**End-Year Project**
**T-YEP-600**

# Projet de fin d'année
# Cahier des charges

Léo CRENON
Boris DURAND
Titouan LE GOVIC
Christian LESEURRE

# Résumé explicatif du projet

## Problématique

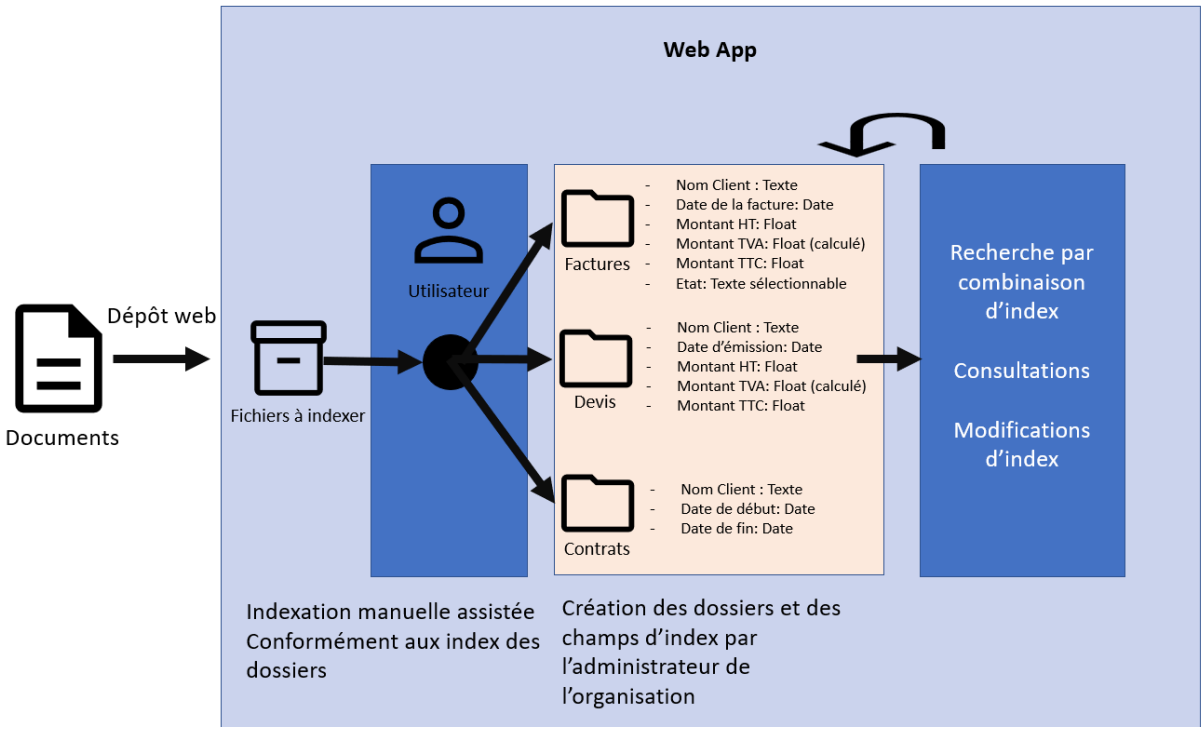
Au sein des organisations (entreprise, association, service publique, etc…), le partage et la gestion de documents et de fichiers peut être une vraie problématique. La classification ou la recherche d'une information précise parmi une quantité de données importante peut être une réelle difficulté sans un service optimal. Certaines entreprises gérant une grande quantité de documents tel que des fiches de paie, des factures ou des bons de commandes ont souvent du mal à trier correctement ces documents entrainant des pertes de temps lors de la recherche d'un document, voir la perte d'un document.

## Solutions

Notre logiciel permettra de faciliter le classement et la recherche des documents grâce à l'intermédiaire de mots clés appelés tags. L'outil s'adapte aux besoins de chacun afin de faciliter la gestion des fichiers en gros volume et d'optimiser le flux de travail des organisations.
Le logiciel permettra de :

- Gérer des fichiers
- Trouver les documents en un seul clic
- Répartir les documents par dossier unique (factures, salaires, bon de commandes, etc.)
- Personnaliser l'indexation dans chaque dossier
- Permettre d'avoir un aperçu sur les documents
- Pouvoir lier plusieurs fichiers entre eux
- Répartir des rôles aux différents utilisateurs

Le logiciel pourra potentiellement être doté de la technologie OCR pour permettre d'indexer plus facilement les PDF. À long terme, il sera disponible sur toutes les plateformes numériques.



# Répartition des charges de travail

## Notre équipe

Le projet se déroule par équipe de 4 à 6 personnes. Pour notre choix de projet nous avons composé une équipe de 4 personnes : Léo CRENON, Titouan LE GOVIC, Christian LESEURRE, Boris DURAND. Nous avons décidé de séparer le groupe en 2 équipes, la première s'occupera de la partie front end du SaaS qui sera porté sur Web, Logiciel PC et potentiellement sur smartphone grâce à Dart avec le SDK Flutter. La seconde équipe s'occupera du back end qui sera une API REST développée en PHP Symfony.

**Répartition :**

- Front end (Léo CRENON & Boris DURAND)
- Back end (Titouan LE GOVIC & Christian LESEURRE)

**Ordre de priorité :**

- 1ère couche
  - Upload
  - Download
  - Affichage
- 2ème couche
  - Triage par dossier
  - Type de fichier
  - Administration pour les dossiers
  - Liaison inter-fichier
- 3ème couche
  - Triage par index
  - Taille des fichiers
  - Date de mise en ligne
  - Administration des index
- 4ème couche
  - Listage des logs des downloads
  - Gestion des utilisateurs
  - Rôles
  - Notifications par mails et sur le logiciel
  - Générer un PDF de récapitulatif
- Couche optionnel
  - OCR
  - Version mobile

# Méthodologie

## Front End

Dans un premier temps, nous créerons des maquettes du visuel du logiciel pour toutes les plateformes à long terme. Le projet sera développé en version Web et Logiciel Ordinateur. En cas de temps bonus, il pourra intégrer une version smartphone. Ces maquettes devront prendre en compte l'ergonomie que l'on peut avoir sur les différentes tailles d'écran et, si la version smartphone est développée, il faudra prendre en compte les différences entre iOS ou Android. Nous ferons donc une étude sur l'ergonomie et l'expérience utilisateur pour que ceux-ci soient intuitifs et agréables à l'utilisation. Nous procéderons à des benchmarks pour voir ce qui fonctionne et ce qui peut être optimisé pour notre cas. Il faudra aussi penser à la charte graphique du logiciel pour donner une identité visuelle au projet et au site web. Nous allons par la suite faire un modèle pour voir ce qui fonctionne bien et ce qui peut être amélioré pour optimiser l'ergonomie sur ces diverses plateformes. Ce modèle ne prendra en charge aucune base de données, les données seront mises en dur dans le visuel des plateformes. Quand le visuel du projet sera validée, nous commencerons l'intégration des url de l'api petit à petit en procédant à des tests en continue.

## Back End

Dans cette partie, nous établirons un diagramme de relations de données.
Nous organiserons la gestion des droits et privilèges des utilisateurs afin de délimiter leur périmètre d'action au sein du système de fichiers de l'organisation. L'utilisateur pourra associer des mots clés à des documents afin de pouvoir les retrouver plus facilement et d'optimiser la gestion des fichiers de l'entreprise.
Par la suite nous mettrons en place la base de données. Pour finir nous construirons l'API.

### Tests

Durant la phase de développement, nous effectuerons des tests unitaires, puis des tests fonctionnels en fin de celle-ci, pour s'assurer du bon fonctionnement de l'application.

### Idées d'implémentation/Améliorations

Afin de diminuer le temps de traitement, un OCR pourrait être mis en place et des masques pourraient être définis par l'administrateur de l'organisation cliente pour remplacer/appuyer les formulaires remplis par les utilisateurs.
Nous pourrons aussi générer des PDF afin de vulgariser et synthétiser les demandes du client.

## Planning prévisionnel

### FRONT (Dart Flutter)

- Ergonomie et expérience utilisateur (2jours)
- Charte Graphique (1jour)
- Maquette (2jours)
    - Page login (0.5jour)
    - Page admin (0.5jour)
    - Page user (0.5jour)
    - Page invité (option) (0.5jour)
- Mode vitrine du front (4jours)
- Modification suite au mode vitrine et ajout de l'API (4jours optionnel)
- Liaisons du front avec les API (10jours)
    - Recherche des différents tags avec l'API (2jours)
    - Upload & download file (2jours)
    - Lecture des fichiers en fonction des tags (2jours)

### BACK (API PHP Symfony)

- Établir un diagramme de relation de données (4jours)
- Mise en place de la BDD (MySQL?) (4jours)
- Mise en place d'un diagramme de classe (2jours)
- Système d'authentification par token avec inscription et login (+ forgot password avec SendInBlue) (2jours)
- Créer une organisation (1jour)
- Gérer les membres (droits par utilisateurs / par groupes) (2jours)
- Gérer des dossiers (1jour)
- Mise en place des liens de documents. (2jours)

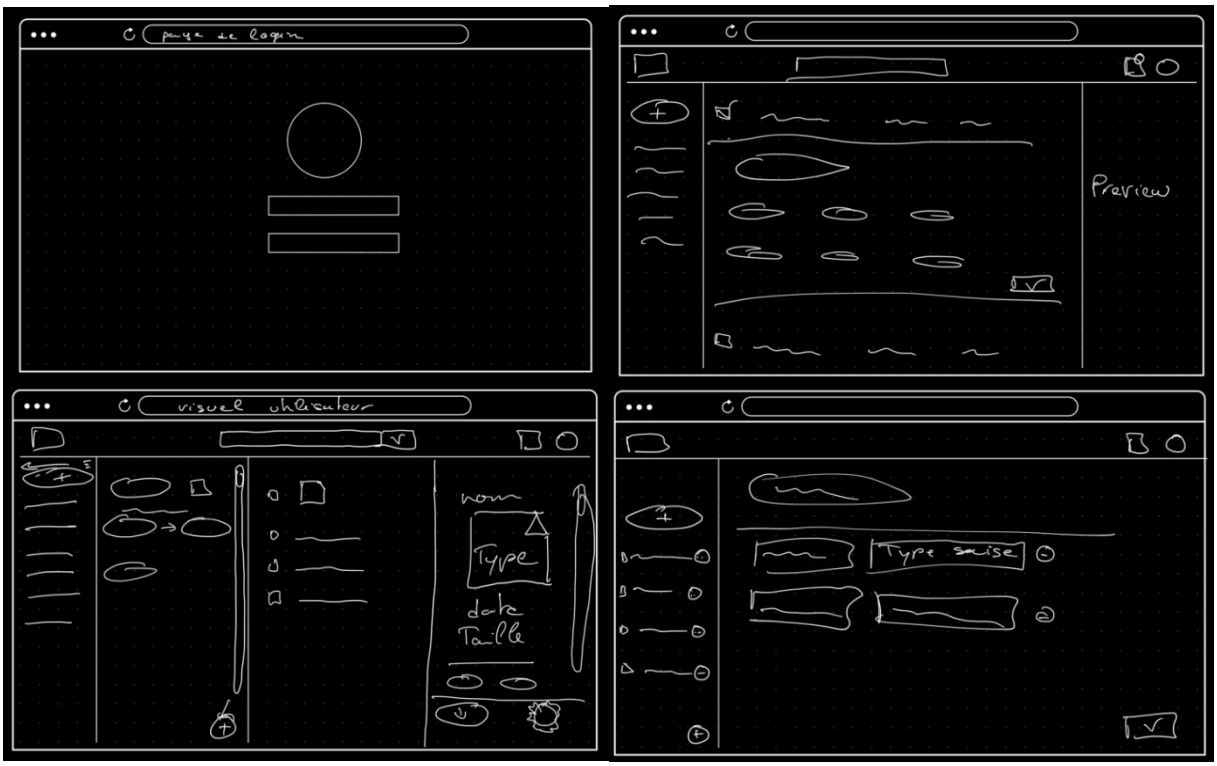
### OCR (Optionnel)

- Océrisation des documents (scans)
- Prise en main de la technologie
- Mettre en place un « Mask-editor » pour OCR
- Tests fonctionnels
- Tests unitaires
- Documentation utilisateur
- Documentation technique

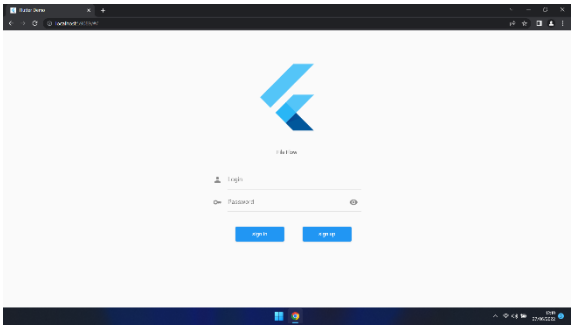
# Compte rendu pour le premier follow-up

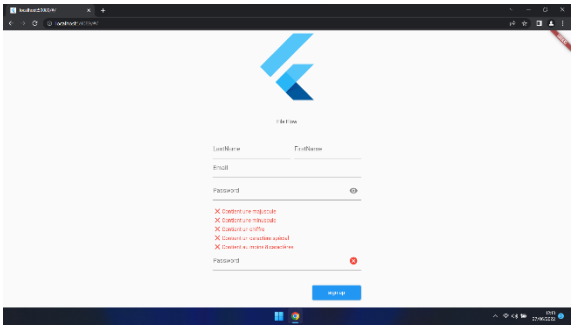
## Front

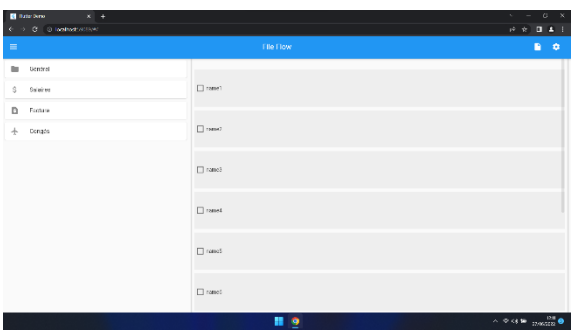
Nous avons dans un premier temps créer un croquis du visuel du site pour une grande partie des pages (login, admin, éditeur, utilisateur).

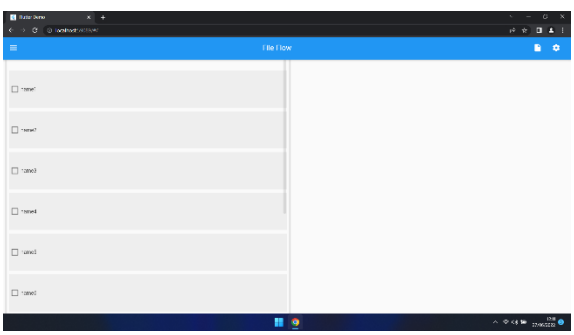



Les croquis ont été validés lors d'une réunion des deux équipes avec l'ajout de quelques modifications. Certains des icônes ou boutons ne s'afficheront qu'en fonctions de certains rôles. Cette partie a pris environ une journée à deux pour les croquis et une heure de réunion à quatre. Ce qui fait un total de 2,5 jours de travail pour la conception du croquis.



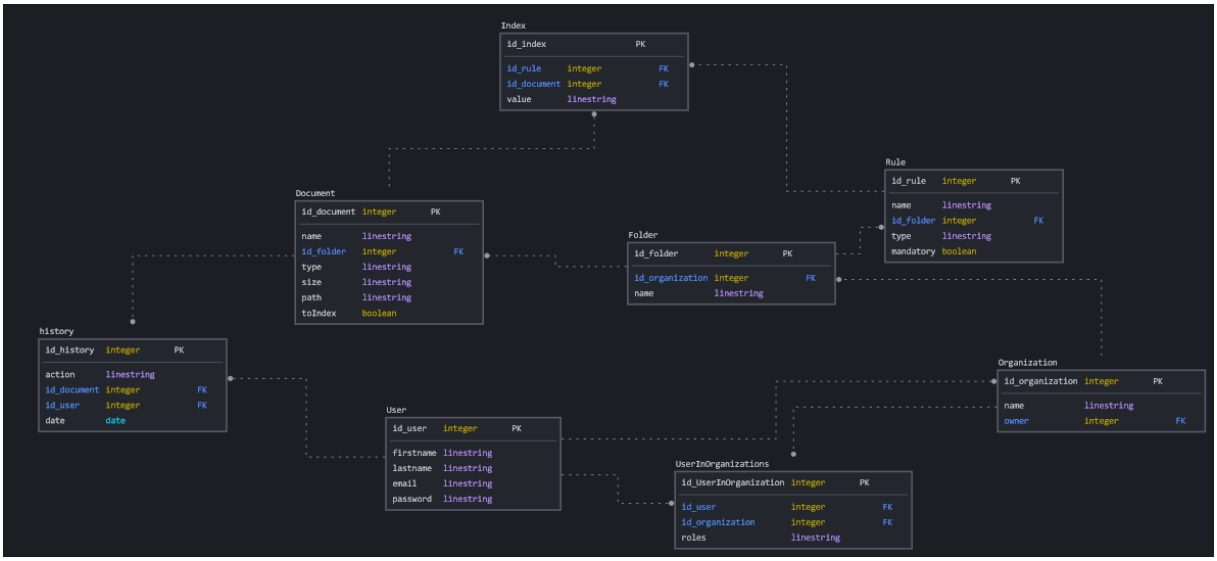






## Back

Pour la partie backend nous avons dans un premier temps fait un modèle conceptuel de données (voir schéma ci-dessous) :




Nous avons créé un projet Symfony afin d'y construire notre API.
Nous l'avons ensuite lié à une base de données MariaDB.
Dans cette API nous avons généré la base de donnée à l'aide de Doctrine, puis nous avons créé toutes les entités, les contrôleurs et mis en place le routage.

| Name | Method | Scheme | Host | Path |
|---|---|---|---|---|
| getDocument | GET\|HEAD | ANY | ANY | /Document/{id} |
| modDocument | PUT | ANY | ANY | /Document/{id} |
| addDocument | POST | ANY | ANY | /Document |
| delDocument | DELETE | ANY | ANY | /Document/{id} |
| download | ANY | ANY | ANY | /Download/{idOrganization}/{idFolder}/{idDocument} |
| modIndices | PUT | ANY | ANY | /Indices/{id} |
| addIndices | POST | ANY | ANY | /Indices |
| delIndices | DELETE | ANY | ANY | /Indices/{id} |
| getOrganization | GET\|HEAD | ANY | ANY | /Organization/{id} |
| modOrganization | PUT | ANY | ANY | /Organization/{id} |
| addOrganization | POST | ANY | ANY | /Organization |
| delOrganization | DELETE | ANY | ANY | /Organization/{id} |
| getRule | GET\|HEAD | ANY | ANY | /Rule/{id} |
| modRule | PUT | ANY | ANY | /Rule/{id} |
| addRule | POST | ANY | ANY | /Rule |
| delRule | DELETE | ANY | ANY | /Rule/{id} |
| getaUser | GET\|HEAD | ANY | ANY | /User/{id} |
| modUser | PUT | ANY | ANY | /User/{id} |
| addUser | POST | ANY | ANY | /User |
| delUser | DELETE | ANY | ANY | /User/{id} |

Nous avons testé les requêtes avec Postman et produit une documentation utilisateur pour le front.
Nous avons en parallèle initié la création du serveur sur une machine virtuelle.