assure au visiteur un retour visuel et garantit l’exploration de tous les contenus essentiels de la page. Voici un extrait du code CSS  et sa contre-partie HTML :

*// assets/styles/custom.scss*

.tab-nav > li:focus {

  outline: none;

  padding: 3px 0 3px 5px;

  margin: 5px 1px 3px 0;

  border: 1px solid **rgba**(81, 203, 238, 1);

  border-radius: 7%;

  box-shadow: 0 0 5px **rgba**(81, 203, 238, 1);

  max-width: 40%;

}

*// templates/services/index.html.twig*

<ul *class*="tab-nav">

   <li *tabindex*="5">Construction de maisons individuelles</li>

   <li *tabindex*="10">Maçonnerie générale tous corps d'état</li>

   <li *tabindex*="15">Gros oeuvre</li>

   <li *tabindex*="20">Aménagement d'intérieur et extérieur</li>

   <li *tabindex*="25">Isolation</li>

   <li *tabindex*="30">Aire de jeux</li>

</ul>

En sus de la navigation facilitée grâce à la balise HTML tabindex susmentionnée, une navigation (partielle) au clavier a été intégrée pour améliorer l’expérience utilisateur. A cet effet, deux raccourcis ont été implémentés :

* Ctrl + A : accéder à l’accueil
* Ctrl + D : déconnecter la session utilisateur.
* Escape : masquer l’affichage de la fenêtre modale relative aux conditions générales d’utilisation

L’implémentation en JavaScript consiste à pointer sur des noeuds HTML auxquels sont ajoutés des écoute d’évènements à déclencher lors de la pression sur des combinaisons de deux touches pré-déterminées.

###### Bonnes pratiques

Des bonnes pratiques ont été employées pour assurer un fonctionnement optimal, soit :

* un délai entre chaque pression au clavier avant d’éxécuter une fonction de rappel (*i.e.* callback) ;
* un vidage systèmatique de la structure de données chargée de conserver une trace des pressions enregistrées.
* une structure de données de type Set (dédupliquant automatiquement les item), plus adaptée et sémantiquement plus claire grâce à des méthodes efficientes et faciles à comprendre à la relecture (*viz.* add(), has(), delete())
* une cessation du comportement par défaut est intégrée pour éviter d’exécuter l’évènement utilisateur à deux reprises, et un dernier garde-fou pour assurer qu’aucun effet ne se produise si ce même évènement est déjà computé par le compilateur JavaScript (*i.e.* event.defaultPrevented).
* améliorer les performances d’affichage grâe à un chargement conditionnel d’un des scripts JavaScript uniquement lors lorsqu’une certaine route PHP est accédée par l’utilisateur.

"use strict";

import \* as Geolocation from './geolocation.js';

*/\* Global keyboard shortcuts \*/*

let pressedKeys = new **Set**();

let timer = null;

document.**addEventListener**("keydown", (event) => {

   let logout\_btn = document.**getElementById**('logout');

*// Populate the object with keys'state.*

   pressedKeys.**add**(event.key);

   if (event.defaultPrevented) {

      return;  *// Do nothing if event is already processed.*

   }

*// Clear any existing timer.*

   if (timer) {

**clearTimeout**(timer);

   }

*// Set a delay of 50ms before executing the callback to avoid triggering a combination of a single key.*

   timer = **setTimeout**(() => {

      switch (true) {

         case pressedKeys.**has**("Control") && pressedKeys.**has**("a"):

            window.location = 'http://localhost:80/'

            break;

         case pressedKeys.**has**("Control") && pressedKeys.**has**("d"):

            if (logout\_btn != undefined) {

            window.location = 'http://localhost:80/logout';

         }

            break;

         default:

            return;

      }

      event.**preventDefault**();

   }, 50);

   },

   true

);

*// Release the key when released to prevent further awkwardnesses.*

document.**addEventListener**("keyup", (event) => {

  pressedKeys.**delete**(event.key);

});

Figure 2.2 – Navigation partielle au clavier

{% block javascripts %}

    {{ **parent**() }}

    {% if app.request.attributes.**get**('\_route') == 'app\_register' %}

        <script *type*="module" *src*="{{ **asset**('public/assets/js/cgumodals.js') }}"></script>

    {% endif %}

{% endblock %}

{% block body %}

Figure 2.3 – Chargement conditionnel d’un scriptrelative à la navigation au clavier

#### Ergonomie

Une police de caractères lisible et universelle est retenue pour servir le contenu éditorial du site.

Elle est de taille petite pour afficher beaucoup de contenu avec un besoin en défilement d’écran réduit, disponible dans tous les navigateurs modernes et lisse pour assurer un espacement facile à déchiffrer :

font: small 'Lucida Bright', monospace;

# Back-end

## Administration de la base de données avec Adminer

L’outil de pilotage de la base de données SQL retenu est Adminer, disponible à l’adresse : https://www.adminer.org/. Le choix d’une telle solution face à son concurrent plus populaire PhPMyAdmin s’est fait naturellement :

* Parfois plus rapide pour certains traitements, e.g. les requêtes d’insertion.
* Portabilité extrême : Adminer présente la particularité unique d’exister sous la forme d’un unique fichier PHP, rendant la solution d’administration facile à déplacer, copier, sauvegarder et archiver.
* Prise en main facile : sa courbe d’apprentissage et requiert moins d’effort pour atteindre un niveau opérationnel que sa contr-partie PhpMyAdmin, peut-être plus complet mais plus difficile à utiliser, et maîtriser.

## Patron de conception : MVC

Le patron de conception (*viz.* design pattern) Modèle-Vue-Contrôleur a été adopté afin d’atteindre les objectifs suivants :

* Meilleure lisibilité : la base de code est plus claire à lire et comprendre par les développeurs actuels et futurs.
* Meilleure modularité : les problèmatiques d’accès à la base de données, la couche métier et la couche visuelle directement accessible aux visiteurs sont tous trois implémentés sous forme de composants réutilisables. L’application devient alors moins monolitique et réduit les coûts et délais de maintenance.
* Flexibilité technologique : ce découplage réduit aussi la dépendance de la base de code à une technologie spécifique et permet l’expérimentation et l’intégration d’autres langages.

Par example, migrer vers un autre système de base de données n’impacte que le composant Contrôleur.

Autre example, adopter un nouveau cadriciel CSS, comme basculer de Bootstrap à TailwindCSS voit son impact contingé au seul composant Vue et réduit le risque d’introduire des régressions lors d’une nouvelle itération logicielle.

### Rôle du contrôleur

Le contrôleur reçoit une requête HTTP (*i.e.* classe Request) et communique avec les modèle, service et constructeur de formulaire puis retourne une réponse HTTP (*i.e.* classe Response) contenant une page HTML.

Il ne génère pas nécessairement une vue, lorsqu’il produit par example une réponse au format JSON, XML ou CSV.

## Mise en place du projet sous Symfony

Symfony est un cadriciel écrit en PHP et doté une couche d’abstraction forte pour accélérer et normaliser le développement de projets web pour la partie back-end. Il intègre des outils d’interfaçage entre la couche métier et la base de données comme doctrine et la création d’entités (*i.e.* les classes définissant le domaine métier) comme marker-bundle.

### Installation du cadriciel

Téléchargement de l’installateur hébergé dans l’infrastructure infonuagique de l’éditeur, lancement du script d’installation et création d’un nouveau projet

**curl** -1sLF ‘https://dl.cloudsmith.io/public/symfony/stable/setup.deb.sh’ | sudo –E bash

**sudo** apt-install symfony-cli

### Création d’un projet

L’option --webapp est recommandée car elle installe des paquets additionnels et ajoute des fichiers de configuration orientés web (*e.g.* jsconfig.json, tsconfig.json) pour affiner le comportement de l’éditeur et la gestion de ses dépendances logicielles ainsi que les paramètres du compilateur TypeScript.

**symfony** new ubccreation --webapp

### Installation du gestionnaire de dépendances PHP

composer est l’outil de management de dépendences des projets PHP. Son script d’installation est téléchargé, testé contre sa clé d’identification hashée fournie par l’éditeur, installé, supprimé puis une copie dans le sous-dossier des portables exécutables binaires du système de fichiers local est effectuée pour rendre la commande disponible partout dans l’arborescence de vie (viz. system-wide) :

php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"

php -r "if (hash\_file('sha384', 'composer-setup.php') === 'dac665fdc30fdd8ec78b38b9800061b4150413ff2e3b6f88543c636f7cd84f6db9189d43a81e5503cda447da73c7e5b6') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP\_EOL;"

php composer-setup.php

php -r "unlink('composer-setup.php');"

sudo mv composer.phar /usr/local/bin/composer

### Installation des dépendences

L’outil composer permet de manager les dépendances écrites en PHP. Sa syntaxe inclut optionnellement le dépôt du mainteneur d’un paquet, dont le nommage explicite est parfois nécessaire pour éviter les ambiguités de résolution de téléchargement lorsqu’un conflit survient en raison de deux paquets nommés de manière identique.

*Nota* : l’option –dev implique une version *beta* incluant les derniers changements, bien que potentiellement moins stable et/ou moins sécurisés.

composer require symfony/orm-pack

composer require --dev symfony/maker-bundle

composer require form validator security-csrf

symfony composer req phpunit --dev

composer require symfony/asset-mapper symfony/asset symfony/twig-pack

composer require symfony/security-bundle

composer require symfony/serializer-pack

composer require twig/cssinliner-extra

composer require --dev symfony/test-pack

composer require symfonycasts/sass-bundle

composer require twbs/bootstrap

composer require symfonycasts/verify-email-bundle

composer require symfony/asset-mapper symfony/asset symfony/twig-pack

twig-pack est un meta-paquet incluant le coeur de la bibliothèque twig, l’intégration avec Symfony twig-bundle et des extensions additionnelles (*e.g.* autocomplétion) extra-bundle.

verify-email-bundle permet de chiffrer le mot de passe utilisateur lors de la soumission d’un formulaire en méthode POST via la commande :

**php** bin/console make :registration-form

### Support du protocole TLS

Enfin, le support du protocole de chiffrement de couche réseau sécurisée est activé et offre les communications entre le navigateur web et le serveur web via l’établissement d’une communication en HTTPS grâce à la commande :

**symfony** server:ca:install

#### Configuration de la base de données SQL

La configuration de la base de données relationnelle s’effectue dans le fichier .env situé à la racine du projet.



Ce fichier permet (entre autres choses) de basculer de l’environnement de développement à celui de production, en changeant la valeur de la variable APP\_ENV. Le mécanisme de redirections bascule alors de la gestion d’exceptions (en développement) à une redirection HTTP vers une page simple affichant le code de status (en production).

Cela évite ainsi de dévoiler le code source coupable et communiquer *in fine*, une surface d’attaque potentielle à un utilisateur malveillant.

La variable DATABASE\_URL communique des informations importantes à la couche d’interfaçage Doctrine :

1. Le type de base de données SQL
2. Le nom d’utilisateur
3. Le mot de passe
4. Le nom du service Docker associé à la base de données
5. Le port de mappage entre le monde extérieur (accédé depuis le navigateur web du visiteur) et le conteneur virtuel
6. Le nom de la base de données
7. La version de la base de données
8. Le format d’encodage de caractères.

## MVC (Modèle Vue Contrôleur)

### CRUD

#### Définition

Le CRUD incarne les quatre opérations essentielles au fonctionnement basique d’une entité logique entre l’utilisateur final et le programme qui interagit avec la base de données. Cet acronyme signifie Create Read Update Delete, soit la fonctionnalité de création d’une instance du modèle, sa possibilité de lecture de ses enregistrements, la mise à jour et bien sûr sa suppression *logique* (vis-à-vis des utilisateurs finaux) ou *physique* (suppression définitive depuis la base).

Chaque modification du modèle de données (*i.e.* App\Entity) requiert une re-écriture des requêtes SQL à migrations/Version<datetime>.php et son exécution pour insertion dans la base de données relationnelle dans ubc\_dc/doctrine\_migration\_version/.

Ces deux opérations font l’objet de commandes séparées énoncées ci-dessous et doivent être exécutées directement dans le conteneur Docker depuis le service associé la base de données :

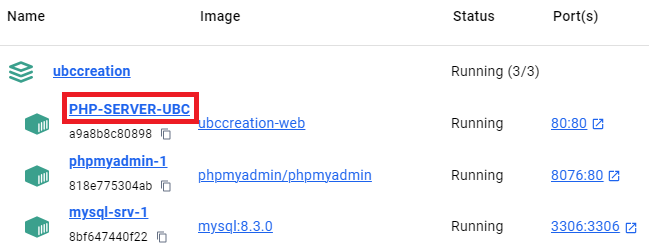


Figure 1.4 - Conteneurs en cours d’éxecution dans Docker Desktop

Rendez-vous dans l’onglet Exec puis sous-dossier de travail correspondant au projet :

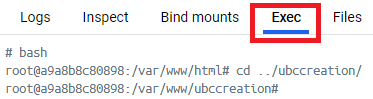


Figure 1.5 - Interface en ligne de commande du service contenant la couche métier

**php** bin/console make:migration

**php** bin/console doctrine:migrations:migrate

### Implémentation

Un contrôleur permet de manipuler le modèle de données (ou meta modèle, traduisant le domaine métier) afin d’actualiser la vue accessible à l’utilisateur. Ce même contrôleur fait l’objet d’un chargement du micro-kernel lorsqu’une route est accédée par l’utilisateur.

La route est un mécanisme issu du patron de conception MVC possédant un nom et une URI (*i.e.* Uniform Resource Identifier), le nom formel pour adresse internet.

#### Création

En voici un extrait chargé de la création d’un chantier par un utilisateur authentifié :

#[Route('/new', name: 'app\_yard\_new', methods: ['GET', 'POST'])]

public function **new**(**Request** $request, **EntityManagerInterface** $entityManager): **Response**

    {

*/\*\**

*\* Creates a new yard, which requires to be an authenticated user.*

*\*/*

    if (!$this->**isGranted**('ROLE\_USER'))

    return $this->**render**('yard/need-to-register.html.twig');

    $yard = new **Yard**();

    $form = $this->**createForm**(**YardType**::class, $yard);

    $form->**handleRequest**($request);

    if ($form->**isSubmitted**() && $form->**isValid**()) {

      $yard->**setCreationDate**(new **DateTimeImmutable**('now'));

      $yard->**setUser**($this->**getUser**());

      $yard->**setProposal**(**Proposal**::Estimate);

      $entityManager->**persist**($yard);

      $entityManager->**flush**();

      return $this->**redirectToRoute**('app\_yard\_index', [], **Response**::HTTP\_SEE\_OTHER);

}

        return $this->**render**('yard/new.html.twig', [

            'yard' => $yard,

            'form' => $form,

        ]);

    }

Figure 2.5 – Contrôleur pour la création d’un chantier

Cette route intègre un commentaire de documentation pour faciliter la maintenance du fichier.

Un contrôle d’accès est en place pour garantir qu’un visiteur non enregistré/ non connecté ne peut accéder à cette fonctionnalité et se voit rediriger vers la page d’inscription.

Un objet User est instancié et l’utilisateur actuellement connecté est récupéré grâce à son jeton de stockage (obtenu depuis l’interface UserInterface) et un nouveau formulaire est créé. Lorsque ce dernier est soumis et tous ces noeuds enfant (*i.e.* champs de formulaire) sont réputés valides, des champs invisibles comme la date de création (usant d’une classe immuable, une autre bonne pratique) et le type de chantier, lu depuis un type Enumerate (fonctionnalité récente issue de PHP 8.1), sont ajoutés en mémoire avant injection.

L’objet instancié de l’interface EntityManager appelle la méthode persist() pour intégrer de manière permanente les changements dans la table Yard de la base de données.

Ce même objet appelle ensuite la méthode flush() pour purger les données restées en cache afin de synchroniser l’état mémoire et éviter les comportements incohérents associées à des requêtes futures.

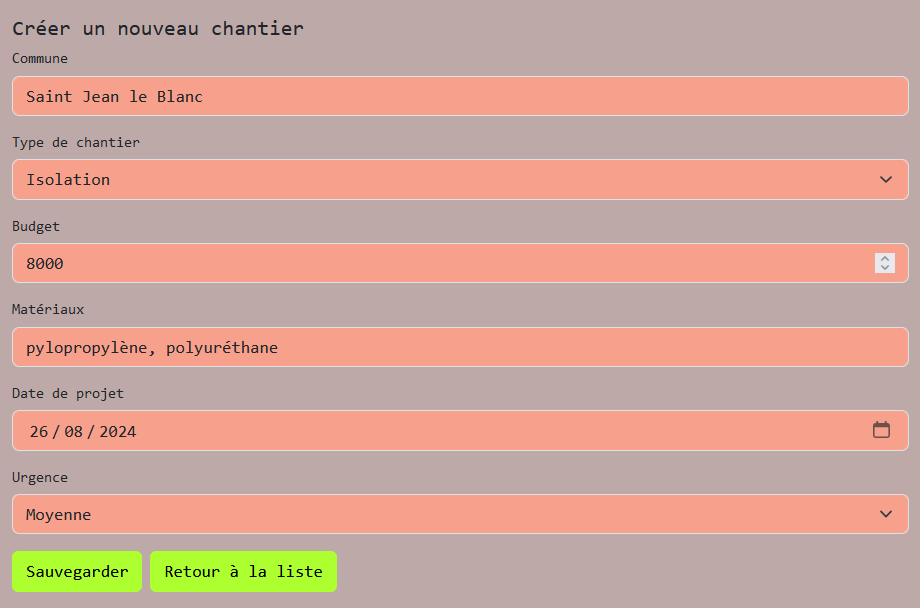
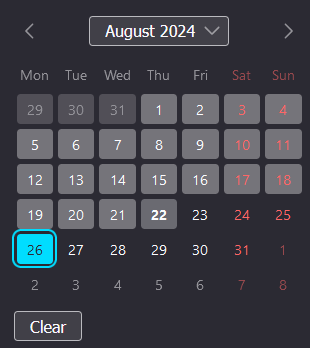


Figure 2.6 – Création d’un chantier

L’utilisateur remplit les champs du formulaire après avoir cliqué sur le menu Demande de travaux de la barre de navigation.

Les champs "types de chantier " et "urgence" font l’objet d’une liste déroulante garantissant la normalisation des données saisies par l’utilisateur avant injection dans la base de données.

Le champ "date de projet "de type calendrier, intègre quant à lui, une règle PHP empêchant la saisie d’une date incohérente pour fixer un projet dans le passé. Voir les cases grisées sur la capture d’écran ci-jointe.

Et son implémentation en PHP dans le fichier de construction du formulaire :

*# src/Form/YardType.php*

public function **buildForm**(**FormBuilderInterface** $builder, array $options): void {

  $builder

->**add**('projectDate', null, [

   ‘ label' => 'projectDate',

   ' widget' => 'single\_text',

    ‘attr' => **array**(

    'min' => (new **\DateTimeImmutable**('now'))->**format**('Y-m-d'))])

##### SQL

La requête de **création** correspondante exécutée en SQL est la suivante :

INSERT INTO `yard` (`id`, `city`, `budget`, `proposal`, `urgency`, `creation\_date`, `edition\_date`, `user\_id`, `materials`, `project\_date`, `type\_site\_id`) VALUES (NULL, 'orléans', '5000', '', 'medium', '2024-08-26', NULL, '1', 'bois, ciment', '2024-08-28', '4');

#### Lecture

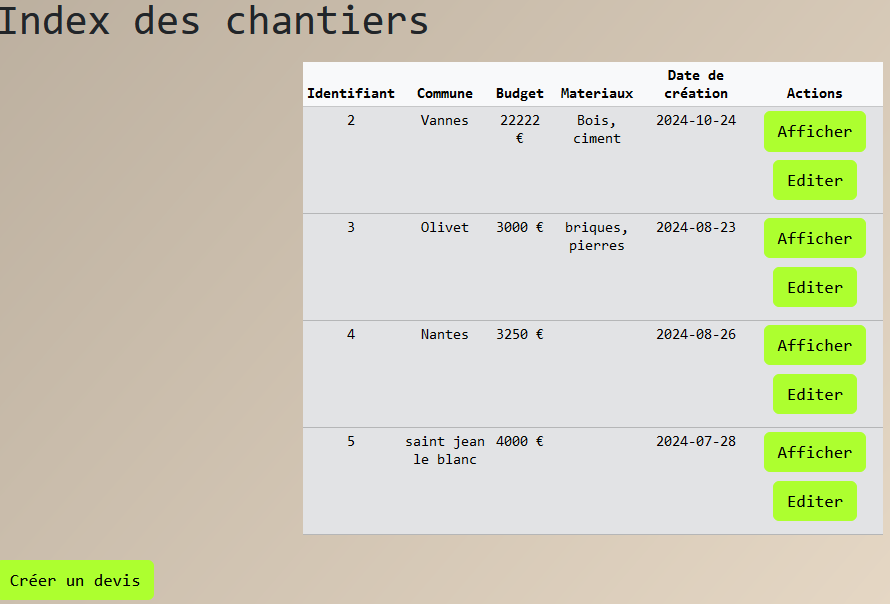


Figure 2.7 – Lister les chantiers crées auparavant par un utilisateur

#[Route('/yard')]

class **YardController** extends **AbstractController**

{

    #[Route('/', name: 'app\_yard\_index', methods: ['GET'])]

    public function **index**(**YardRepository** $yardRepository): **Response**

*/\*\**

*\* Provides an entry point to yards' managing feature. List existing yards.*

*\*/*

    {

        $this->**denyAccessUnlessGranted**('ROLE\_USER');

*# List only this user's yards.*

            $user = $this->**getUser**();

            return $this->**render**('yard/index.html.twig', [

                'yards' => $yardRepository->**findBy**(['user' => $user]),

            ]);

}

La route se nomme /yard. En effet, depuis la branche 7.x du cadriciel Symfony, les attributs #[] remplacent dorénavant les annotations /\*\* \*/, mieux intégrées sémantiquement et plus proches de la syntaxe naturelle de PHP.

Cette route intègre d’autres sous-routes, possédant une URI (*i.e.* chemin d’accès), un nom et possiblement une méthode de soumission, en l’occurence GET, laquelle transmet des données *via* l’URL, soit une ou plusieurs paires de clés valeurs selon la nomenclature :

<protocole>://<sous-domaine><domaine>.<extension>/<route> ?<clé1>=<valeur1>&<clé2>=<valeur2>

Un contrôle d’accès en PHP est appliqué grâce à la méthode denyAccessUnlessGranted() appliquée à l’instance du YardController (via le mot clé this) contrôle que l’actuel utilisateur possède le rôle ROLE\_USER, lequel est stocké dans la colonne roles de la table user.

Le fichier de configuration config/packages/security.yaml ajoute par défault une méthode getRoles() assurant que chaque utilisateur se voit attribuer par défault le rôle minimum de l’utilisateur afin de passer avec succès les contrôles d’accès et effectuer les actions courantes auxquelles peut s’attendre un utilisateur standard. Ce paquet est installé grâce à la commande :

composer require symfony/security-bundle

public function **getRoles**(): array

    {

        $roles = $this->roles;

*// Ensures that every user has at least ROLE\_USER*

        $roles[] = 'ROLE\_USER';

        return **array\_unique**($roles);

    }

*Nota bene* : les accolades au début de leur propre ligne (en sus d’autrespratiques comme les alias d’espace de nom sous la forme \<NamespaceName>(\<SubspaceNames>\)\*\<ClassName>) font partie des conventions d’écriture imposées par PSR-4 relatif à l’auto-chargeur. Les spécifications techniques sont maintenues par le comité inter-groupes PHP Framework Interop Group accessible à l’adresse : https://www.php-fig.org/psr/psr-4/.

L’utilisateur est récupéré via son jeton de session puis une instance de YardCOntroller génère un squelette de modèle (i.e. template) auquel il transmet comme paramètre tous les résultats trouvés selon le critère utilisateur grâce à la méthode findBy() invoquée depuis l’objet yardRepository.

Cet objet, qui hérite de la classe ServiceEntityRepository, possède un constructeur simplifié et contient toutes les requêtes SQL nécessaires (en usant du mécanisme createQueryBuilder()) aux cinq opérations élémentaires de création, lecture, mise à jour, suppression et recherche par critère.

##### SQL

La requête de **lecture** correspondante exécutée en SQL est la suivante :

SELECT \* FROM yard WHERE user LIKE ‘:id’

*Nota bene*: LIKE est la syntaxe canonique et inclut la gestion des détrompeurs pour filter des chaînes de caractères *(e.g.* ‘%dupont%’ pour inclure les résultats contenant "dupont" ou ‘Saint%’ pour tous les noms commençant par "Saint").

#### Mise à jour

Une fonctionnalité de mise à jour est implémentée pour permettre aux utilisateurs de modifier un ou plusieurs champs d’un chantier existant.

#[Route('/{id}/edit', name: 'app\_yard\_edit', methods: ['GET', 'POST'])]

public function **edit**(**Request** $request, **Yard** $yard, **EntityManagerInterface** $entityManager): **Response**

{

*/\*\**

*\*  Edits a existing yard to update current value(s) and/or add a nullable value.*

*\*/*

    $this->**denyAccessUnlessGranted**('ROLE\_USER');

    $form = $this->**createForm**(**YardType**::class, $yard);

    $form->**handleRequest**($request);

    if ($form->**isSubmitted**() && $form->**isValid**()) {

      $entityManager->**flush**();

      return $this->**redirectToRoute**('app\_yard\_index', [], **Response**::HTTP\_SEE\_OTHER);

        }

   return $this->**render**('yard/edit.html.twig', [

     'yard' => $yard,

     'form' => $form,

  ]);

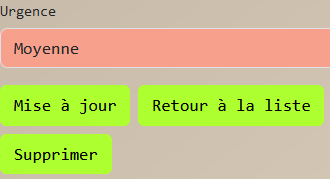
}

##### SQL

La requête de **mise à jour** correspondante exécutée en SQL est la suivante :

UPDATE yard SET city = :city WHERE id LIKE :id

#### Suppression

Enfin, la possibilité de supprimer une entrée existante est introduite et accessible par l’utilisateur *via* un simple bouton.

Présentement, supprimer un chantier requiert de la part de l’utilisateur de se rendre préalablement dans le menu d’édition.

#[Route('/{id}', name: 'app\_yard\_delete', methods: ['POST'])]

    public function **delete**(**Request** $request, **Yard** $yard, **EntityManagerInterface** $entityManager): **Response**

    {

*/\*\**

*\* Deletes a specific yard from the database.*

*\*/*

        $this->**denyAccessUnlessGranted**('ROLE\_USER');

        if ($this->**isCsrfTokenValid**('delete' . $yard->**getId**(), $request->**getPayload**()->**getString**('\_token'))) {

            $entityManager->**remove**($yard);

            $entityManager->**flush**();

        }

        return $this->**redirectToRoute**('app\_yard\_index', [], **Response**::HTTP\_SEE\_OTHER);

    }

La méthode remove() exécute une requête SQL pour supprimer l’enregistrement dans la base de donnée et prend en paramètre l’identifiant du chantier, via le getter getId(), à la condition que le jeton d’identification soit réputé valide. La méthode statique de la classe AbstractController, dont hérite la classe métier YardController, retourne un booléen.

Enfin, une redirection vers la route app\_yard\_index est effectuée en transmettant au navigateur web dans l’en-tête HTTP un objet réponse avec le code de status HTTP 303 (usant d’une constante associée à Symfony dans le cas présent). Le code 303 redirige le client pour obtenir la ressource à délivrer vers une autre adresse (*i.e.* URI) selon la méthode GET.

##### SQL

La requête de **suppression** correspondante exécutée en SQL est la suivante :

DELETE from yard WHERE id LIKE ‘:id’

### Restriction des accès

Il existe trois possibilité de restreindre l’accès des utilisateurs de l’application et diffèrent en termes de flexibilité selon le langage adopté (*i.e.* XML, PHP et YAML). La possibilité via un fichier XML est écarté, le YAML permettant un contrôle plus holistique et la version PHP permet d’offrir une granularité plus fine.

*# config/packages/security.yaml*

security:

    access\_control:

        - { path: ^/admin, roles: ROLE\_ADMIN }

        - { path: ^/yard/(show|edit|delete)\d+$, roles: ROLE\_USER  }

Le contrôle d’accès écrit en YAML présente une structure de donnée de type *map* qui associe une expression rationelle spécifiant le chemin d’accès et le groupe d’utilisateur (i.e. administrateur ou simple utilisateur). Son avantage est que ces paramètres impactent globablement toutes les routes définies dans le projet.

Afin d’assurer un contrôle affiné des permissions réservées aux groupes d’utilisateurs, un contrôle d’accès est inséré en PHP dans les contrôleurs de composants métier. En voici un extrait tiré de Controller/YardController.php :

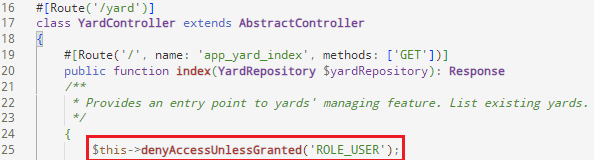


Figure 2.1 – Sécurisation de patron URL

***Nota Bene***: il est également possible d’adopter la méthode de définition des hiérarchies des rôles. Cette approche n’as pas été étudiée par manque de temps.

## Sécurité du code

Des surfaces d’attaque peuvent apparaître dans la base de code tout au long du cycle de vie du projet logiciel dû à des fonctionnalités (ou même des pans fonctionnels) présentent des failles de sécurité qu’un client ou tierce partie non autorisé (*i.e.* un programmeur malveillant) peut exploiter afin d’accéder à des données sensibles du serveur web. Le vol ou la corruption de données personnelles ou même injecter du code malicieux pour voler des informations saisies par l’utilisateur ou encore saturer la bande passante et/ou capacité computationnelle du microprocesseur hébergeant le serveur web du site sont des scénarii courants.

C’est pourquoi il est de la responsabilité du programmeur de s’assurer de la résolvation ou du moins de la mitigation des failles de sécurité de la base de base de code afin de protéger à la fois la protection de la confidentialité des données transmises par les utilisateurs et le maintien en conditions opérationnelles du serveur web et in fine, garantir une prestation de qualité avec un taux de disponibilité du service surpassant les 99% (la règle des cinq neufs recquiérant une main d’oeuvre très qualifiée ainsi que l’implémentation de mécanismes de redondance des baies serveurs et grappes de disques durfaisant partie du domaine fonctionnel de l’administrateur réseau et systèmes).

De plus, la continuité d’activité est un critère de pérénité pour le site web développé et déployé et impacte directement l’image de l’entreprise qui l’exploite.

### La faille de sécurité XSS

Dite "Cross-Site Scripting", cette surface d’attaque consiste à injecter du code écrit en JavaScript dans une requête chaînée (*viz.* query string) alimentée après l’URI dans une requête HTTP GET transmise dans l’en-tête de requête par le navigateur au serveur web, sous la forme : "?name1=value1&name2=value2".

Une ancienne méthode (dépréciée car utilise ke jeu d’encodage de caractères ISO-9951-1) consiste à remplacer les caractères spéciaux suivants : **&**, **<**, **>**, **"** et **‘** par leur entités correspondantes, afin qu’ils soient affichés et nonexécutés comme du code.

<a href= "login.php "htmlentities({{ userName }} )”;>Log in</a>

Une méthode privilégiée est l’emploi de  htmlspecialchars() ;

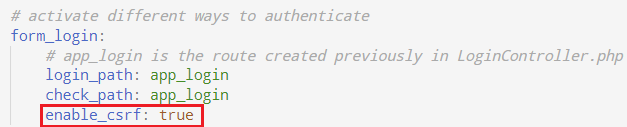
### La faille de sécurité CSRF

Dite "Cross-Site Request Forgery", il s’agit d’un type de vulnérabilité permettant à un attaquant de forcer un utilisateur légitimement authentifié à effectuer des actions indésirables à son insu.

Cette faille est contrecarrée par l’emploi d’un champ de formulaire HTML masqué dans le template associé à la route app\_login, invoquant la règle de sécurité lors de la soumission du formulaire pour générer un jeton d’identification.



Dans config/packages/security.yml, un provider est écrit pour référencer le jeton avec les règles augmentées de sécurité, prévenant l’usage de la faille CSRF. Le provider mappe via une structure de données éponyme, écrite en YAML, l’entité User avec la clé, qui est le jeton de référence lors de la création d’une session d’identification.



### User

UserInterface crée une jonction (*i.e.* un bridge) entre l’interface et la classe métier grâce à sa définition dans le fichier security.yml qui établie des règles de sécurité. Le jeton identifiant l’utilisateur actuellement connecté est récupéré depuis sa session utilisateur.

Dans l’entête de requête HTTP (objet Request) du navigateur web de l’utilisateur, le jeton est transmis au serveur web

### Internationalisation

La base de code est entièrement écrite en anglais pour assurer sa compréhension et in fine, sa maintenabilité par un bassin de programmeurs internationaux dans l’hypothèse où la direction de l’entreprise UBC Création décide d’embaucher ou faire appel à des consultats freelance pour prendre en charge l’évolution du site web durant la poursuite de leurs activités. Limiter l’écriture du code en français prive effectivement l’entreprise de programmeurs talentueux disponibles partout dans le monde et à des tarifs compétitifs.

C’est pourquoi le concept d’internationalisation du projet de génie logiciel prend son sens : les champs de formulaires écrits dans les vues et affichés sur les moniteurs et/ou périphériques utilisateurs doivent être traduits dans une langue adaptée au segment de marché ciblé, savoir le français.

Symfony offre un process complet pour normaliser la traduction de données et garantir le succès de son internationalisation :

##### Activer les traductions dans Symfony

Le composant de traduction doit être activé dans le projet via l’édition du fichier config/packages/framework.yaml :

framework:

*# Enable french translation*

translator: { default\_path: '%kernel.project\_dir%/translations'}

##### Créer un fichier de traduction

Il faut maintenaint créer un fichier de traduction par langue, soit translations/messages.fr.yaml :

##### Créer un fichier de traduction

Ajouter la traduction française pour chaque champ de tous les formulaires :

title: 'Titre'

body: 'Corps du message'

postDate: 'Date de publication'

city: 'Commune'

typeSite: 'Type de chantier'

budget: 'Budget'

materials: 'Matériaux'

projectDate: 'Date de projet'

urgency: 'Urgence'

labelSite: 'Type de chantier'

teamSize: "Taille d'équipe"

Low: 'Faible'

Medium: 'Moyenne'

High: 'Elevée'

Critical: 'Critique'

Estimate: 'Devis'

Bill: 'Facture'

##### Configurer les paramètres de localisation

Il faut s’assurer que l’application fasse appel à la localisation française par défaut, *via* l’édition du fichier config/packages/translation.yaml :

framework:

    default\_locale: fr

#### Format d’encodage UTF-8

Le format d’encodage UTF-8, spécifiquement utf8mb4, est retenu pour le projet car il cible un segment de marchand constitué de peuples communiquant en alphabet latin.

Ce nom dérive de *Universal Coded Character Set + Transformation format*, soit jeu de caractères encodés universel et format de transformation basé sur la longueur d’adresse mémoire 8 bit.

Selon le réseau documentaire revu par pairs *infogalactic.com*, l’UTF-8 est le format d’encodage de caractères dominant pour le consortium Word Wide Web en septembre 2015 et est employé sur 85.1% des pages web.

Il a pour rôle d’assurer le support d’un large panel de langues vivantes et langues mortes pour assurer l’affichage correct et uniforme des caractères quelque soit le périphérique de sortie de l’utilisateur.

Le précédent format dominant était ISO-8858-1 et couvrait 69% de la population globale. Néanmoins, son usage a nettement reculé dû à l’émergence des peuples ayant dorénavant accès au réseau des réseaux.

L’UTF-8 est un format qui stocke chaque caractère sur une largeur d’adressage 8 bit (soit 256 valeurs possibles). Il allie compatibilité et universalité, bien qu’affichant un manque d’efficience dans le cas où l’alphabet dominant est cyrillique. L’UTF-8 va effectivement consommer deux fois plus d’espace mémoire qu’un format comme Windows-1251.

Intégrer le format d’encodate UTF-8 pour garantir le succès de l’internationalisation du site web fait l’objet d’une intégration dans plusieurs fichiers :

*# php.ini*

default\_charset = “UTF-8  
internal\_encoding = “UTF-8”

output\_encoding = « UTF-8”

*# in every .PHP files*

header(‘Content-Type: text/html; charset=UTF-8’);

*# in HTML headers*

<meta charset=”UTF-8”>

*# in HTML headers*

<meta charset=”UTF-8”>

*# in vanilla HTML forms*

<form accept-charset=”utf-8”>

*# or in Symfony-rendered HTML forms*

{{ form\_start(form, {‘accept-charset’: ‘UTF-8’}

## Tests unitaires

Les tests unitaires représentent une phase essentielle de tout projet de génie logiciel. Ils garantissent à la fois la non-régression d’une itération logicielle et sa conformité avec le comportement attendu défini dans les spécifications techniques et/ou fonctionnelles.

Cette garantie est toutefois limitée par le taux de couverture de tests.

Dans l’hypothèse où la couverture de test est égale à 100%, c’est-à-dire que tous les appels de fonction, évaluation des déclaration, branches conditionelles et alternatives (i.e. if/ else...if / else, switch et match) ainsi que la gestion des exceptions ont été évalués. Ce score de 100% est invraisemblable et ne saurait être atteint dans des grands projets, où seules les fonctionnalités critiques des composants essentiels peuvent être raisonnablement testés dans le respect des délais, budgets et des périmètres fonctionnels.

### Désinstallation préalable

Dans un environnement existant et dénué de conteneurisation, il est judicieux de préalablement supprimer toute installation système relative au paquet PHPUnit :

vendor/phpunit/phpunit/phpunit --generate-configuration

vendor/phpunit/phpunit/phpunit --migrate-configuration

sudo apt remove --purge phpunit

### Installation des dépendances

Ajout de paquets élémentaires, ainsi que d’extensions pour les structures de données PHP (depuis PHP 7.X). L’objectif est de remplacer l’usage des array par des vecteurs, deque, map et set pour réduire l’empreinte mémoire et le nombre de cycles processeurs pour certaines computations :

sudo apt install -y php-cli php-json php-mbstring php-xml php-pcov php-xdebug zip

composer [global] require phpunit/phpunit --dev

composer require php-ds/php-ds

### Fichier de configuration

Le fichier phpunit.xml se situe à la racine du projet et détermine le comportement de l’outil de tests. En voici le contenu :



La partie encadre correspond à une personnalisation du fichier afin d’affiner le comportement de l’outil :

1. Annule la création et restauration de variables globales après chaque test
2. Affichage des couleurs dans l’émulateur de terminal
3. Définit le cadriciel CSS à exécuter avant le lancement des tests
4. Transforme les dépréciations en exception afin de causer l’échec immédiat du test. Utile pour capturer tôt les mécanismes dépréciés du langage.

### Execution des tests

Rendez-vous dans le service Docker associé au serveur web PHP.La commande suivante permet de lancer tous les tests unitaires écrits :

.vendor/bin/phpunit --testsuite units

Ils peuvent également être générés automatiquement une fonctionnalité expérimentale de Symfony lors de la création d’une entité au moyen de (par example) : php bin/console make :entity yard

#### Test unitaire

Un test unitaire permet de s’assurer qu’une fonction produit l’effet attendu.

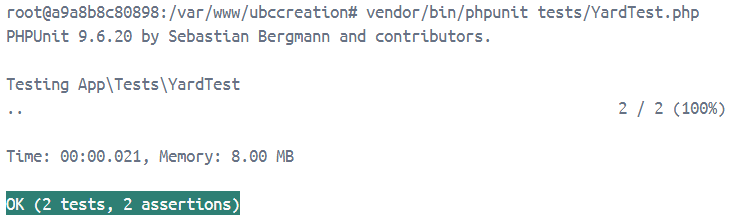
public function **setCity**(string $city): static

    {

        $this->city = $city;

        return $this;

}



Dans l’hypothèse où les spécifications fontionnelles du client doivent évoluer, un membre de l’équipe ou moi-même devons reprendre le code, potentiellement des mois voire des années après, et l’altérer pour répondre aux nouveaux besoins.

Le code relatif au setter, chargé de définir la valeur d’un champ d’une classe, voit dorénavant sa représentation chaînée converties en lettres capitales. Le test unitaire détecte l’erreur (i.e. une régression, car l’existant est brisé) et le signale dans le rapport.

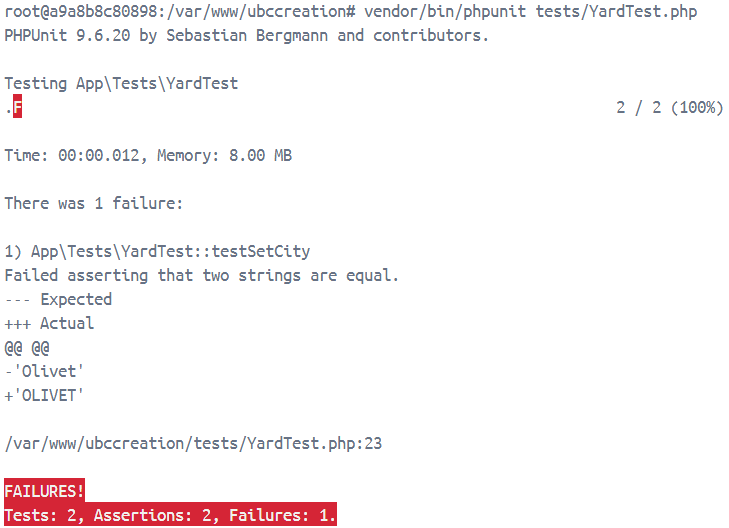
    public function **setCity**(string $city): static

    {

        $this->city = **strtoupper**($city);

        return $this;

    }



### Test fonctionnel

Le test fonctionnel est plus proche du scénario nominal effectué par l’utilisateur final. Il permet de contrôler la bonne exécution d’une action utilisateur, englobant parfois un pan fonctionnel complet.



L’instance de classe userRepository accède au méthodes d’interrogation de la base de données. Un mock (simulation d’une classe sans implémenter toutes ces caractéristiques et fonctionnalités) est créé pour un utilisateur.

La méthode loginUser() exécute programmatiquement le même comportement qu’un visiteur se connectant effectivement via le formulaire d’identification depuis le site web.

L’objet crawler, qui permet de traverser et extraire des informations depuis des documents XML ou HTML, interroge le serveur web dans les en-tête HTTP transmis par le navigateur et communique le chemin d’accès, via la méthode GET, pour rediriger vers la page intitulée ‘Index des chantiers’.

Cette même instance (*i.e.* self), affirme que le code de status HTTP est bel et bien 200 (*i.e.* la ressource requise est délivrée au navigateur web).

# Exploitation

## Containerisation avec Docker

### Pourquoi Docker ?

Docker permet d’empaqueter (*i.e.* packaging) une application avec toutes ses dépendances afin d’exécuter cette applicationdans n’importe quel environnement logiciel, la rendant portable entre centre de données,ordinateurs de bureau ou portables et l’infonuagique.

Docker brise les barrières entre équipes de développement et d’exploitation, assure le transfert de compétences d’un pôle à l’autre et accélère le déploiement de projets de génie logiciel tout en réduisant les phases de configuration.

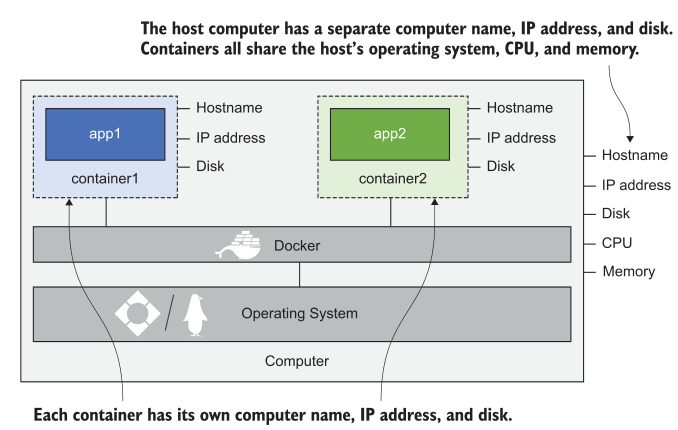
### Rôle et bénéfices

Docker est un conteneur virtuel léger : il partage le système le CPU, la mémoire vive et, contrairement aux outils de virtualisation, partage aussi le système d’exploitation de la machine hôte.

Cela offre l’avantage de réduire le parc informatique et diminuer le temps de déploiment grâce à une simple commande identique quelque soit le système d’exploitation, l’architecture du micro-processeur central ou la pile technologique constituant l’éco-système des équipes de développeur :

docker-compose –f ‘docker-compose.yml’ up –d --build

Cette commande construit (ou redémarre si existant) le conteneur depuis l’image définie par le manifeste docker-compose.yml en arrière-plan. Notez qu’un clic droit pour afficher le menu contextuel et l’option "Compose Up " sur le fichier .yml produit le même effet.

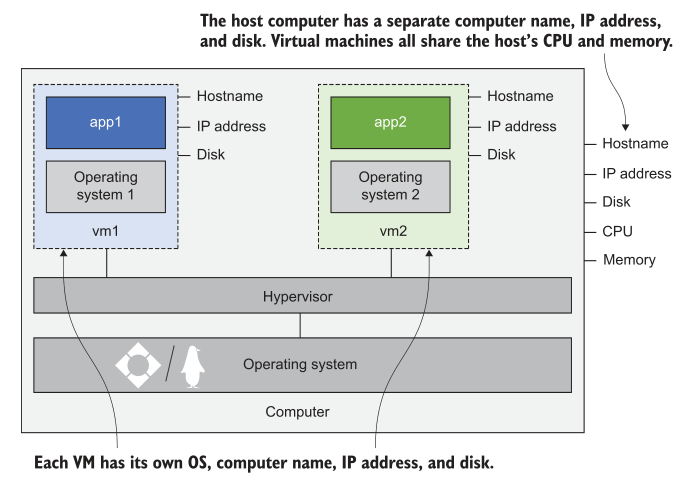


*Source : Learn Docker in a month of lunches, 2020, Stoneman*

*Figure 1.1* – Plusieurs conteneurs sur un ordinateur partagent l’OS, le CPU et la mémoire

Utiliser un outil de containeurisation est bien moins coûteux en ressources matérielles que les machines virtuelles et ne requièrent pas de formation associée à cette technologie complexe.

Le parc informatique de l’organisation s’en trouve réduit (ainsi que les factures énergétiques pour alimenter en électricité ce dernier) et la courbe d’apprentissage est accélérée, diminuant le nombre de jours-homme requis pour les nouveaux développeurs.



*Source : Learn Docker in a month of lunches, 2020, Stoneman*

*Figure 1.2* – Plusieurs machines virtuelles sur un ordinateur embarquent chacune leur propre OS

### Manifeste

Comme mentionné dans la rubrique précédente, le manifeste, a.k.a. le fichier à l’extension .yml (ou .yaml) situé à la racine du dossier décrit l’image Docker et ses services la composant.

Il décrit les mappages de port,les liens symboliques vers les systèmes de fichiers virtuels et (possiblement) les mots de passe.

##### Bonne pratique

Le mot clé depends\_on a été favorisé par rapport à link pour garantir l’ordre de construction des services et bénéficier d’un déploiment séquentiel logique.

En effet, le service web PHP n’est construit qu’après le service associé à la base de données et évite ainsi de possibles effets indésirables difficiles à détecter.

links:

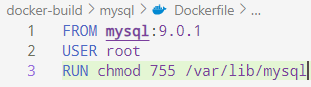
  - mysql-srv:db

Cette syntaxe aurait fonctionné en mappant le service mysql-srv à l’alias dans le système de fichiers local db, mais sans contrôle sur l’ordre effectif des constructions de service.

### DockerFile

Les fichiers DockerFile situés dans le sous-dosierr docker-build/ décrivent chacun des services à exécuter dans le conteneur. C’est pourquoi docker-compose (en sus de l’outil standard docker) est employé pour construire un conteneur constitué de plusieurs services.

Ces fichiers décrivent couche par couche l’installation et la configuration des paquets logiciels nécessaires à la bonne exécution de l’environnement logiciel sur lequel se déroulent les opérations associées aux services.

A ce motif, il est judicieux d’intervenir sur les dernières couches logicielles lors de l’ajout de nouveaux outils afin d’éviter la longue reconstruction d’un conteneur lors de nouveaux tests de déploiement.

Le DockerFile PHP peut stocker des variables d’environnement pour les rendre accessibles partout dans le système de fichiers du conteneur web.



## Système de gestion de contrôle de version

Contrôler la version du code source comme git, un outil répandu dans toute l’industrie du génie logiciel, présente divers avantages indéniables :

* Augmentation de l’employabilité : de nombreuses entreprises exigent la connaissance et la pratique régulière de git dans leurs offres d’emploi pour garantir la bonne intégration du candidat au sein des équipes et assurer l’itération des livrables dans des conditions optimum (sans heurt et dans les branches de développements et environnements spécifiquement définis par la culture technologique de l’organisation.
* Réduction des barrières entre les équipes : git chevauche les périmètres fonctionnels à la fois des équipes de développement et d’exploitation. Il facilite ainsi la transition entre les environnements logiciels et réduit les frictions les coeurs de métier.

Git assure également l’adoption d’un vocabulaire contrôlé commun entre ces deux parties prenantes pour communiquer de manière efficiente entre différents interlocuteurs, parfois délocalisés hors site (*e.g.*télétravail et internationalisation des projets).

* Accélération de la productivité des nouveaux participants: avantage proche de celui lié à l’employabilité, git permet de cloner un projet en une ligne de commande, quel que soit l’état du projet et l’architecture matérielle du nouvel intervenant. Cette prise en main immédiate assure l’utilisateur d’être aussi opérationnel grâce à l’acquisition aisée de l’instantané du projet.
* Facilitation du partage de projets open-source : le tissu économique open-source n’est pas nécessairement synonyme de gratuité et nombre d’organisations basent leur modèle d’affaires sur l’un des types de licence administrés par la corporation à but lucratif californienne Open source initiative (voir https://opensource.org/about).

C’est pourquoi de nombreuses compagnies font usage d’un VCS comme git pour porter leur projet au plus grand nombre, quel que soit leur localisation ou fuseau horaire.

* Faciliter la recherche des causes d’une régression : explorer l’arbre des soumissions de code (*i.e.* "commit") permet de (re)-découvrir quel label (*i.e.* "tag "), branche et/ou date a conduit à l’introduction d’un code brisant l’existant et introduisant ainsi une régression.
* Possibilité de *rollback :* recharger un état antérieur de l’applicatif grâce à la préservation des soumissions de code précédentes permet de recouvrer dans des conditions normales d’exploitation une situation où la base de code ne saurait être restaurée dans des délais jugés raisonnables.

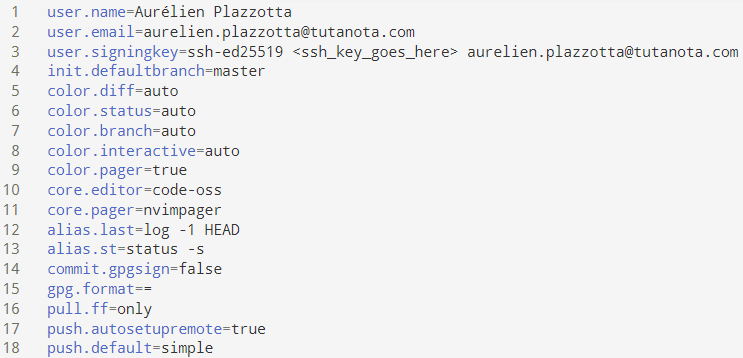
### Configuration de git

git peut être configuré de trois manières :

* système
* utilisateur
* local.

avec une précédence croissante au fur et à mesure de l’accroissement de la spécificité du périmètre fonctionnel (*i.e.* "system-wide" à local).

Voici le contenu du fichier de configuration git utilisé pour la réalisation du projet :



*Figure 3.1* –Fichier de configuration global git

### Lexikon des commandes git

Une fraction des commandes git est utilisée durant le développement, intégration et déploiement du projet de site web ; en voici un résumé :git

##### Configuration et gestion du dépôt

|  |  |
| --- | --- |
| Commande | Description |
| git config--global <property> <value> | Définit un nouveau paramètre de configuration global |
| git add remote origin git@github.com:<user>/<repository>.git | Actualise l’index de l’espace de travail pour pister de nouveaux fichiers ou intégrer les modifications dans la zone de tampon intermédiaire avant future soumission |
| git branch –M master | Détermine le nom de la branche principale |
| git remote set-url origin git@github.com:<user>/<repo\_name>.git | Bascule du protocole HTTP vers SSH pour téléverser les itérations (*i.e.* "push") |
| git clone [--recursive, --depth=1] <URL> | Télécharge le dépô, et optionnellement ses dépendances et la dernière entrée dans l’arbre des "commit" |
| git init | Initialise un dépôt git en ajoutant le dossier .git nécessaire au fonctionnement interne du VCS (i.e. Version Control System) |

##### Développement et livraison

|  |  |
| --- | --- |
| Commande | Description |
| git commit –mS <message> | Soumet un jeu de modifications de fichiers dans l’arbre des itérations de la branche. Chiffre ce commit avec une clé publique en usant du protocole PGP. |
| git push –u origin <branch> | Téléverse les modifications depuis le dépôt local vers le dépôt distant. |
| git switch –c <branch> | Crée une branche et fait pointer le curseur logique HEAD pour basculer immédiatement dessus. |
| git branch –M master | Détermine le nom de la branche principale |
| git branch –a | Lister toutes les branches existantes |
| git branch (–d/-D) <branch> | Supprime une branche. Outrepasse la confirmation. |
| git log -3 --oneline --decorate | Afficher les 3 dernières entrées (*i.e.* "commit") en une ligne |
| git remote set-url origin git@github.com:<user>/<repo\_name>.git | Bascule du protocole HTTP vers SSH pour téléverser les itérations (*i.e.* "push") |
| git merge <branch> | Fusionne la branche cible avec la branche actuelle |
| git –set-upstream origin <branch> | Définit une nouvelle référence virtuelle pour maintenir une nouvelle branche distante depuis une branche locale |

##### Résolution des problèmes

|  |  |
| --- | --- |
| Commande | Description |
| git pull [--no-rebase] | Rappatrie les modifications du dépôt distant vers le dépôt local. Intègre les changements lorsqu’une divergence de l’historique survient. |
| git reset –hard origin/master  git clean –d --force | Résoud la situation lorsque la branche est dans un état HEAD détaché (utile lors du pointage vers une nouvelle branche après une suppression logique de la référence). |
| git checkout --(<filename>/.) | Annule les changements effectués sur un fichier pour recouvrer l’état de l’ancient "commit". Annule TOUS les changements effectués |
| git log -<n> --pretty=oneline  git reset –hard <hash\_id>  git switch <branch> | Recouvrer l’état de l’ancien "commit" avec destruction physique des données |
| git restore –staged . | Révoquer le dernier git add sans perte de données |

### Editeur de texte et extensions

Visual Studio Code de Microsoft est utilisé pour la réalisation du projet. De multiples extensions ont été installées pour bénéficier de fonctionnalités comme l’auto-complétion, la documentation ou la coloration syntaxique, icônes graphiques, thème ou micro-serveur HTTP embarqué pour le rechargement automatique du rendu HTML.

### Migration de la base de données

Après avoir implémenté le modèle de données, une migration vers la base de données permet d’insérer les nouvelles tables, leurs colonnes et leurs enregistrements. Dans le terminal, deux commandes sont exécutées directement dans le service associé au serveur web du conteneur. La première écrit le code SQL nécessaire à la dusdite migration et la deuxième exécute le code pour effectuer la persistence des données :

php bin/console make:migration

php bin/console doctrine:migrations:migrate

En voici un extrait depuis le dossier migrations :

public function **up**(**Schema** $schema): void

{

    $this->**addSql**('ALTER TABLE yard ADD CONSTRAINT FK\_29B02F28DE9C79B3 FOREIGN KEY (type\_site\_id) REFERENCES type\_site (id)');

    $this->**addSql**('CREATE INDEX IDX\_29B02F28DE9C79B3 ON yard (type\_site\_id)');

    }

public function **down**(**Schema** $schema): void

{

    $this->**addSql**('ALTER TABLE yard DROP FOREIGN KEY FK\_29B02F28DE9C79B3');

    $this->**addSql**('DROP INDEX IDX\_29B02F28DE9C79B3 ON yard');

}

*Figure 1.3* – Ajout d’une clé étrangère à la table existante yard

La fonction up() exécute la logique métier pour persistence dans la base de données tandis que la fonction down() annule les changements effectués pour recouvrer une version (ou instantané) précédent de la base.

Ce retour en arrière (i.e. "roll-back") de la base s’effectue en appelant un argument optional dans la ligne de commande associée à l’outil PDO (pour interfaçage entre la couche métier et la base) Doctrine :

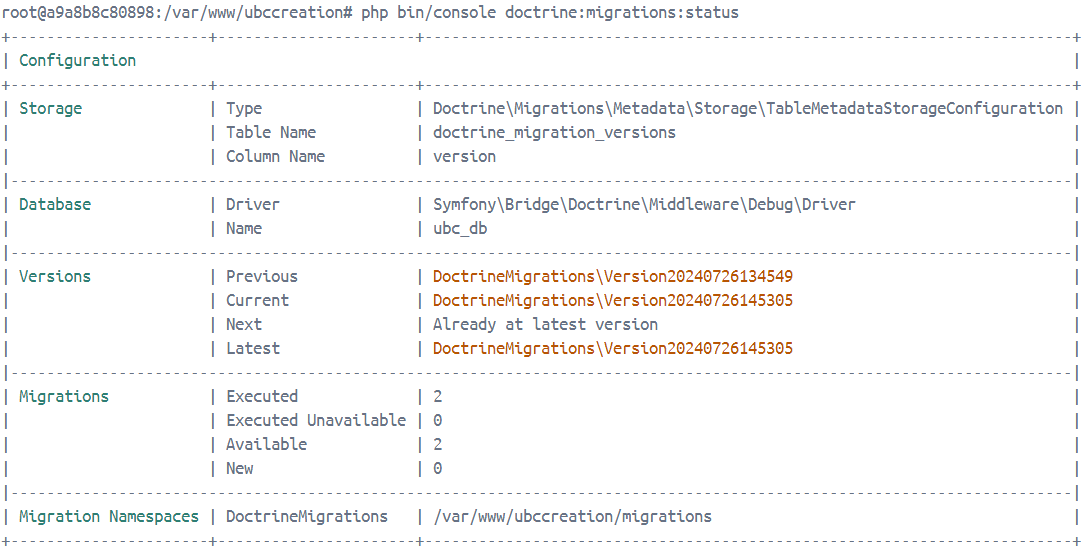
php bin/console doctrine:migrations:migrate prev

Il est possible de recouvrer de multiples changements en une passe à l’aide de l’argument -n. Un autre argument existe pour simuler les changements (appliqués seulement virtuellement) afin d’assurer une maintenance de la base de données avec succès. L’example suivant effectue une simulation d’annulation des deux dernières migrations:

php bin/console doctrine:migrations:migrate [--dry-run] –n -2

Une autre commande très utile permet de diagnostiquer rapidement l’état de l’art grâce à un récapitulatif des dernières opérations réalisées.

php bin/console doctrine:migrations:migrations:status



*Figure 1.3* – Status de la base de données

### Sauvegarde de la base de données :

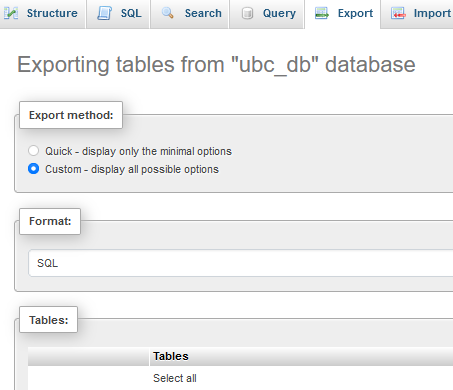
#### Contexte

Dans une civilisation entièrement numérisée où les données sont roi, la base de données stocke la quasi-intégralité du système du système d’informations lié à l’informatique de gestion, d’une organisation et incarne *in fine*, un actif capital pour la continuité d’activité des opérations quotidienne.

La parte d’une ou plusieurs base(s) représente est une catastrophe et doit donc être évitée à tout prix. Dans une grosse structure, un admnistrateur de base de données est dédié à cette activité, tel le maintien de sauvegardes régulières, mais dans des petites entités, le développeur web a très possiblement la responsabilité de cette tâche.

#### PhpMyAdmin - export

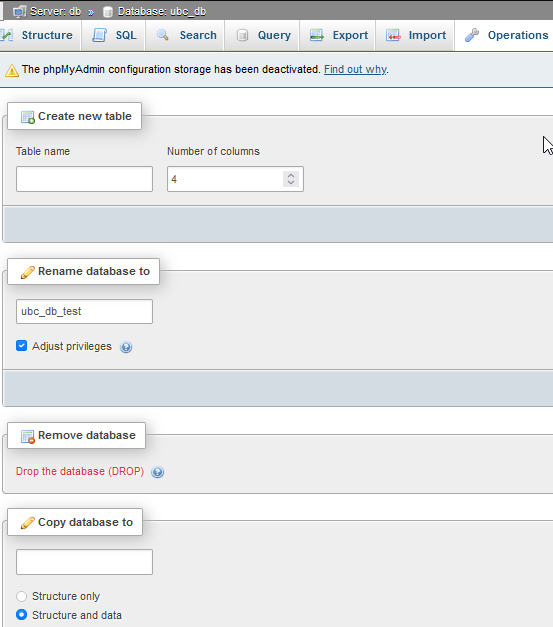
Le panneau d’administration PhpMyAdmin accessible via le service éponyme dans le conteneur Docker inclut un outil simple d’exportation de base de données accessible dans l’onglet *Export*.



*Figure 1.3* – Outil d’exportation de base de données

#### PhpMyAdmin - Operations

Bien que son rôle soit différent d’une exportation en vue d’un stockage long terme et/ou une importation ultérieure en case de défaillance de la base, le panneau d’administration PhpMyAdmin inclut également un outil de copie de base de données.



*Figure 1.4* – Outil de copie de base de données

#### mysqldump

Outil de sauvegarde en ligne de commande, très utile pour inclusion dans un script afin d’automatiser le process en l’associant avec un agent ou un daemon, tel cron ou snooze . Voici le script shell exécuté de manière bi-hebdomadaire pour sauvegarder les deux bases de données (production et tests) :

maintenance/db\_backup.sh

*#!/usr/bin/env elv*

*# Run every Wednesday and Saturday at 03:00 a.m*

*# Halt immediately when encountering a failure.*

**set** -e

*# Set agent every wednesday and saturday at 03:00 a.m.*

**snooze** –w 3,6 –H3

**cd** $(**dirname** "$0") || **exit**;

if [[ ! -d './automata/log ']]; then

**mkdir** -p './automata/log';

fi

currentLogAmount='$(ls ./automata/log | wc -l)'

*# Update signature's database.*

**freshclam** --quiet

**clamscan** -riz /var/www/ubccreation

newLogAmount='$(ls ./automata/log | wc -l)'

if [[ '$($?) == 0']]; then

**mysqldump** -u root -p ubc\_db > backup/ubc\_db-$(**date** +%d%m%Y-%H:%m).sql

**mysqldump** -u root -p ubc\_db > ubc\_db-$(**date** +%Y%m%d-%H:%m).sql

**notify-send** --urgency-low --expire-time=8000 --category='maintenance' "Having performed databases\’ backup with success."

*# An absence of a new log file signals some virii detection.*

   if [[ newLogAmount != currentLogAmount ]]; then

**notify-send** --urgency-normal --expire-time=12000 --category='maintenance' 'Some virii have been found in the partition.'

**exit** 1

   fi

else

**notify-send** --urgency-critical --expire-time=20000 --category='maintenance' "An error has occured during execution's maintenance task!"

**exit** 2

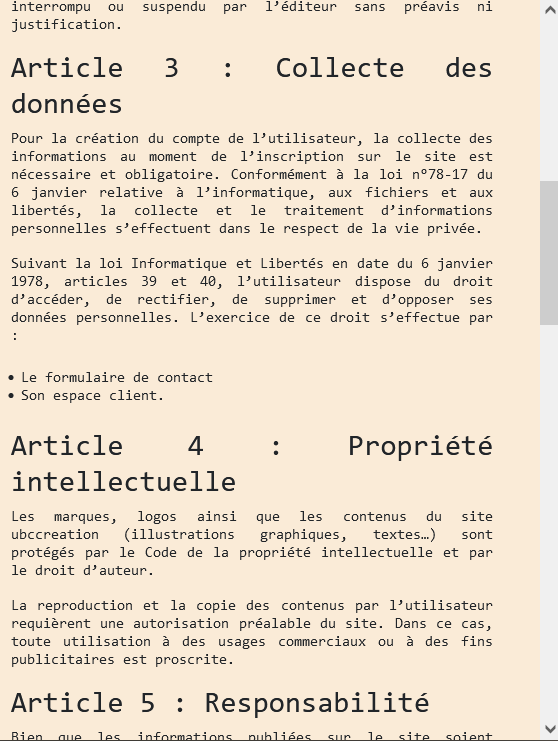
fi

Le développement du projet logiciel s’est effectué à l’aide de l’outil de mise en conteneurs Docker, présentée succintement dans la rubrique **Containerisation avec Docker** du présent dossier. Son usage a facilité nettement le déploiement et sa portabilité entre les deux ordinateurs utilisés durant le cycle de développement et garantit l’homogénéité des environnements.

Deux étapes sont nécessaires pour son implémentation : l’écriture du fichier docker-compose.yml et des DockerFile.

* docker-compose.yml : manifeste au format YAML (peut être écrit simplement .yml) contenant la description de chacun des services à éxécuter dans un conteneur
* DockerFile : écrit à la fois pour le service PHP et le service MySQL, ce fichier intègre les différentes couches logiques constituées des dépendances logicielles nécessaires à la bonne exécution du conteneur.

## Législation

Respectueux des législations en vigueur à la fois sur le territoire français et au sein de l’union européenne, le site web ubccration.fr intègre des mentions légales relatives à l’immatriculation de l’entreprise au Registre du Commerce et des Sociétés auprès de la Chambre du Commerce et de l’Industrie.

Des conditions générales d’utilisation sont également ajoutées pour se conformer au Règlement Général sur la Protection des Données (*viz.* RGPD) assurant le consentement explicite de l’utilisateur concernant la gestion des données personnelles suceptibles d’être stockées durant la navigation, notamment les cookies.

### RGPD

Le RGPD impose une obligation de moyens visant à sécuriser les données collectées et utilisées relatives aux utilisateurs.

Deux types de données sont concernées par ces dispositions législatives :

* Données personnelles : elles permettent l’identification d’une personne physique.
* Données sensibles : elles font l’objet d’une potentielle discrimination, basée sur la religion, le sexe ou l’*ethnè*.

Si les cookies du siteubccreation.fr ne furent employés que pour que pour effectuer une connexion sur le compte de l’utilisateur, aucun moyen de protection n’eut été nécessaire.

Néanmoins, à l’heure de la date d’édition du présent dossier, une adresse électronique de l’utilisateur. est collectée à des fins de démarchage commercial. C’est pourquoi le consentement explicite de ce dernier est nécessaire et requiert une fenêtre informative avertissant de la collecte de données.

Ce consentement est alors enregistré grâce à la variable globale $\_COOKIES et fait l’objet d’une durée d’expiration de deux ans si aucune action de l’utilisateur n’est effectuée pour renouveller les échanges commerciaux avec l’entité UBC Créationdurant ce délai.

Bien que Symfony gère automatiquement les cookies, sa syntaxe en vanilla PHP pour générer un nouveau cookie est la suivante :

**setcookie**($user->**getId**(), **crsfToken**(**randomHash**()), **time**() + 63072000);

Controller la présence d’un cookie, *ditto* existant et dont la valeur est différente de null :

if (**isset**($\_COOKIE[$user->**getId**()])) { }

## Conclusion

Dû à la dernière période de stage (soit un tiers de la durée assignée globale) très peu profitable en raison d’une absence totale de connectivité internet empêchant d’accéder au dépôt github et aux documentations en ligne, un manque de temps flagrant s’est fait sentir eu égard à la réalisation d’objectifs préalablement fixés.

Le système de messagerie et le système d’actualités qui représentent des objectifs additionnels n’ont pu être finalisés.

De plus, les scenarii "Bannir un membre " et "Demander le mot de passe oublié" (à l’aide de l’outil open source PHPMailer) n’ont pas été implémenté.

Voici les différentes exploiratoires jugées pertinentes pour la continuité et amélioration du projet présenté :

* Remplacer la base de code JavaScript par TypeScript pour bénéficier du système de type afin de capturer plus d’erreurs grammaticales en amont (avant l’exécution).
* Ecrire des tests de bout-en-bout en TypeScript grâce à l’exécuteur Cyfra.
* Ecrire des personae pour affiner la connaissance des prospects.
* Ajouter un système de statistiques en PHP (compteur de visites) avec une table supplémentaire en SQL.
* Intégrer un diagramme de Gantt pour suivre au plus près la consommation des ressources en "jours/homme " et ajuster au fil de l’eau les priorités selon les échéances imminentes.
* Pipeline CD/CI : implémenter un "gazoduc" développement continu / intégration continue pour normaliser et industrialiser la livraison des itérations logicielles via leur sujétion à une batterie de tests unitaires et de cas d’utilisation exécutés automatiquement et sur plusieurs systèmes d’exploitation.
* L’emploi de l’outil de gestion des tâches sur trello.com pour assurer une communication avec les dirigeants et poser la base d’une gestion de projet formalisée.
* Une redirection personnalisée des codes de status HTTP 400, 401, 403, 404 et 500 depuis le fichier .htaccess et l’édition du fichier de configuration /etc/apache2/sites-available/000-default.conf
* Configurer les extensions PHP XDebug et OpCache dans le conteneur. Ces extensions favorisant le débuggage et l’améliorations de performances n’ont pu être effectuées que sur mon ordinateur personnel à titre expérimental, par faute de temps.

### Outils informatiques utilisés

#### Bases de données

|  |  |  |
| --- | --- | --- |
| Nom | Type | Version |
| PHPMyAdmin | panneau d’administration d’une base de données MySQL | N/A |
| MySQL | base de données SQL | 9.0.1 |

#### Conception logiquelle

|  |  |  |
| --- | --- | --- |
| Nom | Type | Version |
| Designer | outil en notation graphique de conception de modèles de données intégré à PhPMyAdmin | N/A |
| drawio.com | outil de conception en notation graphique pour produire des diagrammes SysML (ou UML) | N/A |
| figma.com | outil de conception d’interface collaboratif | N/A |
| mocodo.net | outil de conception en notation textuelle pour produire des MCD en notation graphique | N/A |
| SysML | langage de de modélisation systèmes | 1.7 |

#### Développement

|  |  |  |
| --- | --- | --- |
| Nom | Type | Version |
| Angular / Angular CLI | cadriciel JavaScript orienté applications web. Inclut son outil en ligne de commande. | 18.1.2 |
| Helix | Editeur de texte modal, successeur de neo-vim | 24.07 |
| Symfony | cadriciel PHP orienté applications complexes | 7.1 |
| Visual Studio Code | éditeur de texte | 1.91.1 |

#### Exploitation (viz. Ops)

|  |  |  |
| --- | --- | --- |
| Nom | Type | Version |
| Docker | environnement d’exécution autonome pour conteneurs | 20.10.25 |
| docker-compose | outil de définition et exécution d’applications multi-conteneurs | 2.27.0 |
| Docker Desktop | logiciel de construction, partage et exécution d’applications et microservices en conteneurs | 4.33 |
| git | système de contrôle de version du code source | 2.45.2 |
| github.com | plateforme CI/CD, management de projet, VCS | N/A |
| phpunit | outil de tests PHP | 9.6.20 |

#### Front-end

|  |  |  |
| --- | --- | --- |
| Nom | Type | Version |
| Bootstrap | cadriciel CSS | 5.3.3 |
| sass | pré-processeur CSS | 0.7.0 |
| Twig | moteur de squelettes de modèles | 3.10.3 |

#### Gestion de projet

|  |  |  |
| --- | --- | --- |
| Nom | Type | Version |
| trello.com | outil en ligne gestion de projet collaboratif | N/A |

#### Langages

|  |  |  |
| --- | --- | --- |
| Nom | Type | Version |
| ARIA | sous-spécification du HTML permettant de définir des comportements spécialisés orientés accessibilité des mal-voyants et liseuses numériques | 1.3 *(draft)* |
| HTML | langage de présentation des données | 5.3 |
| JSON | langage de formation d’échange de données léger | ECMA-404 |
| PHP | langage back-end | 8.3 |
| CSS / SCSS | langage de structuration des données et sa contre-partie doté d’un pré-processeur CSS | NA**\***/ 3.6.6 |
| SQL | langage de manipulation et interrogation de bases de données | ISO/IEC 9075-1 :2023 |
| YAML | langage de sérialization de données lisible par un humain | 1.2 |

***\*****: le groupe de travail CSS du World Wide Web consortium est dorénavant scindé en plusieurs sous-groupes implémentant séparément leurs propres spécifications techniques, aux dates de publication et numéro de version distincts (e.g. Text Level 4, Media Queries level 3, Scroll Snap level 2, etc.).*

#### Systèmes

|  |  |  |
| --- | --- | --- |
| Nom | Type | Version |
| bash | Emulateur de terminal par défaut sur de multiples distributions GNU/Linux | 5.2.21 |
| Debian unstable | Système d’exploitation GNU/Linux | linux-image 6.1.10 |
| elvish | Emulateur de terminal et langage de script écrit en Go | 0.20.1 |
| VoidLinux | Système d’exploitation GNU/Linux | linux-image 6.6.42 |
| Window Subsystem Linux | Couche de compatibilité pour l’exécution native d’exécutables binaires Linux au format ELF | 2.2.4.0 |
| Windows 10 Pro | Système d’exploitation développé par Microsoft | 22H2 |

### Bibliographie

#### Git

* Pro Git de Scott Chacon, 2014, mis à jour en ligne sur git-scm.com/book/en/v2

#### PHP

* php.net
* *PHP & MySQL web :Server-side Web Developement* de Jon Duckett, 2022
* symfony.com/doc/current/reference/
* symfonycasts.com/screencast/
* discord.com – canal de discussion PHP DEVELOPERS
* phpandmysql.com

#### SQL

* stackoverflow.com
* dev.mysql.com/doc/refman/9.0/en

#### SysML

* SysML Distilled : A brief guide to the systems modeling Language de Lenny Delligatti, 2013
* A practical guide to SysML: The systems modeling language, 3rd edition, 2014
* drawio.net

## License

Les contenu du présent document sont sujets à la license ISC, dont voici les termes :

Copyright (c) 2024 Aurélien Plazzotta [aurelien.plazzotta@tutanota.com](mailto:aurelien.plazzotta@tutanota.com)

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

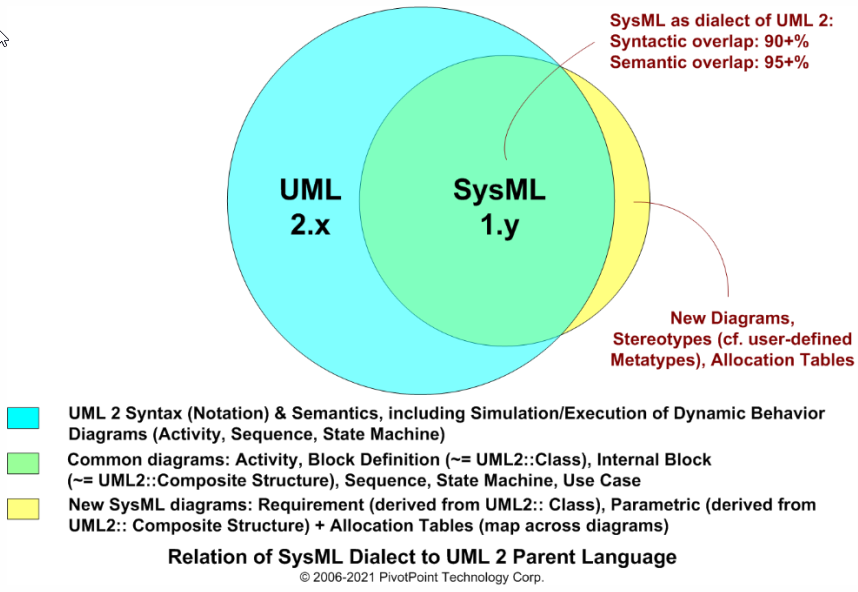
# Annexes

#### Location du nom de domaine et location du serveur web pour héberger les données du site :

* Migration entrante gratuite
* Créer des comptes N0C isolés, redondants et illimités
* Caching Pro inclus et accélération LSCache native, activée par défaut
* Litespeed Entreprise offert sur la version Pro
* Bande passante illimitée et trafic jusqu'à 60 000 visiteurs par mois
* WAF Pro inclus et protection Anti-Malware
* Réseau Internet premium AS53589 avec protection anti-DDOS
* Latence optimale (connecté à France IX, Equinix IX, QIX et TorIX)
* IPextension - Plus de 1000 IP supplémentaires (SEO)
* Plusieurs versions PHP, Python, Node.js, Ruby
* Installateur de CMS en un clic
* Outil de création de site web inclus
* Nombre de bases de données illimitées
* Nombre de boîtes mail illimité et filtre anti-spam N0C Pro inclus
* 1 nom de domaine inclus et bureau d'enregistrement accrédité par l'ICANN (.fr, .ca, .be, .ch, .us, .uk
* Certificats SSL dédiés et wildcard Let's Encrypt offert pour chaque nom de domaine
* Partition externe N0C Storage extensible (100 GB gratuits)
* Sauvegardes quotidiennes (jusqu'à 20 jours) vers un 2e centre de données
* Datacenters Tier III et Tier IV en France, Suisse et Canada
* Garantie satisfaction de 60 jours ou remboursement
* Support 24/7 et activation instantanée
* Panneau de contrôle N0C en marque blanche (revendeur)
* Haute disponibilité (99.99% en moyenne)

*Figure 1* - Plaquette descriptive de l’offre d’hébergement de planethoster

#### Comparatif holistique des languages UML et SysML



*Figure 2 – Relation entre le dialecte SysML et son langage parent UML*

#### Extrait du dépôt github hébergeant le code source du projet

