# Analyse : CMS

## Fonctionnel

* 1. Via une interface utilisateur, le service Offre de plbsi a le loisir de mettre à jour les pages institutionnelles
  2. **UC 1 : Création de page**
  3. Il peut créer une nouvelle pages en indiquant :
  + Une référence (unique)
  + Un titre de page
  + le menu (1 seul niveau de menu) auquel elle est rattachée et son rang d’affichage
  + Un booléen (Publiée ou non).   
    Si non publiée, la page n’apparaît pas dans le menu, seul l’URL direct est accessible

On suppose que le gabarit de rendu est unique ou fixé par le menu sélectionné

Après validation de cette première étape, un formulaire d’édition apparaît. Il contient plusieurs blocs de saisie (en fonction du template), la plupart étant rich-text

* 1. **UC 2 : Liste des page**
  2. On peut accéder aux pages déjà créées, via une liste. Les colonnes reprennent la référence, le titre, la rubrique (menu). Il est possible de filtrer par la rubrique. Des liens permettent d’accéder au formulaire d’édition ou d’ouvrir une nouvelle fenêtre permettant de prévisualiser le rendu final.
  3. **UC 3 : Edition de page**
  4. Formulaire avec plusieurs blocs rich-text. Le bloc rich-text propose des styles CSS prédéfinis correspondant à la charte graphique de PLB

## Architecture

Base de données : Base MySQL *plbconsult* utilisée par ***plbsi***

Service Back-end : Application Spring Boot ***plbsi-api*** (déjà existante : Importation des données partenaires)

Front-end : Applications Angular 5/6 avec composants Angular/Bootstrap et CKEditor (Editeur rich-text présent actuellement dans plbi sous sa version JSF. Voir https://ckeditor.com/docs/ckeditor5/latest/builds/guides/integration/frameworks/angular.html)

## Modèle de donnée :

* + Une table rubrique (menu)
  + Une table pages avec les colonnes
    - Id
    - Référence, Nom
    - IdRubrique, rangRubrique
    - template
    - contenu
    - publié (Oui / Non)

## Estimation pour cet incrément

Le service back-end est déjà mis en place, il suffit d’ajouter les tables et le code Java

Back-end : 2 jours

Front-end :

* Mise en place environnement de dév, de recette, de production + jobs jenkins : 1,5j
* Code classique Angular (Liste/formualire/navigation) : 2 j
* Intégration CKEditor : 1j

=> Total 4,5j

Tests/recette : 2j