

## Introduction

Ce projet a pour objectif de vous faire réaliser une application web de gestion de copropriété. Elle devra permettre à une copropriété autogérée (sans syndic) de gérer en toute transparence les dépenses communes liées à la copropriété, de débattre de ces sujets, de rassembler les fonds nécessaires au paiement des charges et de centraliser tous les documents (contrats, devis, factures, autres) liés à la gestion de la copropriété.

Dans le cas où vous n'auriez pas le temps de réaliser toutes les fonctionnalités demandées, certaines fonctionnalités seront marquées comme prioritaires avec un marqueur « \* » dans le titre. Cela vous aidera à choisir par quoi commencer, sachant que s'il manque une de ces fonctionnalités prioritaires, vous serez plus pénalisés sur la notation que s'il manque des tâches non prioritaires.

En travaillant régulièrement chaque semaine, vous devriez pouvoir tout réaliser (tâches non prioritaires comprises) ce qui vous permettra d'obtenir des points bonus et de maximiser vos chances d'obtenir une très bonne note de projet. En effet la réalisation des tâches optionnelles peut vous apporter des points bonus jusqu'à dépasser le barème de 20.

Les fonctionnalités sont présentées dans un ordre qui paraît le plus naturel pour exposer le projet mais il ne reflète pas forcément l'ordre exact dans lequel vous devez implémenter ces fonctionnalités. Vous êtes donc libres de commencer par ce que vous voulez, en gardant en tête que certaines fonctionnalités sont prioritaires.

## Regroupement en binômes

Si vous le souhaitez, vous pouvez vous regrouper en binôme pour vous répartir les tâches. Pensez bien une fois que vous avez formé votre binôme à m'envoyer par email avec qui vous allez travailler et sur quel dépôt github vous allez envoyer vos modifications.

## Organisation personnelle

N'attendez pas la dernière minute pour commencer, au risque de vous retrouver à court de temps pour rendre un projet finalisé. Commencez dès maintenant à travailler sur les concepts que vous maîtrisez, puis chaque jour, nous verrons de nouvelles facettes du framework qui vous permettront de raffiner votre code pour se rapprocher du besoin final.

Par exemple, en attendant d'avoir vu le cours sur Doctrine, vous pouvez commencer à préparer les routes, controllers et templates initiaux sans gérer la persistance des données. Idem pour l'aspect sécurité, il est possible de s'en abstraire tant que l'on a pas vu comment s'y prendre. Si vous voulez prendre de l'avance, vous pouvez également rechercher sur la documentation officielle comment gérer tel ou tel problème (comme la sécurité et la gestion des utilisateurs par exemple) pour en mettre une version simple en œuvre.

## 1) Page d'identification \*

Toutes les pages de l'application sont sécurisées et nécessitent une authentification. Pour simplifier cette partie, la création des comptes utilisateurs pourra être faite à la main dans un premier temps, selon la méthode de votre choix (saisie manuelle en BDD ou utilisation du `user provider in_memory` par exemple).

La tâche 11 (non prioritaire) est complémentaire à celle-ci puisqu'elle implique la création d'une interface de gestion des utilisateurs.

## 2) Communications entre les copropriétaires \*

Il ne s'agit là ni plus ni moins que d'une sorte de forum simplifié. Chaque copropriétaire peut créer un message à destination de tous comme par exemple « Attention, samedi 12 nous allons réaliser des travaux dans notre jardin. Désolé d'avance pour le dérangement » ou encore « Quelqu'un a-t-il un fer à souder à me prêter ? ».

Il est également possible de s'adresser à un ou plusieurs copropriétaires en particulier lorsque le message ne concerne pas tout le monde. Dans ce cas, le message ne sera visible que par les destinataires précisés lors de la rédaction du message.

Le but étant de communiquer, il est évidemment possible de répondre aux messages. Chaque création de message et chaque réponse envoie une notification aux copropriétaires concernés.

Pour simplifier l'implémentation, vous pouvez choisir de gérer la liste des messages sous forme de liste non arborescente, libre à vous.

Enfin, l'administrateur peut décider d'archiver un fil de messages.

## 3) Gestion de la liste des charges \*

Une copropriété doit faire face à un certain nombre de charges. Dans le cadre de ce projet, on considèrera qu'il n'existe pas de charges récurrentes (type paiement mensuel de facture d'électricité), mais uniquement des charges ponctuelles. Certaines charges concernent tout le monde, tandis que d'autres concernent uniquement certains propriétaires.

L'interface permettra donc de créer des charges constituées de :

- un titre
- un montant
- une date d'échéance
- un statut : payé / à payer
- une liste de propriétaires concernés (tous par défaut)
- une pièce jointe optionnelle (facture)
- un contrat optionnel (lien vers une entité contrat)

Pour chaque charge on considèrera que son coût est systématiquement réparti de manière proportionnelle entre tous les propriétaires concernés.

Il devra être possible de consulter pour chaque charge, la liste des versements effectués et restant à effectuer.

**Attention :** Lorsqu'il est mentionné optionnel sur une propriété d'une entité, comme les charges ci-dessus, cela ne signifie pas que l'implémentation est optionnelle, mais que l'utilisateur final n'est pas obligé de remplir la propriété pour

## 4) Gestion de la liste des contrats en cours

Chaque signature de contrat devra être consignée dans l'application en envoyant le scan du contrat en question. Un contrat est composé de :

- un nom
- une date de signature
- une date de fin optionnelle

## 5) Création d'un projet d'évolution \*

Pour soulever un problème et demander une prise de décision à ce sujet, il doit être possible de soumettre un projet. Le projet expose la situation et ouvre la discussion pour que chacun puisse donner son avis.

Un fil de discussion est donc attaché à chaque projet, mais il peut rester vide. En effet, certains projets ne nécessitent pas de longue discussion mais uniquement un vote.

Il doit donc être possible d'attacher un nombre arbitraire de questions à choix multiples (pas de questions ouvertes, ces dernières sont gérées dans le fil de discussion) à un projet.

Le type de question auxquelles un projet pourrait être soumis seraient par exemple :

- « Devrait on acheter des bacs à poubelle ou opter pour un contrat de location ? » (oui / non)
- « Suite aux devis réalisés, quel contrat d'entretien du portail choisissons nous ? » (contrat 1 / contrat 2 / contrat 3)

Un projet est constitué des éléments suivants :

- Un nom
- Une description
- Un statut (« en discussion », « en attente d'exécution », « exécuté »)
- Une date d'ouverture
- Une date de clôture (optionnelle / nullable)
- Un fil de discussion
- Une liste de sondages (0 à n sondages pour voter sur 0 à n sujets)
- Une liste de pièces jointes (devis, propositions de contrats, etc.)
- Un propriétaire (celui qui crée le projet)
- Une liste de personnes concernées (tout le monde n'est pas forcément concerné par tous les projets)

Lorsqu'un projet est créé, les utilisateurs concernés reçoivent une notification par email. Ils ne peuvent pas modifier le projet mais ils peuvent à tout moment ajouter une pièce jointe (nouveau devis par exemple) ou un message dans le fil de discussion lié. Cela permet alors à tout le monde de faire faire des devis complémentaires pour les ajouter au dossier.

Le propriétaire du projet peut quant à lui modifier le projet, pour mettre à jour la description, le statut ou encore soumettre un nouveau questionnaire.

Chaque modification du projet envoie une notification par email aux utilisateurs concernés.

### Sondage

Quand on parle de liste de sondages dans le bloc ci-dessus, on parle en réalité d'une entité responsable de gérer une et unique question à choix multiple ainsi que les réponses associées. On pourrait gérer des groupements de questions, mais pour simplifier, chaque élément correspondra à une seule question.

### Journalisation d'une activité

Il devra être possible pour n'importe lequel des copropriétaires concerné par le projet de journaliser une activité liée au projet. Cette entrée de journal comprendra les éléments suivants :

- Un titre (« appel de la société X pour obtenir un devis. »)

- Une note optionnelle (« le devis devrait être envoyé d'ici la fin de la semaine » )
- Une date
- Un rédacteur / propriétaire (l'utilisateur qui a saisi la note)

La liste des notes attachées à un projet sera alors visible directement sur la fiche projet.

## 6) Appel de fonds

Lorsque les votes à un questionnaire doivent donner lieu à un paiement (ex : il est décidé de commander une prestation de réparation du portail à une entreprise donnée), on doit pouvoir réaliser un appel de fonds.

Il s'agit alors de récolter les fonds sur le compte bancaire de la copropriété pour pouvoir payer le prestataire.

Un appel de fonds porte forcément sur une charge. La création d'un appel de fond envoie un email à l'ensemble des copropriétaires concernés par la charge.

## 7) Gestion des versements \*

Lorsqu'un appel de fonds a été réalisé, les copropriétaires envoient leurs paiements respectifs par virement ou par chèque à un des copropriétaires responsable de gérer le dépôt en banque. La personne responsable de déposer les chèques doit alors pouvoir saisir un versement dans l'interface. Un versement est constitué des éléments suivants :

- Un propriétaire (celui qui paye)
- Un montant
- Une date
- Une ou plusieurs pièces jointes optionnelles (scan du chèque, scan du bordereau de remise de chèques, autre preuve si nécessaire)
- Un type (virement bancaire / chèque)
- Une charge liée

Il n'y a pas d'obligation que le paiement corresponde à la part du propriétaire pour la charge liée au versement. En effet, un propriétaire est libre de s'acquitter de sa part en plusieurs fois.

Par ailleurs, on notera qu'un versement ne fait pas obligatoirement suite à un appel de fonds. Il peut être spontané, suite à une réunion informelle par exemple, mais il devra forcément être lié à une charge.

## 8) Gestion des réunions

Il devra être possible d'organiser des réunions entre les copropriétaires. L'organisation consistera à soumettre aux copropriétaires un choix de date et d'heure puis à afficher la date sélectionnée sur la page d'accueil en tant que « Réunion X planifiée le DATE à HEURE chez M. Y ».

Une fois la date passée, il sera possible de joindre un compte rendu de réunion.

## 9) Page d'accueil \*

La page d'accueil est le point d'entrée de l'application. Elle devra contenir :

- un lien vers chacun des fils de discussion actifs,
- la liste des projets en cours,

- La liste des charges restant à régler,
- La liste des versements restant à effectuer pour l'utilisateur courant.

## 10) Amélioration des notifications par email

Pour éviter de recevoir un message pour chaque modification au sein de l'application, il serait intéressant de ne plus envoyer ces emails systématiquement mais de les stocker pour envoyer une synthèse en fin de journée. La durée de stockage ainsi que l'heure d'envoi de la synthèse sont alors stockés dans des variables de configuration, et une tâche planifiée se charge d'envoyer la synthèse à l'heure requise.

## 11) Interface d'administration et gestion des rôles

Une interface d'administration doit permettre à l'administrateur de créer des utilisateurs, de leur affecter un rôle (parmi `ROLE_USER` et `ROLE_MANAGER`) et de les impersonnifier.

On ajoute un rôle `ROLE_ADMIN` ainsi qu'un rôle `ROLE_MANAGER` qui correspond à la personne qui centralise les paiements.

Le `ROLE_MANAGER` est le seul à pouvoir saisir des charges, des contrats ou des appels de fonds.

Le `ROLE_ADMIN` quant à lui n'a accès qu'à la liste des utilisateurs.

Le `ROLE_USER` a accès à la liste des fils de discussion, projets et charges auxquels il est associé ainsi qu'à la page d'accueil ou il peut consulter les versements qu'il lui reste à effectuer pour s'acquitter de ses charges.

L'administrateur qui impersonnifie un utilisateur (`ROLE_PREVIOUS_ADMIN`) a des droits spéciaux comme l'archivage des fils de discussion, ainsi que tous les droits alloués au `ROLE_MANAGER`.

Enfin, la création d'un utilisateur doit envoyer un email à celui-ci (chaque utilisateur étant identifié par une adresse email) comprenant un lien lui permettant d'initialiser son mot de passe. Le compte ne doit pas être utilisable tant que ce mot de passe ne sera pas initialisé.

## 12) Gestion de la qualité\*

### Tests de performance

Cette application étant destinée à être utilisée par un nombre limité d'utilisateurs simultanés, faire des tests de charge poussés pour tester le comportement de l'application sous des centaines d'utilisateurs simultanés serait de la surqualité. Il ne sera donc pas nécessaire de réaliser ce genre de tests dans le cadre de ce projet.

En revanche, cela ne vous dispense pas d'analyser la performance de votre application de bout en bout et de prendre les mesures nécessaires pour l'optimiser.

Dans votre rapport, vous préciserez les mesures que vous avez prises pour analyser les performances de votre application web, les anomalies rencontrées, les correctifs apportés ainsi que les gains de performance après correction. A ce titre, vous n'oublierez pas de décrire votre mode opératoire et de fournir les captures d'écran correspondantes.

Vous détaillerez également, pour les cas d'usage que vous jugerez les plus importants au sein de l'application, la distribution du temps passé entre les différentes couches entre l'émission de la requête et l'affichage du résultat final pour l'utilisateur.

## Tests unitaires et fonctionnels

Le choix des portions de code à tester unitairement et des scénarios à tester fonctionnellement est laissé à votre libre appréciation.

Vous pouvez donc choisir de tester certaines portions unitairement et d'autres portions fonctionnellement, de garder certains tests manuels, ou de tout automatiser, mais dans tous les cas, vous justifierez ces choix dans votre rapport.

## 13) Webdesign

Le webdesign n'est pas un critère d'évaluation à proprement parler. Toutefois, vous devez vous efforcer de rendre quelque chose de présentable et qui donne un minimum envie d'être utilisé. Si vous n'êtes pas à l'aise avec le CSS et l'intégration HTML, basez-vous sur le framework CSS bootstrap qui fournit un très bon point de départ : <https://getbootstrap.com/docs/4.0/examples/>. Enfin, l'application doit être utilisable sur smartphone.