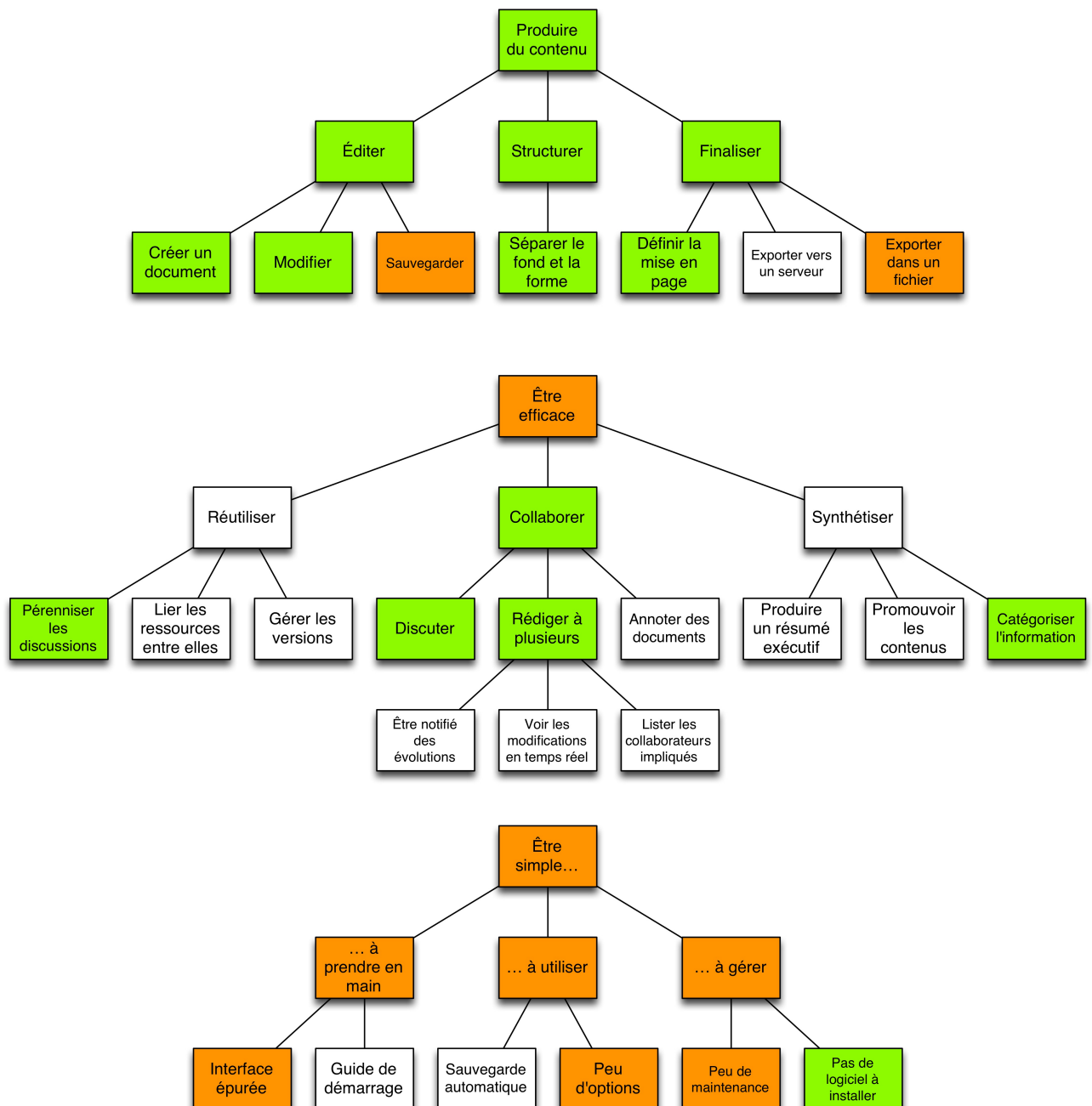


Dossier d'initialisation de phase II

Analyse fonctionnelle

Conformément à l'analyse effectuée dans le cahier OSEO, nous présentons ci-dessous le résultat de notre analyse fonctionnelle.



Sont représentées en vert les fonctions principales, en orange les fonctions contraintes et en blanc les fonctions optionnelles.

Diagramme de processus de l'architecture d'un système de gestion de documents (SGD) pour la vidéo.

Colonnes de dépendance fonctionnelle :

- ↑ Dépendance fonctionnelle
- ▭ Fonction nécessaire
- - - - - Fonction optionnelle

Processus chronologiques (12 mai à 17 juin) :

- Création d'une infrastructure de base** (12 mai) :
 - Choix de l'architecture
 - Choix de l'implémentation
 - Système de gestion de base de données
- Gestion des utilisateurs** (15 mai) :
 - Intégration avec un service d'authentification
 - Gestion des utilisateurs en base de données
 - Groupes hiérarchiques d'utilisateurs
 - Permissions
- Édition de documents sans mise en page** (20 mai) :
 - Interface de rédaction de création de documents
 - Rédaction d'un résumé succinct
 - Stockage en base de données
- Discussions en ligne** (20 mai) :
 - Interface de rédaction
 - Priorisation d'une discussion instanciée
 - Discussions instanciées liées à un document
- Édition de documents avec mise en page** (25 mai) :
 - Interface de rédaction
 - Sauvegarde manuelle
 - Sauvegarde automatique
 - Génération des références
 - Stockage en base de données
- Mise en page d'un document** (5 juin) :
 - Interface d'édition de mise en page
 - Affichage de la mise en page d'un document
 - Stockage en base de données
- Gestion des versions** (5 juin) :
 - Stockage en base de données
 - Ajouter une version
 - Revenir en arrière
 - Afficher les différences
- Export & Interopérabilité** (10 juin) :
 - Export PDF
 - Export Word
 - Export serveur externe
- Soutenance** (17 juin) :
 - Recette

Éléments de collaboration :

- Édition temps réel** : Édition à plusieurs, Visualisation temps réel.
- Classification** : Notifications.
- Envoi d'invitations** : Interactions e-mail.
- Collaboration** : Envoi d'invitations, Interactions e-mail.

Feuille budgétaire

Dans cette section, nous listons l'ensemble des ressources nécessaires afin de réaliser notre prototype.

Nous réalisons une application web destinée à être accessible sur internet à nos clients *via* une offre SaaS. Il nous faut en conséquence un serveur permettant d'héberger notre application ainsi que ses données. Pour le prototype, nous nous orientons vers un serveur KS R-8G proposé par la société OVH.com, 3^{ème} hébergeur mondial. Ce serveur sera largement suffisant pour mettre en place notre prototype, et son coût est de 29,99€ HT par mois. Nous prévoyons d'utiliser ce serveur pendant trois mois, pour un coût total de 89,97€.

D'autre part, notre produit est de manière inhérente une application temps réel, ce qui est un domaine complexe et difficile à appréhender. Afin de monter en compétences sur ce domaine, nous trouvons judicieux de participer à la *Realtime Conference 2013* qui se tiendra à Lyon les 22 et 23 avril 2013. Cet événement auquel participent des conférenciers d'entreprises prestigieuses et reconnues telles que Google, øMQ, Mozilla, Vodafone, etc. De plus, de nombreux représentants de logiciels libres sont présents, ce qui nous aidera dans nos choix techniques. Pour des raisons financières, nous prévoyons de n'envoyer que deux personnes à cette conférence, pour un coût total de 800€.

Au total, le coût de notre prototype revient à 889,97€.

Analyse des risques

Une analyse des risques inhérents à cette seconde phase de projet, complétant celle établie dans le cahier OSEO, est essentielle pour appréhender cette seconde phase du projet :

- *choix des technologies* : il a été choisi de s'appuyer sur des technologies novatrices et récentes pour la réalisation de notre projet (*frameworks web*, base de données orientée documents, paradigmes de programmation) ; cependant, ne possédant aucune expérience sur ces dernières, il est de notre devoir de s'assurer que ces choix n'entraveront pas la réalisation. Ainsi, une solution de repli connue et éprouvée est prévue pour chaque élément de l'architecture
- *retard dans la réalisation* : bien que des méthodes agiles seront suivies pour le développement du produit et des objectifs clairs posés, il est plus que probable que l'imprévu se mêle à nos travaux : il sera donc essentiel de concentrer nos efforts sur ce qui fait la force de notre produit et définir des priorités entre les différentes fonctionnalités
- *interface peu intuitive* : nous l'avons clamé haut et fort dans notre cahier OSEO : l'interface de notre produit constitue une priorité et il est important qu'elle soit simple et

intuitive pour ne pas déboussoler l'utilisateur. Une équipe sera donc consacrée à cette tâche et des bêta-tests auprès de personnes extérieures au projet seront menés pour évaluer l'ergonomie de l'interface

- *mauvaise organisation du groupe* : la phase II étant courte, il est probable qu'une mauvaise coordination puisse entraver le bon fonctionnement du projet. Pour contrer ce risque, une bonne communication est primordiale, et le chef de projet en sera le garant.
- *produit peu performant* : dans le cas où les performances ne seraient pas à la hauteur des attentes des utilisateurs, des investigations seront menées pour déterminer le ou les goulots d'étranglement. Ainsi, les choix d'architecture pourront être remis en cause
- *concurrence* : nous ne sommes pas à l'abri de l'émergence de produits concurrents qui embrasseraient le marché avant nous ; cependant, nous croyons en le caractère innovant de notre solution et en notre capacité à développer et déployer une première version de notre produit rapidement

Éléments nécessaires pour entamer la seconde phase du projet

Entamer la phase II de notre projet du bon pied nécessite quelques ressources clés.

Tout d'abord, nous jugeons nécessaire de s'accorder sur l'interface que proposera notre produit. En effet, une interface claire et réfléchie est essentielle à une bonne expérience utilisateur. De plus, celle-ci doit être pensée au préalable afin d'éviter toute surprise ou désaccord au sein de l'équipe lors du développement du logiciel. Pour cela, nous prévoyons de réaliser un ensemble de *mockups* ou prototypes d'interface utilisateur, qui constitueront un livrable interne.

D'autre part, notre produit étant un logiciel utilisant de nombreux composants externes tels qu'une base de données et des bibliothèques, il est primordial d'effectuer les choix techniques en amont afin de constituer des fondations solides. Nous éviterons ainsi de perdre du temps. Pour nous assister dans ce choix, nous prévoyons d'envoyer deux membres de l'équipe à la *Realtime Conference* mentionnée plus haut. Une fois ces choix techniques effectués, les différents membres présenteront leur décision à l'équipe. Une fois ces choix approuvés un livrable interne sera rédigé.

Enfin, il est crucial de mettre rapidement en place l'ensemble des outils sélectionnés précédemment et le serveur d'hébergement afin de garantir leur bon fonctionnement et une prise en main rapide par l'ensemble de l'équipe. Une phase d'initialisation est alors nécessaire, comme détaillé dans la section d'organisation de la phase II.

Fiche de recette

Afin de vérifier le bon fonctionnement de notre prototype à la fin de la phase II, il est primordial de s'accorder sur des tests de recette. Ils garantissent la capacité de notre prototype à remplir un certain nombre de fonctions. Cette fiche de recette est détaillée ci-dessous.

Informations sur la procédure

Objet de la procédure de recette Cette procédure a pour objectif la validation de la conformité de notre produit selon des critères objectifs et connus à l'avance. Elle concerne l'ensemble logiciel formé par le logiciel client, le logiciel serveur et la base de données. Nous nous inscrivons ainsi dans une démarche qualité forte.

Étapes de recette Nous distinguons deux catégories de tests à effectuer. Les *tests techniques* valident l'ensemble des considérations techniques associées à l'application, telles que son architecture ou ses performances. À l'inverse, les *tests fonctionnels* contrôlent la capacité de l'application du point de vue de l'utilisateur final à effectuer l'ensemble des tâches qui lui sont confiées. Ces tests fonctionnels doivent permettre de valider l'ensemble des fonctions qualifiées de principales dans l'analyse fonctionnelle détaillée plus haut dans ce document.

Conditions de la recette Cette procédure sera effectuée en fin de phase II - conformément au planning détaillé précédemment - et ne concerne ainsi que la version finale du *prototype* développé pendant cette durée. Dans un souci d'économie de temps, l'ensemble des tests techniques sera effectué en amont par l'équipe projet. En revanche, les tests fonctionnels pourront être effectués indifféremment par l'équipe tutorale ou par l'équipe projet. Un procès-verbal de recette listant les tests effectués et leurs résultats sera dressé au cours de la séance.

Procès-verbal de recette

Fonction	Tests à réaliser
Gestion des utilisateurs	<ul style="list-style-type: none">* Le visiteur peut créer un nouveau compte [Oui / Non]* Le visiteur peut se connecter avec son compte [Oui / Non]* Une fois connecté, le visiteur a accès à des informations personnalisées [Oui : un panneau indiquant des notifications personnalisées / Oui : ses informations de profil uniquement / Non]

	<p>* L'utilisateur fait partie de groupes d'utilisateurs [Oui : pas de limitations sur le nombre de groupes / Oui : 1 seul groupe par utilisateur / Non]</p> <p>* Le visiteur peut se déconnecter [Oui / Non]</p>
Gestion des permissions	<p>* Les documents d'un utilisateur lui sont privés [Oui / Non : visibles par tous]</p> <p>* Un utilisateur peut donner des droits en lecture et écriture à d'autres utilisateurs sur des documents [Oui : granularité du document / Oui : sur tous ses documents à la fois / Non] [Oui : granularité de l'utilisateur / Oui : granularité du groupe / Non] [Oui : granularité en lecture et écriture / Oui : lecture-écriture liées / Non]</p> <p>* Un utilisateur peut nommer un autre utilisateur responsable de section [Oui / Non]</p> <p>* Seul le responsable de section peut fermer une discussion y appartenant [Oui / Non]</p>
Rédaction d'un document	<p>* L'utilisateur connecté peut créer un nouveau document vierge [Oui / Non]</p> <p>* L'utilisateur connecté peut rédiger son document [Oui / Non]</p> <p>* L'utilisateur connecté peut sauvegarder son document [Oui : sauvegarde automatique / Oui : sauvegarde manuelle / Non]</p> <p>* La rédaction d'un document sauvegardé peut être reprise plus tard [Oui / Non]</p> <p>* Un document peut être catégorisé <i>via</i> des mots-clés [Oui : autant de mots-clés que souhaités / Oui : 1 mot clé uniquement / Non]</p> <p>* L'utilisateur peut rechercher les documents [Oui : recherche par mots-clés et titre / Oui : recherche par titre uniquement / Non]</p>

	<p>* Le fond est séparé de la forme [Oui / Non]</p>
Relecture d'un document	<p>* L'utilisateur connecté peut visualiser un document sauvegardé [Oui : en même temps que d'autres / Oui : 1 personne par document maximum / Non]</p> <p>* L'utilisateur connecté peut exporter un document sauvegardé dans un fichier brut, proposé au téléchargement [Oui / Non]</p> <p>* L'utilisateur connecté peut exporter un document sauvegardé dans un fichier avec mise en page, proposé au téléchargement [Oui / Non]</p>
Discussions	<p>* L'utilisateur connecté peut lancer une discussion à propos d'un document [Oui / Non]</p> <p>* L'utilisateur connecté peut participer à une discussion [Oui / Non]</p> <p>* L'utilisateur connecté peut commenter une partie précise d'un document [Oui : granularité de la lettre / Oui : granularité de la ligne / Oui : granularité du paragraphe / Non]</p> <p>* Le système met en valeur les posts pertinents dans une discussion [Oui : selon un algorithme complexe / Oui : selon le nombre de commentaires associés / Non]</p> <p>* Il est possible de rédiger un résumé exécutif d'une discussion [Oui / Non]</p>
Gestion des versions	<p>* Un document peut être versionné [Oui : autant de versions que voulu / Oui deux versions seulement / Non]</p> <p>* Si oui, il est possible de visualiser les anciennes versions d'un document [Oui / Non]</p> <p>* Si oui, il est possible d'exporter les anciennes versions dans un fichier [Oui / Non]</p>

	<ul style="list-style-type: none"> * Créer une nouvelle version archive les discussions fermées [Oui : il est possible de relire les conclusions / Oui : impossible de relire les conclusions / Non : les discussions fermées restent toujours visibles]
Architecture technique	<ul style="list-style-type: none"> * La base de données peut communiquer avec le serveur [Oui : ajout et récupération de données stockées / Non] * L'application client peut communiquer avec le serveur [Oui : envoi de requêtes du client vers le serveur / Non]
Performances	<ul style="list-style-type: none"> * Le temps moyen d'une requête d'ouverture d'un document est inférieur à 3 secondes [Oui / Non] * Le temps moyen du chargement d'une page est inférieur à 3 secondes [Oui / Non] * Un utilisateur peut créer au moins 10 documents à partir de son compte [Oui / Non]
Divers	<ul style="list-style-type: none"> * Les bibliothèques utilisées sont utilisables dans un cadre commercial [Oui / Non] * Présence d'une documentation technique concernant l'API [Oui : détaillée / Oui : faible / Non] * Présence d'une documentation technique concernant la procédure d'installation [Oui : détaillée / Oui : faible / Non]