**Travail de Bachelor**

Étude et mise en place d’une plateforme web facilitant la gestion de projets entre mandants et équipes de développeurs indépendants

Non confidentiel

|  |  |  |
| --- | --- | --- |
| 🏢  **Mandant** | ➡️  Liste de fonctionnalités  Délai | 🧑‍💻🧑‍💻🧑‍💻  **Équipe de développeurs** |
| ⬅️  Cahier des charges  Coût |
| ✅  Produit / Rémunération |
| 🐟  **MoonFish** | | |

|  |  |
| --- | --- |
| **Étudiant :** | **Thibaud Alt** |
| **Travail proposé par :** | Guillaume Wägli |
| **Enseignant responsable :** | Patrick Lachaize |
| **Année académique :** | 2020-2021 |

Yverdon-les-Bains, le 24 avril 2021

**Table des matières**

[1 Introduction 2](#_Toc70153051)

[1.1 But du document 2](#_Toc70153052)

[1.2 Problématique 2](#_Toc70153053)

[1.3 Objectifs 2](#_Toc70153054)

[1.4 Valeur ajoutée 3](#_Toc70153055)

[2 Cahier des charges 5](#_Toc70153056)

[2.1 Éléments généraux 5](#_Toc70153057)

[2.2 Éléments d’études 6](#_Toc70153058)

[2.3 Besoins fonctionnels 7](#_Toc70153059)

[2.4 Besoins non-fonctionnels 10](#_Toc70153060)

[2.5 Extensions 10](#_Toc70153061)

[3 Analyse 12](#_Toc70153062)

[3.1 Étude de marchée 12](#_Toc70153063)

[3.2 Graphisme et ergonomie 12](#_Toc70153064)

[3.3 Risques 12](#_Toc70153065)

[4 Modélisation 13](#_Toc70153066)

[5 Conception 14](#_Toc70153067)

[6 Planification 15](#_Toc70153068)

[7 Réalisation 16](#_Toc70153069)

[8 Tests 17](#_Toc70153070)

[9 Conclusion 18](#_Toc70153071)

# Introduction

## But du document

Ce document représente le travail de Bachelor réalisé durant le dernier semestre de formation de la filière Informatique avec orientation « *Systèmes de gestion* » à la Haute École d’Ingénierie et de Gestion du canton de Vaud.

## Problématique

Aujourd'hui, la gestion de projets informatiques respecte la plupart du temps le même processus :

1. Un mandant soumet une liste de fonctionnalités et un délai pour la production d’un produit
2. Un vendeur (et/ou directeur, chef de projet) décrit un cahier des charges et propose un devis au mandant
3. Le vendeur négocie avec le mandant la forme du projet
4. Le mandant accepte et signe un devis comprenant un coût, une qualité et un délai
5. Le vendeur impose un cahier des charges et des délais de réalisation à une équipe de développeurs
6. L’équipe de développeurs implémente le projet et crée un livrable
7. Le vendeur livre le produit fini
8. Le mandant paie le projet (et les dépassements de coûts, la baisse de qualité, le non-respect des délais !)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 🏢  **Mandant** | ➡️  Liste de fonctionnalités  Délai | 💼  **Vendeur**  **Chef de projet** | ➡️  Cahier des charges  Délais | 🧑‍💻  **Développeur** |
| ⬅️  Coût  Qualité | ⬅️  Produit |

Les problèmes rencontrés lors de ces projets sont nombreux, les principaux et les plus problématiques sont les suivants :

* Pour le **mandant** : Dépassement des coûts, non-respect des délais, diminution des fonctionnalités…

*→ Finalement le mandant se retrouve le jour du délai demandé avec aucun livrable ou un livrable ne contenant qu’une partie des fonctionnalités et avec un coût planifié atteint voir dépassé.*

* Pour les **développeurs** : Cahier des charges, problèmes technologiques, manque de temps…

*→ Finalement les développeurs doivent fournir un livrable bâclé présentant des bugs dus au manque de temps, avec une documentation faible voire inexistante.*

## Objectifs

La solution envisagée se base sur la communication directe entre le mandant et l’équipe de développeurs. Le processus se simplifie alors comme suit :

1. Un mandant met en concours une liste de fonctionnalités et un délai pour la production d’un produit
2. Différentes équipes de développeurs décrivent un cahier des charges comprenant une liste de fonctionnalités réalisables, un choix technologique, une qualité et un coût dans le délai donné
3. Le mandant accepte et signe un des cahiers des charges proposés
4. L’équipe de développeurs implémente le projet pour lequel elle s’est engagée et livre un produit final que le mandant paie

Pour mettre en relation les mandants et les développeurs, faciliter la communication, proposer des devis, gérer les délais et les flux financiers la solution s’apparente à créer et développer **une plateforme web**.

|  |  |  |
| --- | --- | --- |
| 🏢  **Mandant** | ➡️  Liste de fonctionnalités  Délai | 🧑‍💻🧑‍💻🧑‍💻  **Équipe de développeurs** |
| ⬅️  Cahier des charges  Coût |
| ✅  Produit / Rémunération |

Au travers de cette plateforme, les équipes de développeurs sont en "compétition sociale" entre elles. Le fait de choisir ses coéquipiers, de définir les technologies, les prix, les choix laissés aux équipes et la dimension sociale motivent et permettent de lisser la plupart des problèmes.

En cas de livraison d’un produit final exemplaire ou au contraire d’un échec (non-respect d’un délai d’un projet, diminution des fonctionnalités, abandon, etc.) les équipes sont notées publiquement. Ainsi la réputation d’une équipe permet à celle-ci de grandir, de décrocher plus de projets et assure la qualité et la réussite des projets.

## Valeur ajoutée

Le but premier de cette plateforme web est d’éliminer un ou plusieurs intermédiaires ceux-ci étant une cause de multiplication des problèmes. Prenons l’exemple du célèbre jeu du *téléphone arabe*, plus il y a d’acteurs, moins le message initial a de chance d’arriver correctement et non déformé au destinataire final. De plus, chaque être humain à une façon personnel d’émettre et d’interpréter les informations qu’il reçoit d’autres êtres humains. Dans un projet, ces éléments s’appliquent également ; plus il y a d’acteurs, plus le contenu du projet diverge, plus le temps de celui-ci est allongé et plus les coûts sont élevés.

L’optique de cette plateforme web est d’éliminer au maximum les acteurs intermédiaires et de mettre en valeur les acteurs apportant une réelle plus-value à un projet. Dans le but d’arriver à créer un produit en réduisant les coûts tout en respectant les délais et en visant une qualité requise. Ces acteurs sont, *tout comme au téléphone arabe l’émetteur et le destinataire du message,* le ou les mandants et le ou les développeurs. Pour que ces deux mondes se comprennent et arrivent à collaborer dans le but de créer un produit répondant au besoin, cela demande des efforts de compréhension des deux côtés ainsi qu’une implication forte de chacun dans le projet.

Pour aider la communication entre les différents acteurs tout au long du projet, diriger la gestion de celui-ci et ainsi remplacer et améliorer les rôles des intermédiaires, la plateforme web disposera de différentes valeurs ajoutées. Premièrement, la gestion des projets sera faite en respectant un cadre de travail SCRUM qui permet d’encadrer, de standardiser et d’ainsi faciliter la conduite des projets. De plus, la plateforme mettra à disposition de ses clients différents services notamment :

* Un espace de travail incluant un forum permettant un échange direct d’informations, de fichiers, d’images, etc. qui sera accessible et à disposition de tous les acteurs du projet.
  + En plus de la version originale du message et si nécessaire, cet espace inclura un système de traduction des messages dans la langue préférée de l’utilisateur. Permettant ainsi à des acteurs de toutes régions du monde de travailler ensemble facilement.
  + Des livrables avec une gestion des versions seront mis à disposition du mandant tout au long du projet, celui-ci pourra alors vérifier et corriger en cours de réalisation le bon déroulement du projet.
  + Une intelligence artificielle pourra être interpelée en cas d’incompréhension de vocabulaire entre mandants et développeur, celle-ci pourra reformuler certaines phrases, vulgariser certains termes et apporter des éléments supplémentaires nécessaires à une bonne compréhension.
* La mise à disposition sporadique de professionnels externes pour traiter un point spécifique du projet. Par exemple le recours à un médiateur en cas de conflit entre certains acteurs du projet, l’appel à un consultant ou à un expert pour prendre une décision stratégique, le suivi par un coach en cas de baisse de motivation, etc.
* Un système de sous-traitance de tâches simples et répétitives permettant de répondre rapidement à un besoin ponctuel. Par exemple la saisie de centaines d’articles dans un magasin de vente en ligne, le nettoyage d’une base de données, la recherche d’images d’illustrations, etc.
* Un *espace de connaissances* mettant à disposition des ressources, des cours, des tutoriels, des outils, etc. apportant des notions théoriques ainsi que différents savoirs nécessaires liés à la gestion de projets.
* Une équipe d’assistance disponible en tout temps pour répondre aux différentes questions liées à la plateforme et ainsi aider ses clients à tout moment.

Dans ce travail de Bachelor, nous nous focalisons sur des produits informatiques pour développer l’idée, mais ce concept pourrait très bien s’adapter à tout type de projets et d’industries. Nous pourrions par exemple imaginer une entreprise de construction avec comme projet l’élaboration d’un bâtiment même si dans ce cas, il y aurait certainement plus que deux types d’acteurs.

# Cahier des charges

## Éléments généraux

### Objectifs du travail de diplôme

Les objectifs de ce projet pour ce travail de Bachelor sont les suivants :

* Réaliser une étude de marché sommaire
  + Travail de recherche de "ce qui se fait" actuellement et des éventuels produits existants concurrents
* Analyser via un "*State of the art*" les différentes techniques permettant le développement d’applications web en 2021 dans le but d’une sélection pour la réalisation
* Définir la structure et la technologie de la ou des base(s) de données à utiliser
* Développer une première version de l’application web client-serveur
* Proposer des améliorations et/ou d'autres fonctionnalités à développer dans des versions postérieures de l’application web

### Périmètre

Dans sa première version, l’application web on attend au minimum les fonctionnalités et point de conceptions suivants :

* La plateforme permettra à un utilisateur de se créer un compte, de gérer son profil et de rejoindre une ou plusieurs équipes de développeurs
* Un mandant pourra se créer un compte, gérer son profil, soumettre un ou plusieurs projets, choisir une équipe de développement et attribuer une évaluation à une équipe de développeurs lorsqu’un projet sera finalisé
* Une équipe de développeur (via un *team leader*) pourra gérer son profil, soumettre sa candidature pour des projets proposés et déposer des livrables pour ses projets en cours
* L’application web sera monolingue et sera proposé en anglais
* L’application web disposera d’une interface fonctionnelle sur les navigateurs web récents et sur une taille d’écran d’ordinateurs classiques

### Planning

Pour ce projet, deux possibilités de rendus sont possibles :

1. La première nécessitant un taux de travail à 100% consiste à un rendu intermédiaire à la mi-juillet et un rendu final à la fin août.
2. La première nécessite un taux de travail d’uniquement 60% et consiste à un rendu intermédiaire à la fin-juillet et un rendu final à la fin septembre.

Ayant des obligations professionnelles et ne pouvant pas réduire mon taux de travail pour les mois de juillet à septembre, j’ai opté pour la seconde option proposée. De ce fait, le planning suivant en découle.

|  |  |
| --- | --- |
| **Date** | **Échéance** |
| Vendredi 21 mai 2021 | *Rendu du cahier des charges* |
| Mardi 13 juillet 2021 | *Rendu intermédiaire incluant le rapport intermédiaire* |
| Jeudi 7 octobre 2021 | *Rendu final incluant le rapport final et l’application fonctionnelle* |
| Du 25 octobre au 5 novembre | *Soutenance du travail de Bachelor* |

## Éléments d’études

### Étude de marché sommaire

Un travail de recherche de "ce qui se fait" actuellement sera réalisé et une étude de marché sommaire présentera les éventuels produits existants concurrents. Cette étude pourra être composé d’une matrice d'affaires (*Business Model Canvas*), des différentes cibles visées par la plateforme web, du marché potentiel et du profil des clients, du secteur d’activité ou encore des éventuels risques et menaces.

### Réception des livrables, validation et paiement

Le déroulement d’un projet, sa validation puis sa rémunération suivent les étapes suivantes.

1. Le mandant signe un devis et paye un acompte à la plateforme.

*Développement du projet…*

1. L’équipe de développeurs a terminé le projet, elle le notifie sur la plateforme.
2. Le mandant doit alors régler le montant total du projet à la plateforme.
3. L’équipe de développeurs livre et éventuellement installe le projet chez le mandant puis en notifie la plateforme. La réception proprement dite des livrables est gérée directement entre le mandant et l’équipe de développeurs.
4. Le mandant complète un rapport sur le déroulement du projet et sur l’équipe de développeurs puis valide la livraison et éventuellement l’installation du projet.
5. La plateforme rémunère l’équipe de développeurs et clos le projet.

Si le mandant ne valide pas la livraison du projet, un expert de la plateforme intervient pour régler le conflit et trouver une solution entre les deux parties. Si le projet est abandonné en cours de route par le mandant, l’équipe de développeurs récupère l’acompte payé par celui-ci.

#### Changement d'équipe de développement

Si en cours de projet, l’équipe de développeurs ou le mandant souhaitent rompre le contrat, les deux parties doivent remplir un rapport expliquant les raisons de cette rupture. Ensuite, le projet est proposé une seconde fois sur la plateforme dans son état actuel. Différentes équipes de développeurs peuvent alors proposer leurs services et le mandant doit à nouveau choisir une équipe pour terminer le projet. Le passage du code source, des processus de déploiement, de la documentation, etc. se fait via la plateforme par l’équipe de développeurs « sortante ». Celle-ci se doit de mettre à disposition de la nouvelle équipe de développeurs tous les éléments nécessaires à la bonne continuation du projet.

Un abandon du projet par une équipe de développeurs entraine de fait un signalement sur le profil de cette équipe. Ce signalement contient le rapport des raisons de l’abandon et est noté par rapport à différents critères notamment sur le travail réalisé, la documentation, le passage des éléments nécessaires, la réactivité, etc.

### Technologies

Le but est de réaliser une application web client-serveur entièrement en JavaScript à l’aide de Node.js (où Deno) et de frameworks comme Express.js, React.js, Vue.js ou équivalent. Pour réaliser cette plateforme l’utilisation des technologies web récentes et actuelles semble cohérente, ces choix techniques devront être vérifiés et validés dans une phase d’analyse.

#### State of the art

Un état de l'art des techniques permettant le développement d’applications web en 2021 sera réalisé. Celui-ci s’intéressera globalement à diverses technologies disponibles puis plus particulièrement aux technologies JavaScript. Cet état de l’art étudiera les deux axes de développement nécessaire, à savoir le front-end avec des frameworks JavaScript et le back-end avec les environnements d'exécution et les frameworks.

#### Persistance des données

Le choix de la technologie de la ou des bases de données à utiliser devra être étudié. Pour stocker les informations des utilisateurs, les informations spécifiques aux projets, les évaluations (…) une unique base de données SQL semble adéquate. Cependant ce choix devra être confirmé et validé durant la phase d’analyse.

#### Gestion de dépendances

La gestion de dépendances sera réalisée avec *npm* qui est le gestionnaire de paquets officiel de Node.js si ce dernier est choisi lors de la phase d’analyse pour y développer la plateforme. Ce gestionnaire de paquet est très pratique car il fonctionne avec un simple terminal, gère les dépendances par application et permet d'installer très facilement des paquets Node.js disponibles sur le dépôt npm. En outre, toutes les informations nécessaires au développement et au déploiement sont écrites en clair dans un fichier JSON ce qui permet de gérer les dépendances de librairies tierces et d’automatiser leur téléchargement.

#### Livraison et intégration continue (CI / CD)

L’environnement d’intégration continue et de déploiement continu utilisé sera git à l’aide de la plateforme web github.com et du logiciel GitHub Desktop. Cet environnement et ses outils associés mettent à disposition un système de gestion des versions complet, ainsi qu’un puissant système de tests et de déploiement.

#### Livrables

Désirant réaliser le développement de l’application web avec la méthodologie SCRUM, celle-ci prévoit de générer autant de livrables que de *sprints* agendés. Une fois les différents *sprints* définis, chaque livrable sera clairement identifié sur le système de gestion des versions. Celui-ci pourra alors éventuellement être déployé, hébergé et soumis au client.

#### Hébergement

Les différents livrables pourront être déployés au fur et à mesure de son développement sur une plateforme cloud tel que *AWS* (*Amazon Web Services*), *Heroku* ou encore *Netlify*. Les principaux avantages de ses plateformes cloud sont qu’elles permettent un déploiement extrêmement rapide qui peut être automatisé avec plusieurs outils de livraison continue, qu’elles ne nécessitent pas de configurations complexes et qu’elles sont gratuites dans une certaine mesure.

## Besoins fonctionnels

### User stories

Une « *user story* » est une explication non formelle, générale d'une fonctionnalité logicielle écrite du point de vue de l'utilisateur final. Son but est d'expliquer comment une fonctionnalité logicielle apportera de la valeur au client.

Dans les user stories suivants, nous prendrons trois points de vue différents à savoir :

* Un utilisateur (il s’agit ici d’un développeur)
* Un mandant
* Une équipe (il s’agit ici de plusieurs développeurs)

Les degrés de priorisation vont de 1 à 3 ; 1 étant la priorité la plus importante et 3 la priorité la moins importante. Les niveaux de complexités s’étendent de 1 à 5 ; 1 étant une tâche facile et 5 une tâche complexe.

#### Création d’un compte et authentification

|  |  |
| --- | --- |
| *Identifiant* | Epic 1 |
| *Titre* | Création d’un compte et authentification |
| *Description* | En tant que développeur ou en tant que mandant, je veux pouvoir me créer facilement un compte utilisateur puis l’utiliser par la suite pout m’authentifier. |
| *Tâches associées* | 1. L’utilisateur doit pouvoir se créer un compte avec au minimum un nom, une adresse email et un mot de passe. 2. L’utilisateur doit pouvoir se connecter et accéder à son compte. 3. L’utilisateur doit pouvoir récupérer son mot de passe en cas d’oublis de celui-ci. |
| *Priorisation* | 1 / 3 |
| *Complexité* | 3 / 5 |
| *Auteur* | Thibaud Alt |
|  |  |

#### Gestion de profil

|  |  |
| --- | --- |
| *Identifiant* | Epic 2 |
| *Titre* | Gestion de profil |
| *Description* | En tant que développeur ou en tant que mandant, je veux pouvoir gérer mon profil. En tant que développeur je peux rejoindre ou quitter une ou plusieurs équipes de développeurs. |
| *Tâches associées* | 1. L’utilisateur doit pouvoir ajouter/modifier/supprimer ses données de contacts (nom, prénom, adresse, etc.). 2. L’utilisateur doit pouvoir modifier ses informations de connexions (email et mot de passe). 3. L’utilisateur doit pouvoir ajouter/modifier/supprimer ses informations professionnels (CV, langages connus, etc.). 4. L’utilisateur doit pouvoir rejoindre et quitter une équipe de développeurs. 5. L’utilisateur doit pouvoir créer une équipe de développeurs. 6. L’utilisateur doit pouvoir gérer les informations de contacts d’une équipe de développeurs (adresse mail, informations bancaires, etc.). 7. L’utilisateur doit pouvoir gérer les membres de son équipe de développeurs. 8. L’utilisateur doit pouvoir transmettre la gestion et la responsabilité d’une équipe de développeurs à un utilisateur tiers. |
| *Priorisation* | 3 / 3 |
| *Complexité* | 2 / 5 |
| *Auteur* | Thibaud Alt |
|  |  |

#### Soumission de projets

|  |  |
| --- | --- |
| *Identifiant* | Epic 3 |
| *Titre* | Soumission de projets |
| *Description* | En tant que mandant, je veux pouvoir soumettre des projets et choisir une équipe de développement pour les réaliser. Pour ce faire, je peux consulter le profil de l’équipe de développeurs ainsi que les profils des différents membres de celle-ci. |
| *Tâches associées* | 1. Le mandant doit pouvoir publier un projet à réaliser. 2. Le mandant doit pouvoir modifier les informations liées à un nouveau projet (listes de fonctionnalités, délais). 3. Le mandant doit pouvoir choisir une équipe de développeurs pour réaliser le projet. 4. Le mandant doit pouvoir consulter les profils des équipes de développeurs. 5. Le mandant doit pouvoir consulter les profils des développeurs. |
| *Priorisation* | 1 / 3 |
| *Complexité* | 2 / 5 |
| *Auteur* | Thibaud Alt |
|  |  |

#### Évaluation d’une équipe de développeurs

|  |  |
| --- | --- |
| *Identifiant* | Epic 4 |
| *Titre* | Évaluation d’une équipe de développeurs |
| *Description* | En tant que mandant, je veux pouvoir évaluer une équipe de développeurs un fois un projet finalisé et livré ou abandonné. Mon évaluation est alors visible et consultable par tous les autres mandataires. |
| *Tâches associées* | 1. Le mandant doit pouvoir attribuer une note à une équipe de développeurs. 2. Le mandant doit pouvoir écrire un rapport (feed-back) sur le déroulement d’un projet. 3. Les notes et les rapports obtenus par une équipe de développeur doivent s’afficher sur le profil de celle-ci. |
| *Priorisation* | 2 / 3 |
| *Complexité* | 3 / 5 |
| *Auteur* | Thibaud Alt |
|  |  |

#### Soumission de candidature

|  |  |
| --- | --- |
| *Identifiant* | Epic 5 |
| *Titre* | Soumission de candidature |
| *Description* | En tant qu’équipe de développeurs, je veux pouvoir soumettre ma candidature, mon cahier des charges et mes coûts pour un projet proposé. |
| *Tâches associées* | 1. Tous les développeurs doivent pouvoir consulter la liste des projets ouverts à la réalisation. 2. Une équipe de développeurs (via son responsable) doit pouvoir soumettre sa candidature pour un projet proposé. 3. Une équipe de développeurs doit être notifié en cas de sélection de leur soumission pour un projet. |
| *Priorisation* | 1 / 3 |
| *Complexité* | 2 / 5 |
| *Auteur* | Thibaud Alt |
|  |  |

#### Dépôt des livrables

|  |  |
| --- | --- |
| *Identifiant* | Epic 6 |
| *Titre* | Dépôt des livrables |
| *Description* | En tant qu’équipe de développeurs, je veux pouvoir téléverser des documents, des livrables et des informations tout au long du projet. Ceux-ci sont alors visible pour tous les membre de l’équipe ainsi que pour le mandant. |
| *Tâches associées* | 1. Une équipe de développeurs (via son responsable) doit pouvoir téléverser des documents électroniques de différents formats (« .*pdf* », « *.docx* », « *.xlxs* », etc.). 2. Une équipe de développeurs (via son responsable) doit pouvoir téléverser des archives « *.zip* » de livrables du projet. 3. Le mandant doit être notifié lorsqu’un téléversement est effectué. 4. Le mandant doit pouvoir consulter tous les documents et livrables liés au projet. |
| *Priorisation* | 3 / 3 |
| *Complexité* | 4 / 5 |
| *Auteur* | Thibaud Alt |
|  |  |

#### Classement des équipes et des développeurs

|  |  |
| --- | --- |
| *Identifiant* | Epic 7 |
| *Titre* | Classement des équipes et des développeurs |
| *Description* | En tant que visiteur, je peux consulter le classement mensuel, annuel et « *de tous les temps* » des équipes de développeurs. En tant que visiteur, je peux également consulter le classement d’un développeur. |
| *Tâches associées* | 1. Un visiteur doit pouvoir consulter une page de classement présentant les équipes de développeurs et comportant des options de tris et de filtres. 2. Un visiteur doit pouvoir consulter le classement d’un développeur via sa page de profil. |
| *Priorisation* | 1 / 3 |
| *Complexité* | 4 / 5 |
| *Auteur* | Thibaud Alt |
|  |  |

#### Espace de travail

|  |  |
| --- | --- |
| *Identifiant* | Epic 8 |
| *Titre* | Espace de travail |
| *Description* | En tant qu’utilisateur de la plateforme, je peux accéder à mon espace de travail incluant un forum par projet en cours. Ces forums me permettent d’accéder à toutes les informations nécessaires au bon déroulement des projets. |
| *Tâches associées* | 1. Un acteur du projet doit pouvoir ajouter différents types de fichiers (texte, images, etc.) 2. Un acteur du projet doit pouvoir consulter toutes les ressources liées à un projet. 3. Un acteur du projet doit pouvoir transmettre une information à tous les participants d’un projet. 4. Le responsable d’une équipe de développeurs doit pouvoir mettre à jour régulièrement un livrable du code du projet, ce livrable inclut une chronologie des versions. 5. Le mandant doit pouvoir télécharger en tout temps les différents livrables. |
| *Priorisation* | 2 / 3 |
| *Complexité* | 3 / 5 |
| *Auteur* | Thibaud Alt |
|  |  |

#### Discussions instantanées

|  |  |
| --- | --- |
| *Identifiant* | Epic 9 |
| *Titre* | Discussions instantanés |
| *Description* | En tant qu’utilisateur de la plateforme, je peux accéder via mon espace de travail à un système de discussions instantanées. |
| *Tâches associées* | 1. Un acteur du projet doit pouvoir discuter, sous forme textuelle, avec un membre du projet. 2. Un acteur du projet doit pouvoir discuter, sous forme textuelle, avec un groupe de membres du projet. 3. Si nécessaire, un utilisateur peut activer un système de traduction instantané des messages dans sa langue préférée. |
| *Priorisation* | 3 / 3 |
| *Complexité* | 4 / 5 |
| *Auteur* | Thibaud Alt |
|  |  |

## Besoins non-fonctionnels

### Contraintes dû à l’environnement

Ce projet étant nouveau et non lié à environnement précis, il ne dispose pas de contraintes techniques définis. Il devra cependant pouvoir s’inscrire dans un portefeuille de projets web existants et de ce fait devra suivre les *bonnes pratiques* de développement actuelles.

Plus tard, il se pourrait que d’autres développeurs soient amenés à faire évoluer ce projet, c’est pourquoi celui-ci devra être correctement documenté et devra être développé avec des frameworks connus et maitrisé par un grand nombre de développeurs.

### Besoins de performance, d’ergonomie et de fiabilité

#### Interface et expérience utilisateur

L’interface utilisateur devra respecter une charte graphique et des maquettes définis. L’application web devra être intuitive, l’expérience utilisateur devra être fluide et l’ergonomie agréable.

#### Charges et ressources

L’application ne devra pas, du moins dans sa première version, supporter un taux de charge excessif. Toutefois, elle devra être pensée et développé de tel sorte à pouvoir l’adapter à ces points dans des versions postérieurs.

## Extensions

### « *Si temps le permet* »

#### Multilinguisme

L’application étant dans sa première version uniquement disponible en anglais, il serait judicieux de la traduire dans d’autres langues permettant ainsi d’attaquer différents marchés. Dans un premier temps, et pour le marché Suisse, l’application pourra être traduite en Allemand et en Français.

#### Adaptation « responsive »

L’application web étant fonctionnelle sur un ordinateur bureau classique, il serait fort agréable pour l’utilisateur d’également disposer d’une interface sur ces appareils mobiles. Cette interface, éventuellement réduite, devra donc être agréablement utilisable et fonctionnelle sur des téléphones mobiles récents et sur des tablettes récentes.

#### Inscription et connexion via des services tiers

Aujourd’hui nous possédons tous de nombreux comptes sur internet et il n’est pas toujours facile de se souvenir quelle combinaison nom d’utilisateur/mot de passe nous avons définis. De plus, les inscriptions à un service web sont souvent des étapes lentes et contraignantes qui vont à l’encontre de l’expérience utilisateur. Partant de ce constat, il serait judicieux d’ajouter des services tiers (Google, Apple…) comme moyen de connexion à la plateforme web.

### Dans des versions futures

#### Solutions de paiement

Dans une version future, il serait intéressant d’étudier puis d’implémenter différentes solutions de paiement à la plateforme web.

#### Intégration de professionnels externes

Une des valeurs ajoutées de la plateforme, est la possibilité de faire appel à un professionnels externe pour traiter un point spécifique du projet. Par exemple le recours à un médiateur en cas de conflit entre certains acteurs du projet, l’appel à un consultant ou à un expert pour prendre une décision stratégique, le suivi par un coach en cas de baisse de motivation, etc. Ces acteurs externes seraient des partenaires indépendant, validés par un système de sélection et leurs services seraient proposés aux mandants et/ou au développeurs au travers de la plateforme.

#### Système de sous-traitance

Un système de sous-traitance de tâches simples et répétitives permettant de répondre rapidement à un besoin ponctuel pourra être intégré directement à la plateforme. Celui-ci pourra être utilisé par exemple pour la saisie d’articles dans un magasin de vente en ligne, le nettoyage d’une base de données, la recherche d’images d’illustrations, etc. L’idée ici serait de trouver différents partenaires effectuant ce type de services et de les intégrer entièrement à la plateforme. De ce fait l’utilisation de ce système par les développeurs seraient très simple et ils n’auraient pas à quitter la plateforme.

#### Espace de connaissances

Un espace de connaissances mettant à disposition des ressources, des cours, des tutoriels, des outils, etc. apportant des notions théoriques ainsi que différents savoirs nécessaires liés à la gestion de projets. Cet espace se construirait au fur et à mesure des projets via les connaissances acquises par les différents acteurs et sur le principe de « l’open source ». De plus, il pourrait être envisageable de mandater un expert en gestion de projets qui réaliserait une série de tutoriels vidéo vulgariser, accessibles à tous et intégrant les différentes spécificités de la plateforme.

#### Vulgarisation et reformulation

Dans une version future, une intelligence artificielle pourra être interpelée en cas d’incompréhension de vocabulaire entre mandants et développeur lors de discussions textuelles instantanées. Cette intelligence artificielle pourra reformuler certaines phrases, vulgariser certains termes et apporter des éléments supplémentaires nécessaires à la bonne compréhension du message.

# Analyse

## Étude de marchée

### Cibles

Ce projet s’adresse principalement à deux types de profils distincts puisqu’il a pour but de les mettre en relation :

1. **Les PME et grandes entreprises**

Les clients pouvant proposer des projets sur la plateforme devraient être des PME ou des grandes entreprises voulant développer un projet spécifique dont la liste de fonctionnalités et les délais sont définit précisément. Au lieu d’engager du personnel pour un projet ou de mandater une entreprise externe, ces entreprises peuvent proposer leur projet sur la plateforme.

1. **Les développeurs informatiques**

De l’autre côté, le projet a besoin de développeurs informatiques indépendant ou employés désirant développer un ou plusieurs projets en équipe. Ces développeurs doivent maitriser une ou plusieurs technologies de développements, être autonomes et savoir travailler en équipe.

## Graphisme et ergonomie

### Charte graphique

Ajoutez avec votre cahier des charges tous les documents concernant l’identité visuelle de votre entreprise (les codes couleurs, le logo et ces variations, les règles de conception etc.).

* Précisez vos exigences concernant les couleurs et les effets de votre site ?
* Quelles sont les polices à utiliser ?
* Quels sont les éléments à conserver ?

Ajoutez également à votre cahier des charges des exemples de sites avec un commentaire, les choses que vous aimez bien sur ces sites (éléments, animation, couleurs, architecture d’informations, fonctionnalités etc.)

### Wireframe et maquettage

Ajoutez à votre cahier des charges toutes les maquettes et prototypes de vos idées concernant le résultat que vous aimeriez obtenir avec ce site Internet.

Les maquettes contiennent :

* L’arborescence du site (les rubriques, sous rubriques, type d’URL’s etc…)

L’architecture des informations concernant l’ensemble de ces pages

## Risques

# Modélisation

# Conception

# Planification

# Réalisation

# Tests

# Conclusion