**Groupe n°14:**

CLERC Louis

DAIKH Nassim

DELHOMME Baptiste

GENTIL Driss

HIGGINS Alec

KADIRI Saad

**LINGO PEER**

**by lingo team**



# **Sommaire**

[**Sommaire**](#_z7w1f9n5yxzu) **2**

[**1) Cadrage**](#_s6imkqkjg1qr) **3**

[1.1) Définition du contexte](#_ja63skiga77e) 3

[1.2) Analyse du terrain et de l’existant](#_z624bolbl1p1) 4

[1.3) Définition des objectifs](#_jyi5689thtda) 9

[1.4) Contraintes et risques (mitigation)](#_lweipqi6g6fg) 9

[1.5) Suivi des risques](#_wxs99pmnf77p) 12

[**2) Expression des besoins**](#_e6rw832ox1vn) **13**

[2.1) Introduction](#_caewuxk5hbt9) 13

[2.2) Identification et priorisation des besoins fonctionnels](#_r40qtcdrdu6g) 14

[2.3) Identification et priorisation des besoins non-fonctionnels](#_vxd1sx8k3quz) 15

[2.4) Cohérence besoins - Modélisation](#_hrureq2q2ih) 16

[2.5) Spécification - Modélisation](#_8v446ews2tpa) 19

[2.6) Mise en avant des choix de critères qualité logicielle](#_7mo5vo35iwzi) 21

[2.7) Mise en avant des critères ergonomiques](#_e17eyccyzbf8) 22

[**3) Mise en oeuvre de la solution**](#_2btr05ih3yyz) **24**

[3.1) Le choix et les justifications des technologies utilisées](#_9gk5egl9b3sq) 24

[3.2) Gestion du travail](#_pti1bbyamw1v) 26

[**Annexe:**](#_he4ykfdmsps0) **28**

**https://lingo-peer.netlify.app/**



# 

# 

# 1) Cadrage

## 1.1) Définition du contexte

Suite aux événements liés au Covid-19 qui nous ont affectés, le monde s’est retrouvé coupé de toutes les interactions physiques qu’il connaissait auparavant. Mais on a aussi observé à quel point internet est complet en termes d’applications pour toutes sortes de choses. Cependant, le fait de devoir rester enfermé chez soi a entravé la communication et la sociabilité entre tous. De plus, il est très compliqué de voyager dans le contexte actuel. Nous en sommes venus à la conclusion que l’aspect international et sociable d’internet pouvait bénéficier d’approfondissements. Pour pallier le manque d’exotisme dans la vie sociale de chacun ou pour bien encadrer les échanges avec des personnes de partout dans le monde, et surtout pour passer le temps de manière productive, nous allons créer une application d'échange dans toutes les langues. Cela nous a conduit à nous poser la question suivante :

***“ Comment continuer à pratiquer ou apprendre des langues tout en ayant un contact social pendant le confinement ? “***

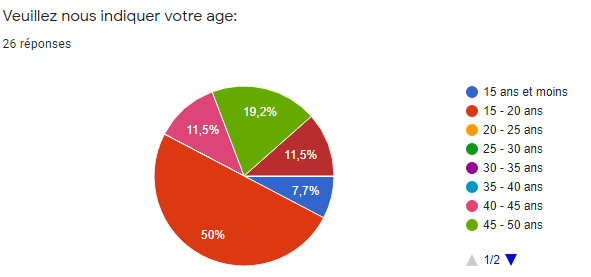
L’application devrait permettre de converser avec des inconnus mais aussi d’avoir des sujets de discussion. L’utilisateur pourra aussi ajouter des participants à sa liste d’amis si l’échange se passe bien et que l’on souhaite réitérer l’expérience. Enfin, il est possible de choisir la manière de discuter pour être plus à l’aise lors de la conversation. Idéalement elle sera accessible depuis une grande quantité d’appareil afin d’être facile d’atteinte pour tous.

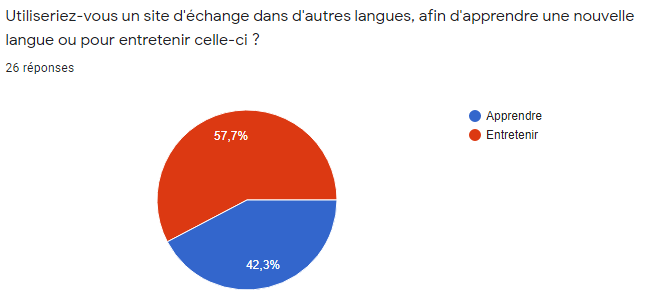
C’est avec ces paramètres que l'équipe est parvenue à trouver un nom pour l’application : “lingo peer”. En effet, ‘lingo’ et ‘peer’ signifient respectivement ‘langue’ et ‘pair’ en latin : ce qui souligne l’interaction entre les utilisateurs. Le but de cette nouvelle application est donc de permettre aux participants de pouvoir améliorer ou conserver leur niveau linguistique en utilisant différents formats de communication.

## 1.2) Analyse du terrain et de l’existant

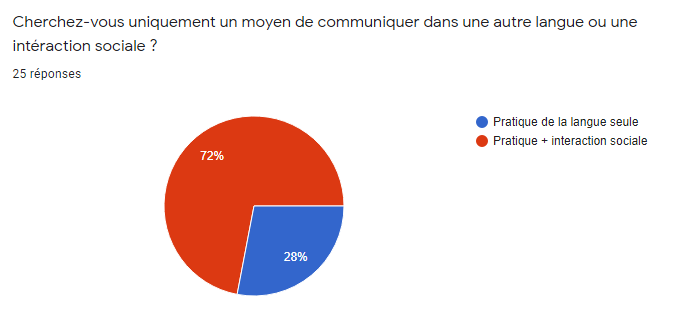
Dans un premier temps, on abordera l’enquête qui a été réalisée afin de savoir ce qui plairait ou non aux utilisateurs : ce qui constitue notre analyse du terrain. Puis, on regardera les différents services déjà proposés afin d’en tirer parti pour la réalisation de notre projet.

L'analyse du terrain passe par des enquêtes, c’est pour cela que l’on a décidé de faire un sondage sur l’intérêt du public envers le projet. Ce dernier permet de déterminer que les utilisateurs cibles de notre application seraient les jeunes entre 15 et 30 ans comme le démontre la *Figure 1*. Il apparaît que cette tranche d’âge est ressortie car ce sont principalement les lycéens et les étudiants qui veulent maintenir leur niveau linguistique. Parmi ces étudiants, on trouve des natifs, non-francophones, qui sont restés en France durant le confinement et qui souhaitent conserver leur niveau dans leur langue natale, c’est ce que montre la *Figure 2*. On peut aussi constater que les plus jeunes attendent de ce service une vraie interaction sociale d’après la *Figure 3*. Bien que cet échantillon soit plutôt restreint, il sera tout de même suffisant pour avoir une idée globale, et donc il est possible d'utiliser cette enquête pour continuer notre projet.

*Figure 1:* Age des participants



*Figure 2:* L'intérêt des participants

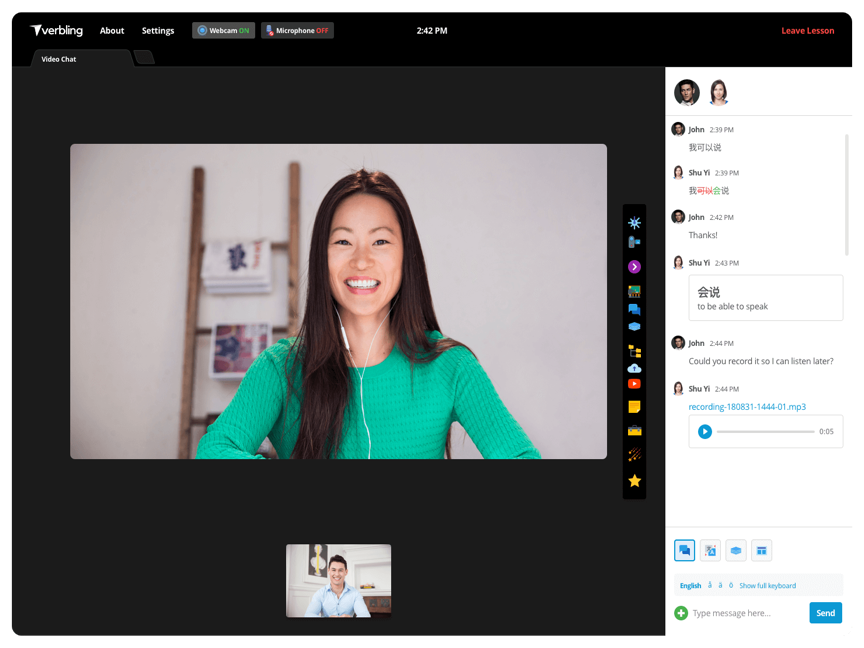


*Figure 3:* Les envies des participants

L'analyse de l'existant permet de se situer par rapport à la concurrence et les applications d'apprentissage de langues sont nombreuses. Les plus ressemblantes sont :

● *Verbling :* c’est une application déjà bien ancrée dans l’apprentissage de langue grâce à plusieurs avantages : l’association avec un professeur vérifié, une variété de formations différentes, la possibilité de demander un programme personnalisé et des outils d’échange efficaces comme des chatbox ou des appels vidéos. Disponible sur Android et IOS en plus du navigateur web, beaucoup de langues sont disponibles (plus de 50) et donc il y a une communauté large et accueillante. Néanmoins le service est payant (environ 20$ par cours), ce qui le limite à une catégorie d’utilisateurs définie. Il est de plus en plus difficile de réserver une place, ce qui montre bien à quel point le site ne peut satisfaire tous ses potentiels clients. Malgré son utilisation, des imperfections telles que la chatbox et le prix freinent plusieurs participants.

|  |  |
| --- | --- |
| Points Forts | Points Faibles |
| Professeur vérifié | Service payant |
| De nombreuses formations | Certains outils dysfonctionnent |
| Outils d’échange efficace |  |
| Disponible sur toutes les plateformes |  |



*Figure 4 :* *Verbling* - Appel vidéo (à gauche) avec chatbox (à droite)

● *Jaaser.com :* cette application web n’est pas encore très connue mais elle a différents points forts. En effet c’est un système de chat efficace puisque le but principal est d’avoir un correspondant ce qui permet un apprentissage moins scolaire. De plus, l’application possède une communauté récente dont des personnes pratiquant l’espéranto (une langue sous représentée) et est complètement gratuite et disponible sur mobile. Mais aucun appel (vidéo ou audio) n’est proposé, et l’interface de la correspondance écrite n’est pas ergonomique.

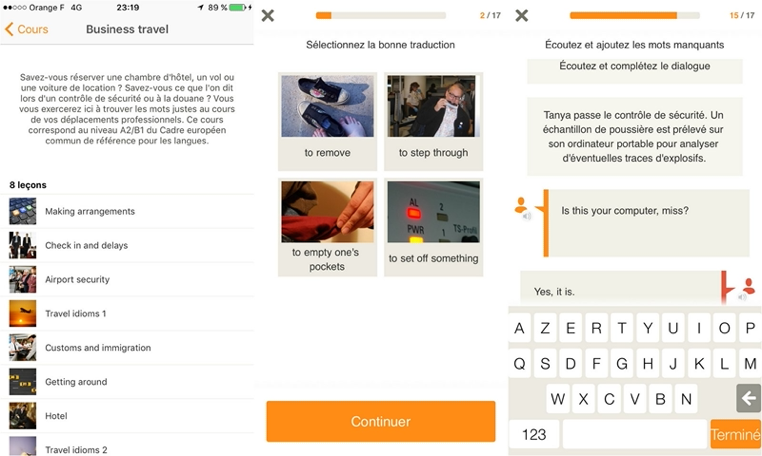
|  |  |
| --- | --- |
| Points Forts | Points Faibles |
| Chat écrit efficace | Pas d’appel vidéo ni audio |
| Pratique de l’espéranto | Interface non ergonomique |
| Application gratuite |  |



*Figure 5:* *Jaaser* - Page d’accueil

● *Babel :* c’est l’application référence en matière d’apprentissage linguistique, proposant ses services sur mobiles et sur navigateur. Voici ces avantages : les cours sont créés par des professionnels en langue, les méthodes d'apprentissage sont efficaces (il faut environ 2 mois pour apprendre une langue), les outils mis en place pour l'apprentissage comprennent des dialogues pratiques, des compréhensions orales, une reconnaissance vocale et un outil de révision. De plus l’interface est très épurée et ergonomique. Malgré tout, différents inconvénients sont à noter : une approche trop scolaire, un entraînement à l'oral réalisable uniquement seul et un service payant (abonnement d’environ 6 €).

|  |  |
| --- | --- |
| Points Forts | Points Faibles |
| Cours créés par des professionnels | Service payant |
| Méthodes d’apprentissage efficaces | Approche trop scolaire |
| De nombreux outils | Entraînement oral défaillant |
| Interface épurée |  |



*Figure 6 :* *Babel* - (De gauche à droite) Cours à base de texte, Exercices visuels et Exercices écrits

Ainsi cette analyse de l’existant permet de saisir les avantages et inconvénients des différents sites qui existent déjà. Le but ici est de réussir à faire un bon compromis pour rassembler un maximum des points forts. Il est alors légitime de mettre en avant une interface épurée, professionnelle et fonctionnelle tout en permettant aux utilisateurs d’échanger de la manière souhaitée. Enfin, il est nécessaire de créer une application gratuite qui regroupe plusieurs langues.

## 1.3) Définition des objectifs

L’objectif principal est de réaliser une une application web c’est a dire un site dynamique qui puisse répondre

Objectifs techniques:

Pour la réalisation de l’application, selon l’analyse du terrain et de l'existant, voici les caractéristiques attendues :

* permettre des conversations entre utilisateurs de manière écrite, orale ou audiovisuelle
* pouvoir répartir les profils des utilisateurs dans des groupes uniques par le biais de leur niveau de langue, la ou les langue choisie(s) et les centres d’intérêts.
* mettre en place une roulette, permettant un choix aléatoire en fonction des affinités des participants
* posséder une interface homme-machine propre, épurée et agréable à l'œil.
* permettre des connexions dites “Peer to peer”, c’est-à-dire une connexion pair à pair
* stocker les données des utilisateurs dans une base de données afin de pouvoir s’en servir

## 

## 

## 1.4) Contraintes et risques (mitigation)

Pour réussir un projet, il faut évaluer en amont les contraintes et les risques.

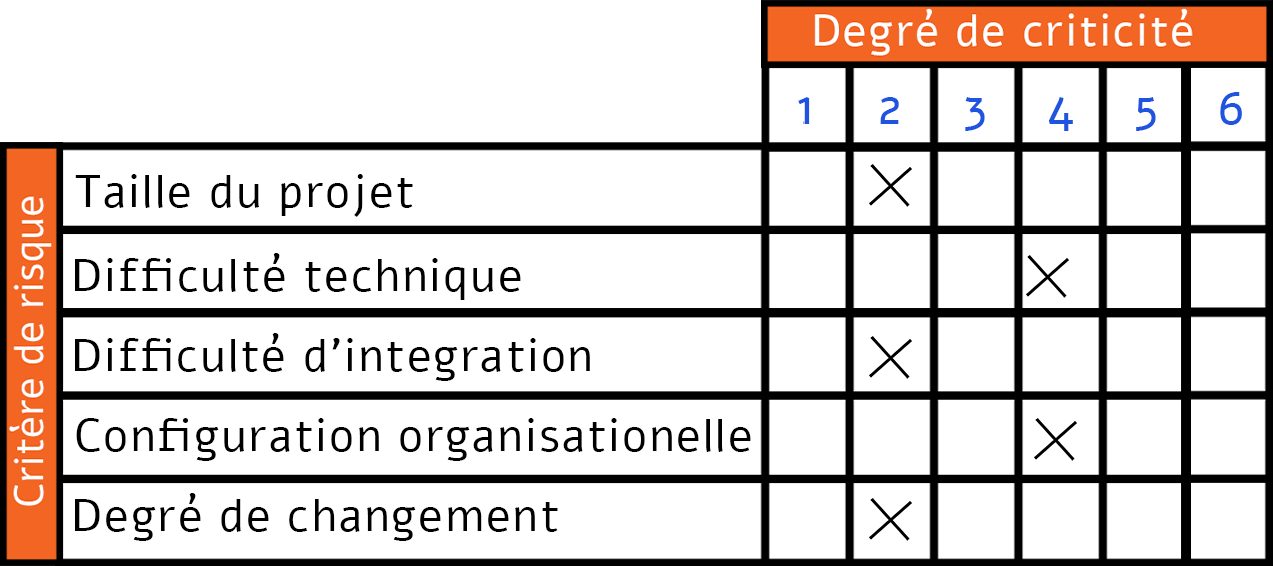
Les contraintes les plus importantes sont les suivantes :

* De manière générale, le respect de la vie privée des utilisateurs est primordial, par conséquent il est important d’évaluer la protection de toutes les données. Ce qui constitue une contrainte légale à respecter.
* Une contrainte de temps nous est imposée dans ce projet, avec 3 itérations et un rendu attendu à la fin de chacune :
* Tout d'abord à *l’itération 1* nous devions trouver une idée originale et qui colle au thème tout en recherchant des solutions pour concevoir le projet.
* Ensuite à *l’itération 2* nous devions confirmer nos idées, choisir les méthodes techniques qui seront utilisées et faire la conception de l’application pour commencer à la réaliser.
* Et enfin à *l’itération 3* nous devons finir de réaliser notre application dans le détail, la peaufiner, commenter le code et se débarrasser des erreurs.
* On est également confronté à une contrainte de moyens. Cela se traduit par les nouveaux outils et langages que nous allons utiliser dans ce projet, et que nous ne maîtrisons pas encore. Nous devons donc minimiser ce problème avec de l’entraînement pour utiliser ces nouveaux outils et langages.

En ce qui concerne les risques il y en a plusieurs facilement identifiable :

* R1 = **La taille du projet** : si celui-ci est trop grand il ne sera pas réalisable dans le temps imparti .
* R2 = **La difficulté technique** : elle est due au manque de compétences de l'équipe à cause des technologies utilisées qui ne sont pas connues par tous les membres du groupe.
* R3 = **La difficulté d'intégration** : elleréside dans le fait qu’il faudra faire appel à des acteurs extérieurs afin de déployer l’application en ligne.
* R4 = Les risques liés à **la configuration organisationnelle** : ils proviennent des différents conflits d'intérêts pouvant avoir lieu entre les différentes parties prenantes.
* R5 = **Le degré de changement** : c’est-à-dire à quel point le produit final à changé et à quel point il correspond aux attentes des utilisateurs.

Pour schématiser les risques on fait la matrice de criticité suivante :



*Figure 7:* Matrice de criticité des risques

Afin de pouvoir pallier aux risques qui mettent à mal la pérennité du projet, il est nécessaire de réaliser un plan de mitigation afin de convenir de mesures dans le cas où les risques cités apparaissent.

Voici le plan de mitigation mise en place pour chaque risque :

|  |  |
| --- | --- |
| Risques | Mitigation |
| R1:taille du projet | Redécoupage du projet en sous projets |
| R2:difficulté technique | Recherche:Quête d’informations complémentaires |
| R3:difficulté d'intégration | Réserves:allocation de ressources supplémentaires |
| R4:configuration organisationnelle | Privilégier le développement évolutif et incrémental |
| R5:degré de changement | Réduire l'incertitude en modifiant les caractéristiques du projet |

## 

## 

## 

## 1.5) Suivi des risques

Dans la partie précédente, nous avons évoqué les différents risques que nous avons envisagé et les plans. Le suivi des risques sert à faire l’état des lieux en comparant les prévisions voulues à ce qui s'est passé. Il nous permet aussi de voir si de nouveaux risques sont apparus.

R1 :En effet le projet à été surdimensionné par rapport au risque R6.

R2 :Il y a avait plusieurs choix techniques qui ont été changés notamment la manière de s’authentifier et cela à été rendu possible grâce à la solution qui était de refaire des recherches.

R3 : Il fallait que l’on envisage une solution pour pouvoir déployer étant donné que nous ne pouvions héberger la solution sur un serveur local et nous avons donc fait appel à des plateformes spécialisées dans le déploiement d'applications.

R4: Ce risque n’est pas apparu étant que la méthode de développement incrémental a permis de prévenir ce risque.

R5:Les caractéristiques initiales du projet n’ont pas changé donc ce risque a été évité.

Un nouveau risque qui n'a pas été prévu est le risque R6.

R6:L'instabilité de l'équipe de projet, en effet un membre à eu une situation où il était dans l’incapacité de travailler .

Il a fallu attribuer le travail on a donc délégué le travail qui lui était attribué qui a forcément eu un impact sur la réalisation du travail.

Le problème étant que l’apparition de ce risque à eu un impact critique sur le projet.

Néanmoins, le suivi des risques à été utile au bon déroulement du projet et il a permis une réponse rapide aux risques qui sont apparus grâce au plan de mitigation .

# 

# 2) Expression des besoins

## 2.1) Introduction

Afin d'établir une cohérence entre les attentes des utilisateurs et les moyens mis en œuvre pour combler ses attentes, il est nécessaire de formaliser l’expression des besoins.

C’est à ces mêmes besoins qu’il faut faire correspondre une réponse technique qui soit étudiée au préalable afin de savoir ce qu’il est possible ou non de faire.

C’est dans cette partie que seront détaillés les besoins, ils seront également prioriser en fonction de leur pertinence mais aussi des critères de qualités logicielles qui rendront compte et permettront d’évaluer le rendu ainsi que la qualité attendue de l’application web

## 2.2) Identification et priorisation des besoins fonctionnels

|  |  |  |
| --- | --- | --- |
| Priorisation des besoins fonctionnels | Nom des besoins fonctionnels | Explications des besoins  fonctionnels |
| 1 | Permettre à l’utilisateur de saisir toutes les informations nécessaires et les stocker dans une base de données | Les informations seront : nom, prénom, âge, adresse mail, langue parlée, langue que l'on souhaite apprendre.  Pour le nom, le prénom, et l’adresse mail : on mettra en place des champs textes.  Pour l’âge, langue parlée, langue que l’on souhaite apprendre et thèmes : on mettra en place des listes déroulantes. |
| 2 | Mettre en relation deux personnes en faisant en sorte que leurs centres d’intérêts correspondent | Les participants devront avoir choisi entre apprendre, enseigner ou discuter.  Le binôme sera alors composé de : apprentissage/enseignement ou  discuter/discuter. |
| 3 | Faire en sorte que les utilisateurs puissent communiquer en chat écrit | Pouvoir avoir une fenêtre de discussion et envoyer des messages |
| 4 | Mettre en place un chat audio | Au préalable l'accord de l’utilisateur sera demandé. |
| 5 | Mettre en place une connexion vidéo | Au préalable l'accord de l’utilisateur sera demandé. |
| 6 | Permettre aux utilisateurs de s’ajouter en amis pour garder contact | Mise en place d'un bouton « Ajouter en ami » pendant une discussion.  On pourra alors retrouver cette personne dans une liste d'amis sur son profil. |
| 7 | Mettre en place une fonction qui permettra à l’utilisateur de changer rapidement de conversation | Mise en place d'un bouton « changer d'utilisateur » pendant une discussion. |
| 8 | Mettre en place un fil de partage | C’est un endroit où les utilisateurs peuvent partager des publications, mettre des j’aime et des commentaires. |
| 9 | Paramétrer son compte | Chaque utilisateur aura la possibilité de modifier son profil : avatar, mot de passe, adresse mail, ainsi que les langues et les centres d’intérêts. |

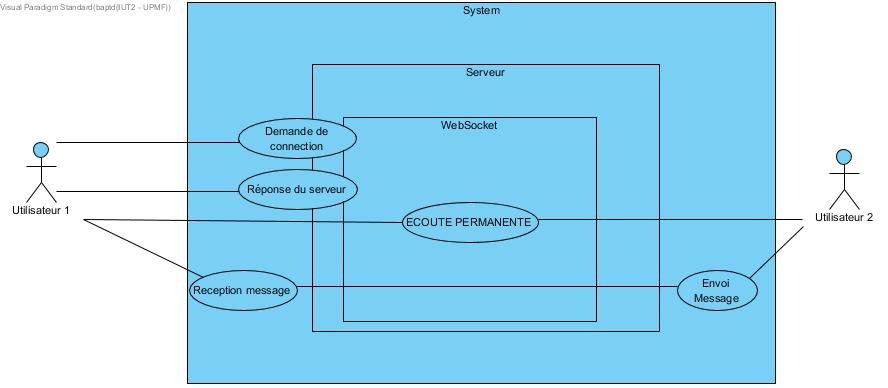
## 2.3) Identification et priorisation des besoins non-fonctionnels

|  |  |  |
| --- | --- | --- |
| Priorisation des besoins non-fonctionnels | Nom des besoins non-fonctionnels | Explications des besoins non-fonctionnels |
| 1 | Fiabilité de l’application web. | Évaluer l'aptitude d'un système à maintenir son niveau de service : tolérance aux pannes, conditions de remise en service, maturité… |
| 2 | Facilité d'utilisation. | Caractériser l'effort nécessaire à un utilisateur potentiel pour utiliser le système : facilité de compréhension, d'apprentissage, d'exploitation ou encore pouvoir d'attraction. |
| 3 | Critère ergonomique. | Respecter les 8 critères ergonomiques de “Bastien & Scapin” (détaillé en partie 2.3) |
| 4 | Maintenabilité  de l’application web. | Évaluer les possibilités pour faire évoluer le système dans le cas de nouveaux besoins |

## 

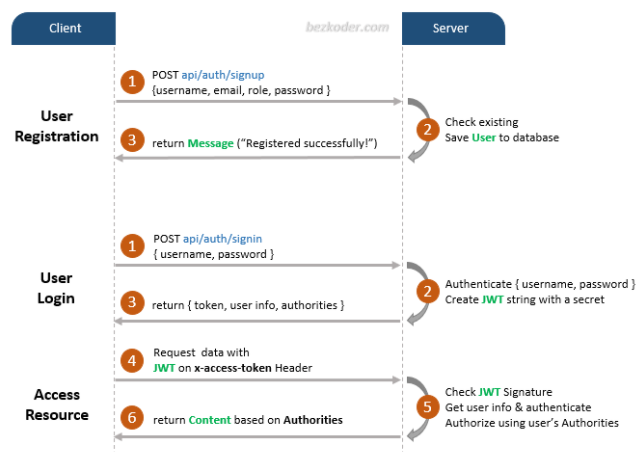
## 2.4) Cohérence besoins - Modélisation

Cette partie explicite la manière dont les besoins précédents sont modélisés au sein de notre application :



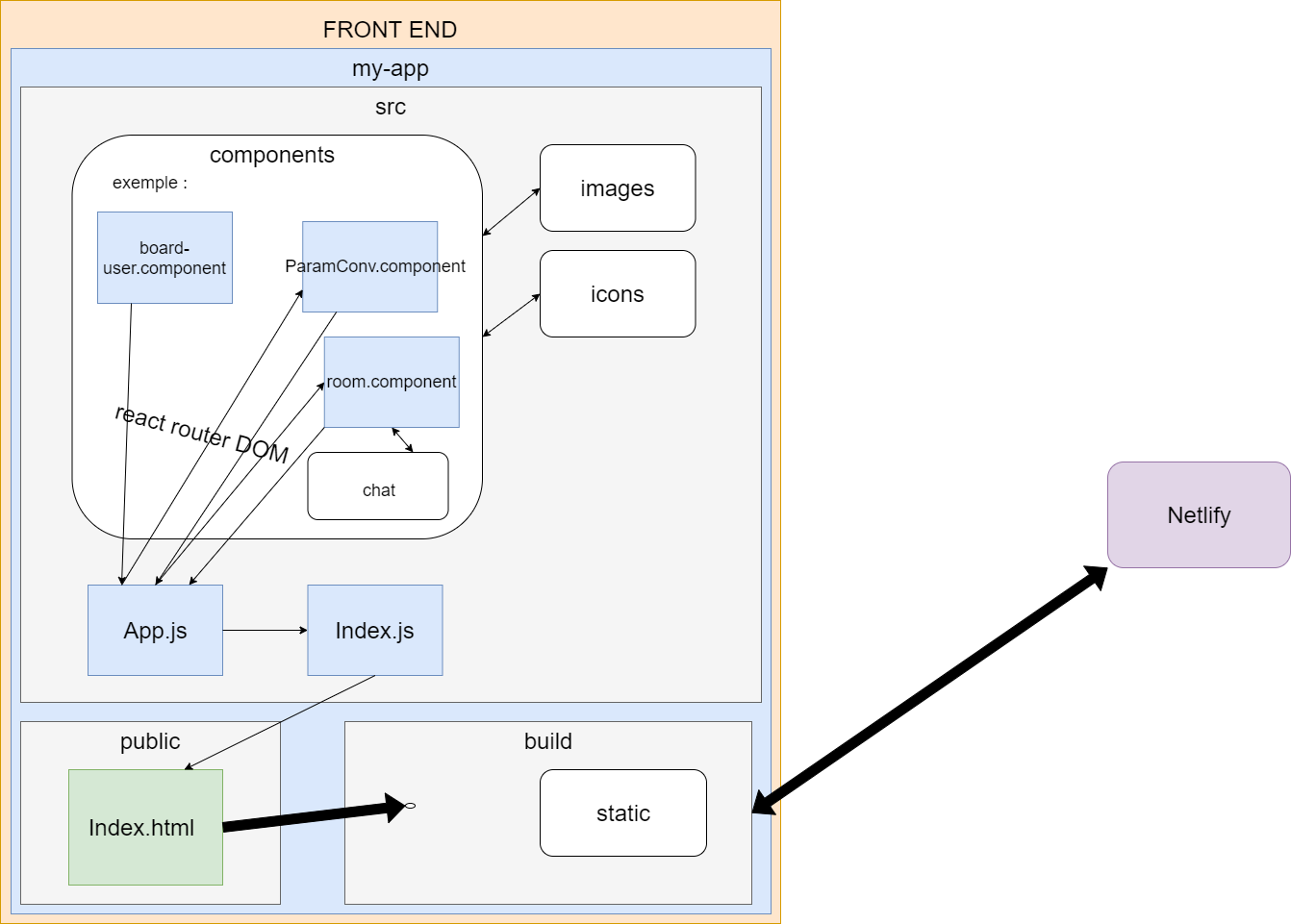
*Figure 8* : Cas d’utilisation 1 (Conversation)

Une fois la connexion entre le client et le serveur effectuée, ils se mettent tous deux en écoute de l’autre. Ensuite, l’utilisateur va déclencher l'événement de l’envoi d’un message, et dès que le serveur le reçoit, il le transmet à l’autre utilisateur connecté. Tout est déjà planifié des deux côtés, notamment l’affichage des messages sur la vue correspondante, ce qui permet une communication instantanée entre les utilisateurs.



*Figure 9 :* Diagramme de séquence de création de compte et de connection

Afin de créer des infos, un client fait une demande au serveur. Si ces infos concordent avec les existantes (donc si un nouvel utilisateur est bel et bien nouveau) alors le serveur accède à la demande. Pour utiliser une ressource dans la base de données, le client fait encore appel au serveur. Dans un premier temps, le serveur vérifie l’authenticité de chaque requête. Ensuite, si celle-ci passe la sécurité, le serveur fait alors le lien entre la requête et la ressource cherchée, puis la renvoie au client.



On peut voir les différents répertoires contenus dans le front-end et comment ils interagissent entre eux, dans src le répertoire component contient quasi toutes les vues, pour se déplacer d’une vue à une autre il faut passer par App.js “le gestionnaire” qui a accès à toutes les routes. Ensuite index.js fait le lien entre le “gestionnaire” et index.html “l’organisateur” de l’affichage. Pour finir on utilise un build “une construction plus compacte” de la partie Front-end qui vérifie aussi les différentes erreurs, elle sera mis sur Netlify le service utilisé pour l’affichage sur le web.

*Figure 10 :* Diagramme expliquant la mise en place de la partie front end.

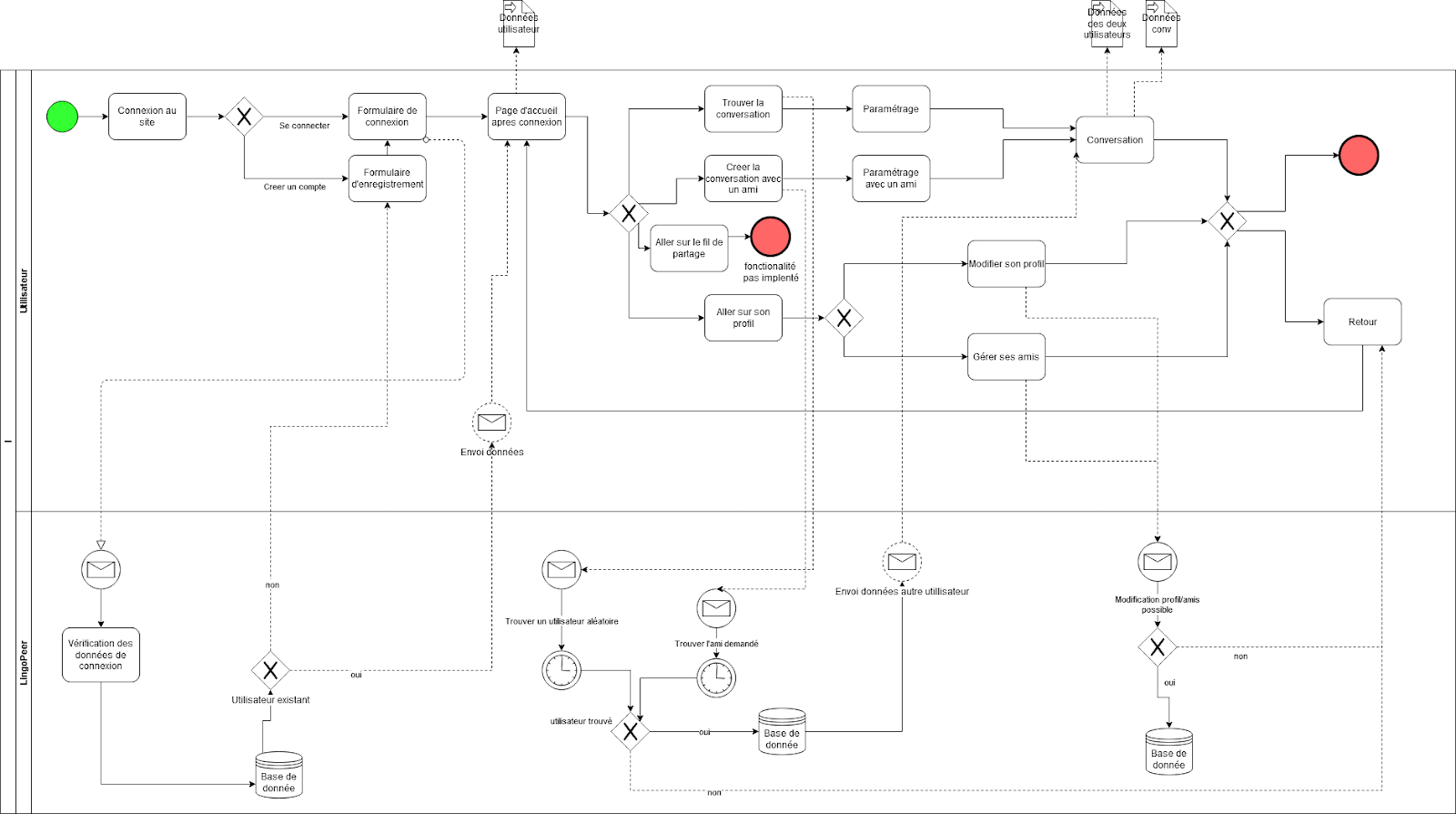


Figure 11 : BPMN lié au scénario du point de vue de l’utilisateur

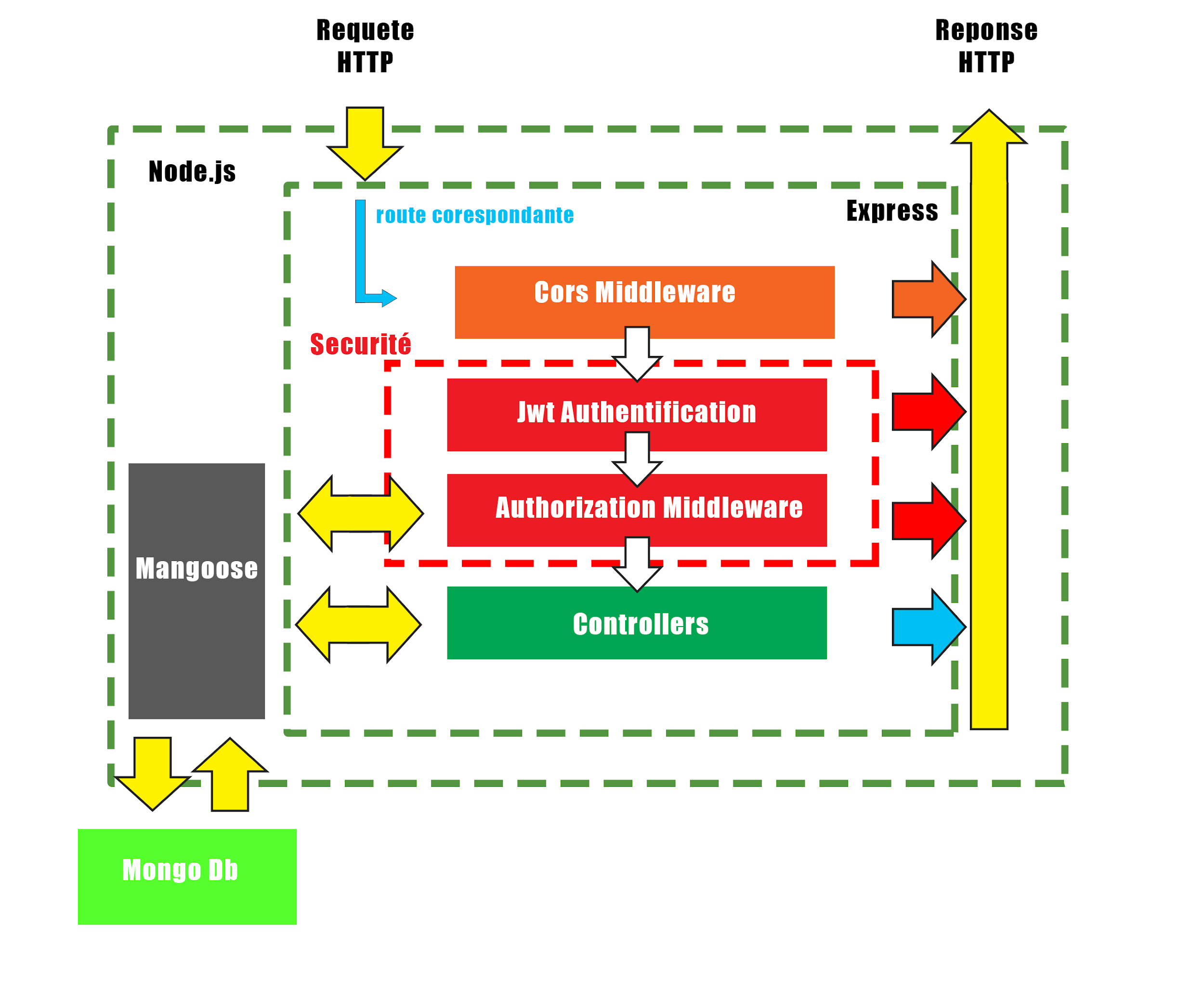
Le scénario est disponible en annexe.

## 

## 2.5) Spécification - Modélisation

Dans cette partie, nous nous attarderons à expliquer les interactions entre les différents modules de Node.js et leur fonction au sein de l’application web.

Tout d’abord voyons ce qui se produit au sein de l’application lorsqu’un utilisateur cherchera à s'enregistrer ou de s'authentifier : il fera un appel à l'api¹ .



*Figure 12 :* Schéma du fonctionnement du système d’authentification

api¹: c’est l’interface avec laquelle le programme interagit et que l’utilisateur ne voit pas.

Via les routes Express, les requêtes HTTP qui correspondent à une route existante seront vérifiées par le middleware CORS avant d'arriver à la couche de sécurité.

La **couche de sécurité** comprend :

* JWT Authentication Middleware : une vérification de l'inscription et du jeton
* Authorization Middleware : une vérification des rôles de l'utilisateur dans la base de données.Un message d'erreur sera envoyé en réponse HTTP au client lorsque les middlewares lancent une erreur.

Enfin, les contrôleurs interagissent avec la base de données MongoDB via le module Mongoose et envoient une réponse HTTP (jeton, informations sur l'utilisateur, données basées sur les rôles...) au client.

Pour la communication en temps réel, c’est un peu différent. Le fonctionnement des WebSockets est le suivant: une fois la connexion entre le client et le serveur établie, les deux communiquent sans délai avec l’un l’autre. Cette connexion s’effectue par le biais d’une requête HTTP de type UPGRADE. Une fois que celle-ci est mise en place, les “messages” entre le client et le serveur sont de nature TCP, ce qui permet une transmission de messages bi-directionnels.

## 

## 

## 

## 2.6) Mise en avant des choix de critères qualité logicielle

Les critères qualité logicielle représentent un ensemble d'attributs permettant de vérifier si le logiciel répond aux besoins fonctionnels exprimés. On y retrouve notamment les notions de pertinence, d'exactitude, d'interopérabilité, de sécurité et de conformité.

* Disponibilité : On a utilisé les services “Heroku” pour déployer la base de données et “Netlify” pour la partie affichage, une fois que ces deux parties sont déployées ensemble, tant que l’on enlève pas notre application de ses services elle restera sur le web. Sous réserve qu'aucun problème n’apparaisse.
* Flexibilité : Avec MongoDB, on a accès à l'entièreté des données de celle-ci que l’on peut modifier, supprimer ou même en ajouter. Concernant les fonctionnalités du site, comme nous utilisons git il suffit de build la dernière version disponible de notre projet et de la mettre soit dans “Heroku” pour le back-end ou dans “Netlify” pour le front-end. Donc on est assez flexible.
* Contrôle des accès : La fonctionnalité du “Login” et du “Sign up” sont tous deux protégés contre les accès non autorisés. Exemple : Pour se connecter l’utilisateur doit exister dans la base de donnée et pour créer un compte le nom de l’utilisateur doit être disponible dans la base de donnée.
* Modularité : Le projet a été restructuré ce qui à permis d’avoir une organisation en différents composants précis comme une partie “back-end” et “front-end” indépendantes et des dossiers bien organisés avec des fichiers pour chaque fonctionnalités , exemple : chat ou connection.

## 

## 2.7) Mise en avant des critères ergonomiques

Notre projet devra respecter les 8 **critères ergonomiques de Bastien & Scapin**

Les critères ergonomiques précisés dans la section précédente doivent se refléter et suivre les règles des critères ergonomiques suivants:

* Tout d’abord on peut voir sur la page d’accueil le **contrôle explicite** : celui-ci permet à l’utilisateur de comprendre où il va être redirigé suivant ses actions. Ici les boutons “Se connecter” et “Créer un compte” permettent d’ouvrir une page de connexion et une page de création d’un nouveau compte respectivement. (*Figure 11*)
* Il y a aussi un **guidage** avec des grandes images dans des dispositions usuelles que l’on retrouve sur d’autres sites connus (Facebook, Twitter, Instagram, …). Cet aspect visuel se retrouve alors sur chacune des pages suivantes pour que l’utilisateur ne se perde pas. (*Figure 11*)



*Figure 13 :* Page d’accueil avant connexion

* Il y a aura aussi une **gestion des erreurs** en grisant les zones non accessibles pour certains utilisateurs afin d'éviter une génération de problèmes au niveau du code. (*Figure 12*)



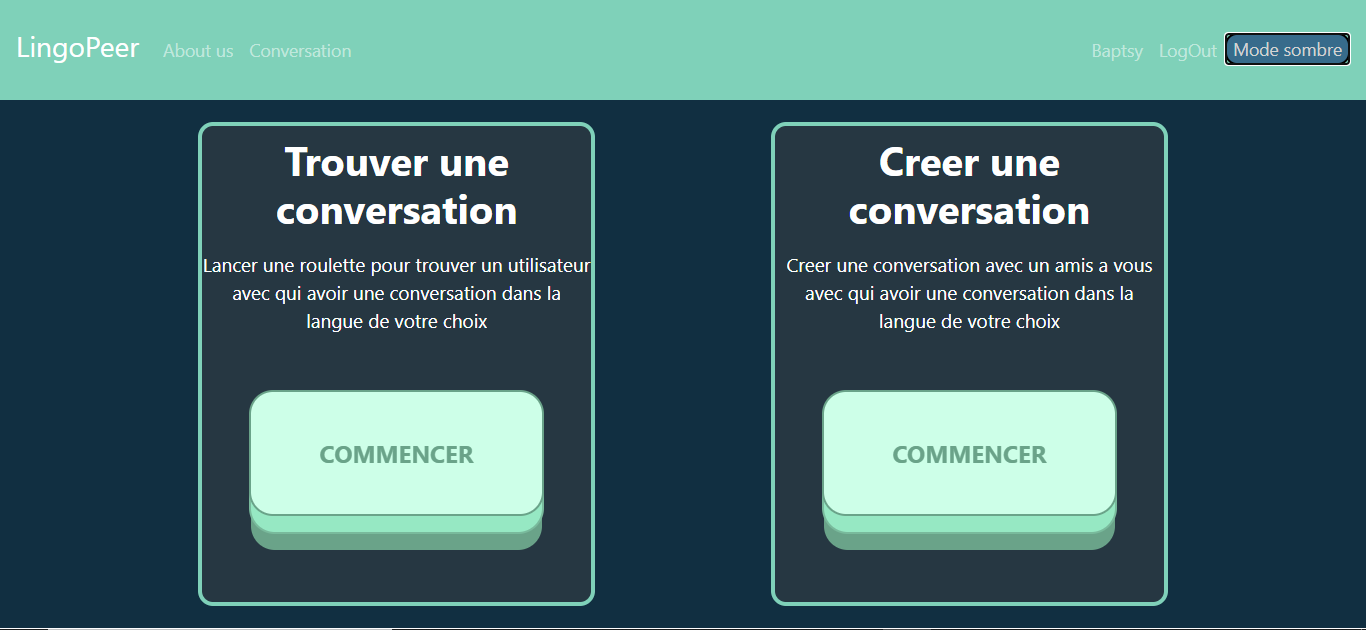
*Figure 14 :* Page d’accueil après connexion avec fonctionnalité indisponible grisé

* Les informations seront présentées de manière à ce que l’utilisateur passe le moins de temps possible à les lire comme le montre la seconde image réalisée à l’aide de “wireframe” : il y a 3 éléments, un minimum de texte dans de larges icônes. (*Figure 13*)
* Le site reste assez **homogène** en suivant une charte graphique. Par exemple, la couleur verte ainsi que la disposition principale restent inchangées. (*Figure 13*)



*Figure 15* : Page d’accueil après connexion

* Le site s'adapte avec l’apport d’un thème sombre, réglable au souhait de l’utilisateur. En effet, ce réglage est de plus en plus apporté au sein même des systèmes d’exploitation. (*Figure 14*)



*Figure 16 :* Page d’accueil après connexion (thème sombre)

Le reste des différentes vues se situe en annexe.

# 3) Mise en oeuvre de la solution

## 3.1) Le choix et les justifications des technologies utilisées

Le détail des outils utilisés pour la création du projet est expliqué dans la suite de cette partie.

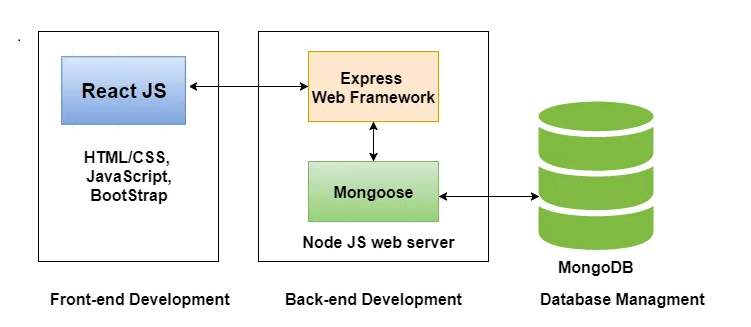
Notre projet est une application web. Ainsi, le choix du langage **JavaScript** était évident puisque :

* il est facile de l’apprendre grâce à divers cours : il est de difficulté relative.
* il est très populaire : c'était le langage le plus utilisé en 2019. Par conséquent, il y a beaucoup de documentation à son sujet.
* il a été conçu pour réaliser des pages web dynamiques.
* il possède de nombreux frameworks¹ qui vont nous permettre de mettre en place différents aspects du projet.

¹framework: un framework désigne un ensemble cohérent de composants logiciels structurels

Avant d’aller plus loin dans nos recherches, nous devions choisir une architecture basée sur javascript. Nous avons retenu la **MERN stack**. Elle permet la création complète de notre application web (back end et front end) en utilisant les quatre technologies suivantes :

* mongodb
* express
* react
* nodejs



*Figure 17* : Schéma M.E.R.N

Ce modèle d'architecture nous permet de gérer à des endroits bien distincts la partie **Back-end**,c’est à dire la partie du code qui se charge d’effectuer les traitements et les operations (exemple: envoyé un utilisateur dans la base de donnée) . L’utilisateur n’as pas besoin de voir cette partie ,la partie visible pour l’utilisateur est le **Front-end** c'est-à-dire la partie visible et la page internet ou l’utilisateur peut interagir avec la page .

Comme annoncé précédemment, nous utiliserons **Node.js**. Il s’agit d’une plateforme logicielle en JavaScript permettant d’utiliser ce langage du côté serveur afin de concevoir des applications en réseau. Cet outil est très similaire au JavaScript ainsi son utilisation sera plus simple. De plus, il est très performant et très utilisé.

Notre application web sera composée d’une seule page qui sera modifiée en fonction des actions de l’utilisateur. De ce fait, nous utiliserons la bibliothèque **React**. Cet outil est conçu pour créer facilement des interfaces utilisateurs. Il met à jour les composants de celle-ci de façon optimale lors d’un changement. Ce dernier peut aussi être utilisé du côté serveur avec **Node**.

Afin de mettre en place un chat audio et peut-être vidéo, nous utiliserons l**’API**¹ **Web Real-Time Communication**, abrégée en WebRTC. Il permettra de mettre en place des communications entre les utilisateurs, et ce en temps réel. WebRTC utilise le système pair à pair pour mettre en place ces communications, donc l’utilisateur n’a pas besoin d’installer de plugins ou de logiciels tiers pour communiquer.

api¹: c’est l’interface avec laquelle le programme interagit et que l’utilisateur ne voit pas.

Pour stocker les données des utilisateurs, nous avons choisis d’utiliser **MongoDB**, il s’agit d’un système de gestion de base de données. Celui-ci permet des liaisons avec les principaux langages de programmation, notamment JavaScript. MongoDB est automatisé, sécurisé et fait régulièrement des sauvegardes : d’où ce choix. Il est aussi composé de beaucoup d’outils qui simplifient le travail des développeurs, par exemple : il permet de visualiser des données grâce à des graphiques mis à jour en temps réel.

## 3.2) Gestion du travail

Afin de mener un projet dans les meilleures conditions, il faut à tout prix une bonne dynamique de travail.

Il faut dans un premier temps, s’assurer que tout le monde ait un rôle prédéfini : c’est donc le chef de projet qui a défini les rôles suivants.

CLERC Louis : Gestion de la connexion vidéo /back end

GENTIL Driss : Gestion du peer to peer /back end

DELHOMME Baptiste : Développeur UX /front end

HIGGINS Alec : Développeur UI /front end

DAIKH Nassim : Interaction client serveur /back end

KADIRI Saad : Gestion de base de donnée/back end

Cette répartition n’est pas anodine puisqu’elle permet de donner une dynamique de binôme. Ce choix a été mis en place en se basant sur le projet réalisé en java, en effet il permet d’avoir une personne avec qui régler un problème sans pour autant impacter toute la progression du projet.

Mais il reste primordial que chacun reste informé sur ce que font les autres membres. C'est pourquoi avant de démarrer chaque session de travail collective, nous organisons une réunion où l’on échange sur ce que l’on a travaillé. Cela permet de garder une bonne motivation de groupe et assure une communication.

Pour connaître rapidement l’avancement de chacun, nous avons décidé d’utiliser le logiciel “trello” qui est un diagramme de Gantt interactif. Celui-ci peut aussi attribuer des tâches spécifiques à un membre de l'équipe. (Voir le calendrier Trello en *annexe*).

Le projet n'a pas seulement besoin d’une bonne gestion des charges de travail, il faut définir un modèle de développement en accord avec celui-ci .

Afin de rendre le projet plus digeste pour une équipe il est nécessaire de procéder à un découpage des de ce dernier afin d’en réduire la complexité .

Nous avons décidé d’adopter un découpage itératif car un effet toute l'équipe travaille à la réalisation de l’application et ce en simultané il est donc important de d’utiliser un processus itératif. En clair ,on souhaite réaliser des prototypes fonctionnels de l’application et ajouter des fonctionnalités au fur et à mesure. Autrement dit, c’est une démarche Bottom-up car on va venir ajouter mais il se peut tout à fait que l’on repasse sur une démarche top down une fois toutes fonctionnalités implémentées.

Cette logique s’inscrit dans les méthodes agiles c'est-à-dire que la méthode est adaptative et prédictive.

3.2)

# 

# 

# 

# 

# 

# 

# Annexe:

*Scénario d’utilisation, sur lequel s’est basé le BPMN:*

**Login:**

L’utilisateur arrive sur la page d'accueil, il aura deux choix :

* Le bouton se connecter (en haut à droite) , cela le renverra sur une page ou il va devoir rentrer son adresse mail et son mot de passe , si son compte existe il y aura un petit pop up lui disant qu'il est bien connecté ou non .

Si l’utilisateur n’a pas de compte il pourra cliquer sur “créer un compte”, il devra remplir un formulaire lui demandant son nom, prénom, adresse mail, mot de passe, langue(s)

* Le bouton “rester déconnecter”, il devra seulement entrer un prénom et des langues pour continuer.

**Changer ses données personnelles:**

L’utilisateur est sur la page d’accueil, il clique sur le bouton “Mon compte” (en haut à droite à côté de sa photo de profil). Il se retrouve sur la page de son profil. Il peut alors:

* Changer son niveau de langue : Il clique sur un menu déroulant qui lui propose les choix suivants: “débutant”, “intermédiaire”, “expérimenté”, “expert”. Après avoir choisi celui qui lui convient, le menu se ferme et son niveau de langue est changé.
* Ajouter une langue: Dans la partie “Mes Langues”, il clique sur le bouton “Ajouter une langue”. Une page pop-up sur son écran, sur cette page il y a une liste de langues (“anglais”, “français”, “espagnol”, “italien”, “Russe”), cette liste de langues ne comprend pas les langues déjà liées à son compte. L’utilisateur choisit alors la langue qu’il veut ajouter, ensuite la page pop-up disparaît et la langue choisie est ajoutée.
* Enlever une langue: Dans la partie “Mes Langues”, il clique sur le bouton “Enlever une langue”. Une page pop-up sur son écran, sur cette page il y a une liste de langues (“anglais”, “français”, “espagnol”, “italien”, “Russe”), cette liste de langues ne comprend pas les langues déjà liées à son compte. L’utilisateur choisit alors la langue qu’il veut enlever, ensuite la page pop-up disparaît et la langue choisie est enlevée.
* Ajouter/Changer sa photo de profil: Dans la partie “Ma photo”, il clique sur le bouton “Changer ma photo de profil”. Il choisit ensuite une photo sur sa machine (formats pdf, jpg, jpeg uniquement). Si il n’y a pas d'erreurs, le message “Changement réussi” apparaît sous le bouton “Changer ma photo de profil”. Si il y a eu une erreur, le message correspondant apparaît en rouge sous le bouton “Changer ma photo de profil”.

**Trouver une conversation:**

L’utilisateur est sur la page d’accueil, il y a trois gros boutons rectangulaire. Il clique sur le bouton : “trouver une conversation”. Il arrive sur la page de paramétrage, il peut alors:

* Choisir la langue sur une liste comprenant “anglais”, “français”, “espagnol”, “italien”,”allemand”.
* Choisir le thème parmi une liste (...)
* Faire tourner la roue en cliquant sur une icône d’une roue, cela lancera une petite animation, le prénom d’un utilisateur en ressortira .
* Lancer la conversation.

**Créer une conversation avec un ami:**

L’utilisateur est sur la page d’accueil, il y a trois gros boutons rectangulaires. Il clique sur le bouton : “Créer une conversation avec un ami”. Il arrive sur la page de paramétrage, il peut alors:

* Choisir un ami de sa liste avec qui avoir une conversation
* Choisir la langue
* Choisir le thème

Son ami recevra une notification lui indiquant le nom de la personne voulant avoir une conversation avec lui, si oui ou non il veut rejoindre celle-ci, la langue sera indiqué, et le thème si il a été choisi.

* Lancer la conversation, seulement si il y a bien deux personnes.

Aller sur le fil de partage

(dépendra du temps qu’il nous reste : pas sur de le faire)

**Gérer ses amis:**

Envoyer une demande avec nom d’utilisateur:

L’utilisateur est sur la page d’accueil, il clique sur le bouton “Mes Amis” (en haut à droite à côté de “Mon compte”. Il arrive sur la page d’amis, il y a la liste de ses amis au milieu de la page. À côté de la liste il y a un input de texte “Email”, un input texte “Message” et un bouton “envoyer la demande”. Il devra remplir les champs précédents (le message est optionnel) puis appuyer sur le bouton “envoyer la demande”. Si il n’y a pas d’erreurs, le message “demande envoyée” apparaît sous le bouton, en cas d’erreur le message associé à celle-ci apparaît sous le bouton.

Demander en amis durant une conversation:

L’utilisateur est sur une conversation, si il le souhaite il peut envoyer une demande d’amis directement de cette page. Pour cela il clique sur le bouton “ajouter en ami”, cela envoie automatiquement une demande à son interlocuteur.

Envoyer un message à un ami:

L’utilisateur est sur la page “Mes Amis”, il a alors accès à sa liste d’amis. Il peut cliquer sur le bouton “Message” devant le nom de l'ami à qui il veut envoyer un message. Il arrive alors sur la page de message, il y a le nom de l’ami, sa photo de profil, un input de texte “Message” et un bouton “Envoyer”. Il remplit alors le champ “Message” et clique sur le bouton “Envoyer”. Si il n’y a pas d’erreurs, le message “Message envoyé" apparaît sous le bouton, en cas d’erreur le message associé à celle-ci apparaît sous le bouton.

**Fonctionnalitées de la page de conversation:**

Page de chat écrit uniquement:

Lorsque l’utilisateur est sur cette page, il y a un chat au milieu de la page. Il peut envoyer et lire des messages en direct avec son interlocuteur. Il peut aussi envoyer une demande d’ami.

Page de chat audio/vidéo:

Lorsque l’utilisateur est sur cette page, il y a le retour vidéo au milieu gauche de son écran, et le chat écrit est au milieu gauche. Il a le choix de converser en chat audio ou en chat écrit. En bas de la page, il peut Activer/désactiver sa webcam, activer/désactiver son micro, mais aussi ajouter en ami son interlocuteur.

Page de connection :

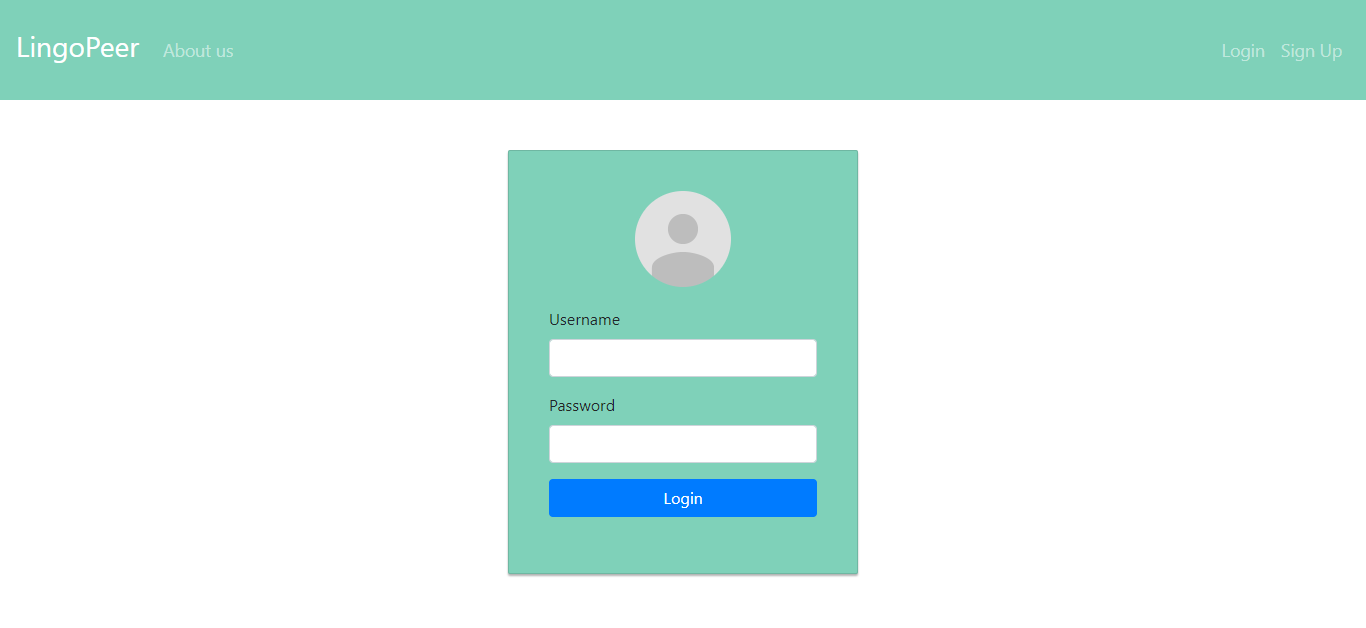


Figure 18

Page de création de compte :

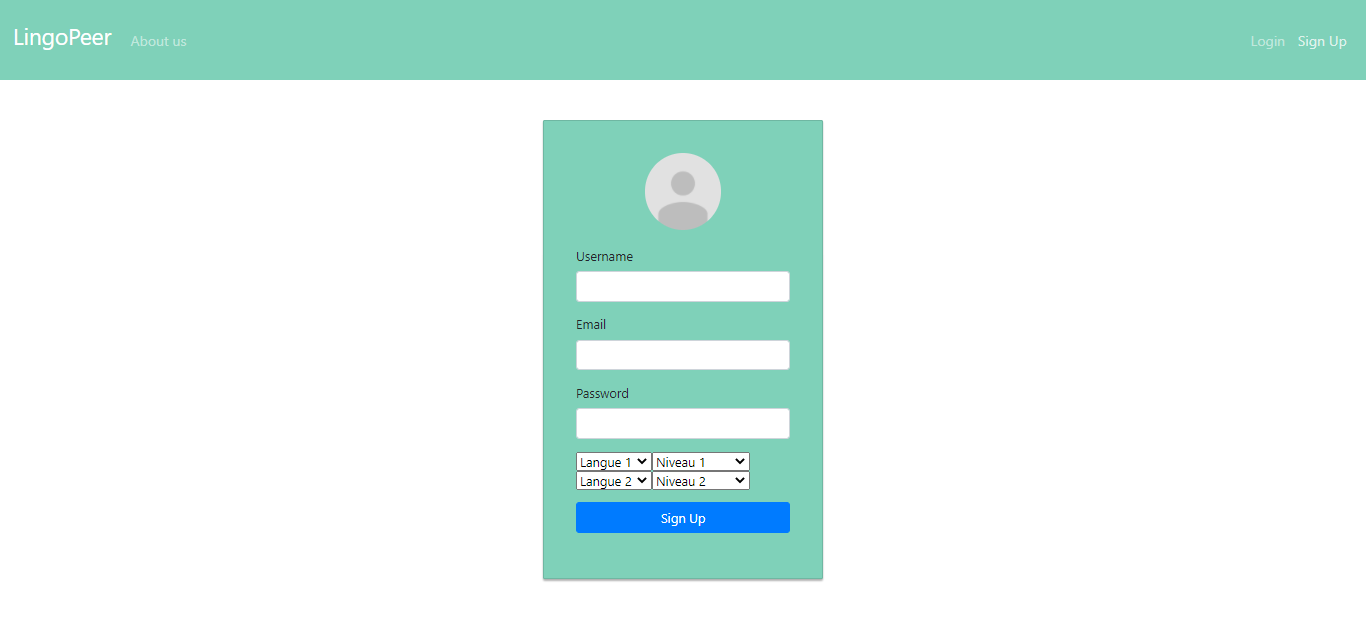


Figure 19

Page de profil :



Figure 20

Trouver une conversation :

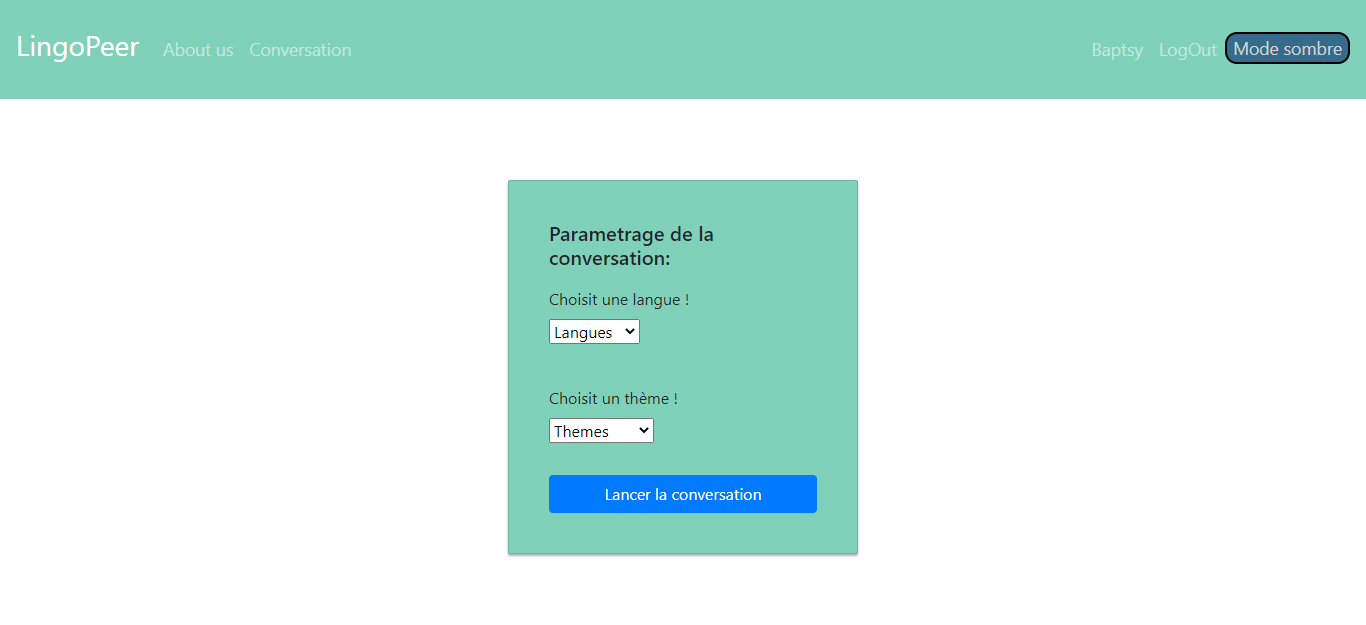


Figure 21

Créer une conversation avec un amis :

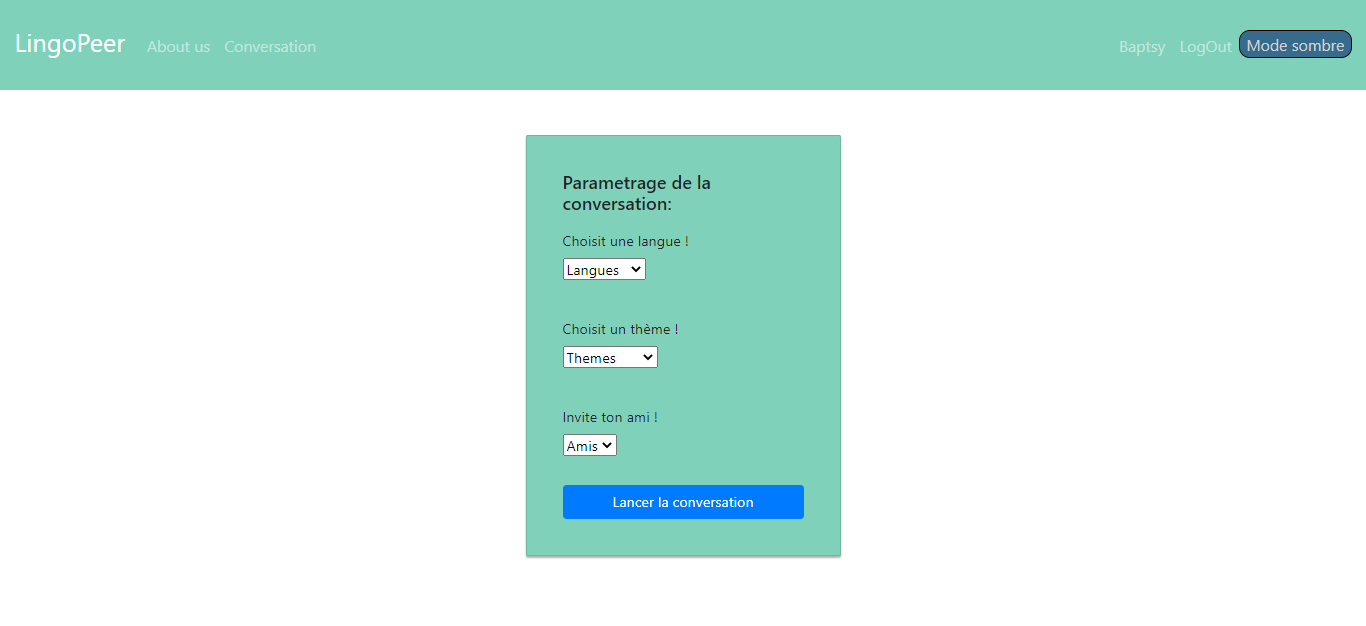


Figure 22

Chat écrit :

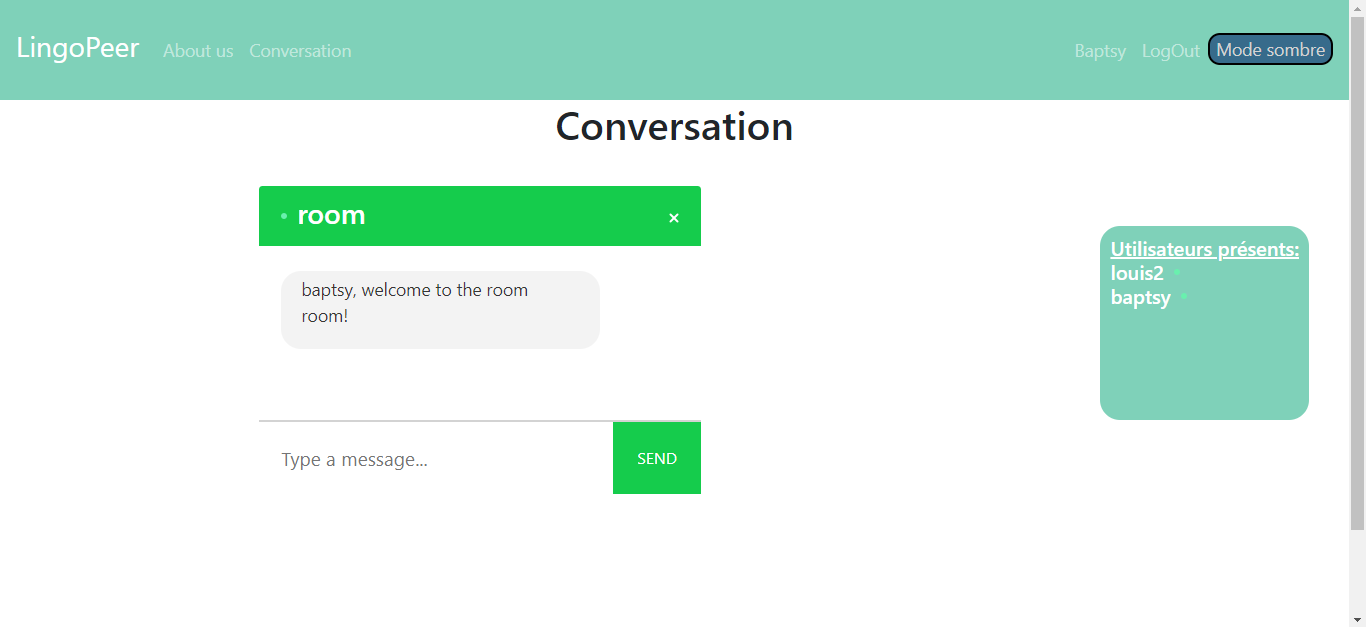


Figure 23

Chat video, oral et écrit :

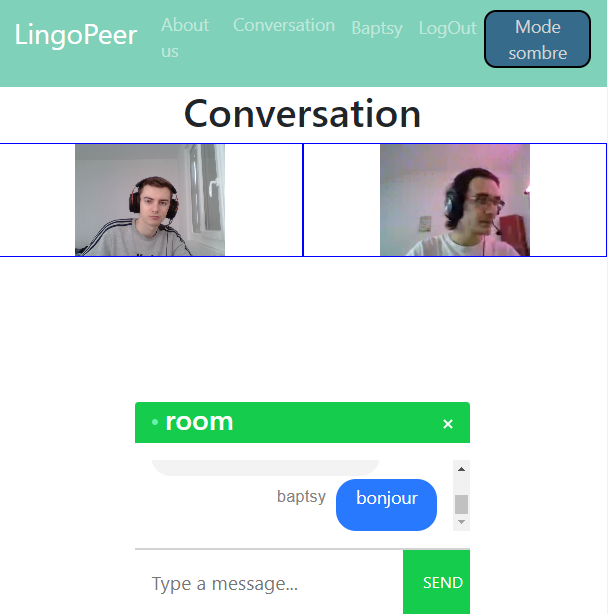


Figure 24

Screen Trello:

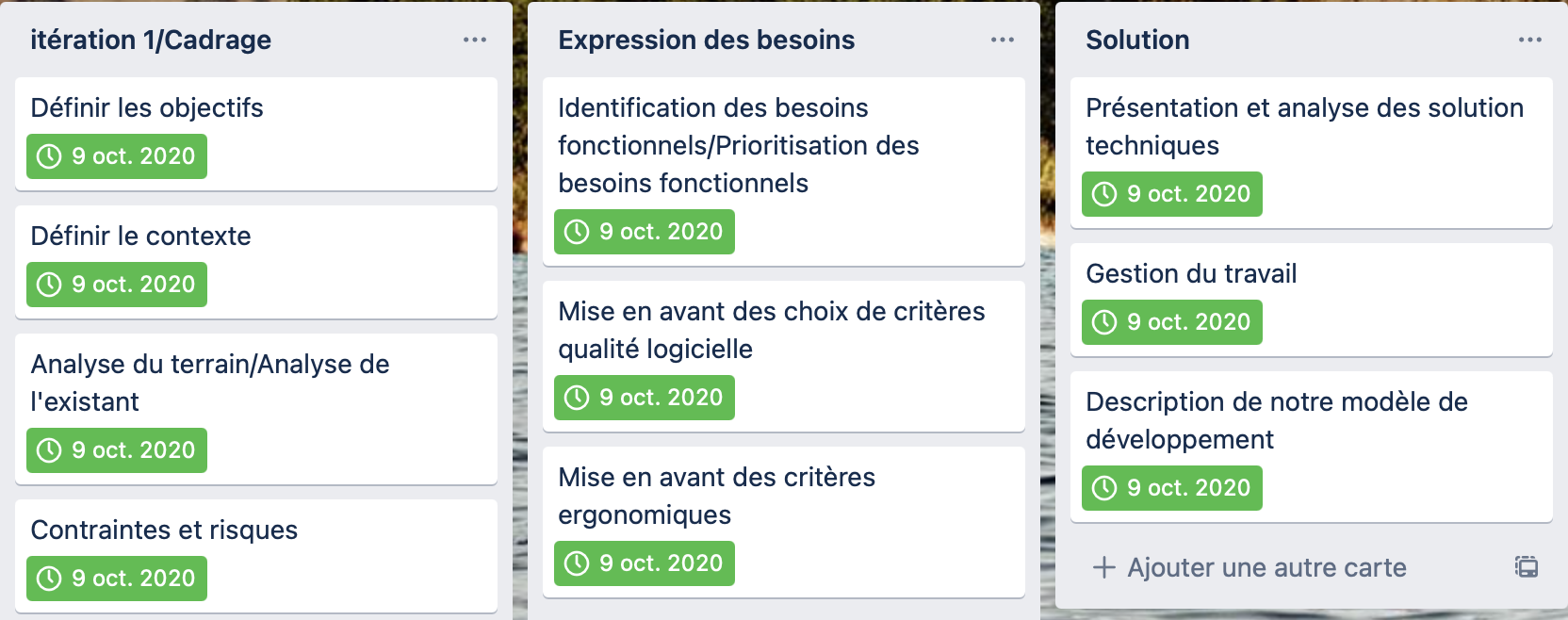


Figure 25

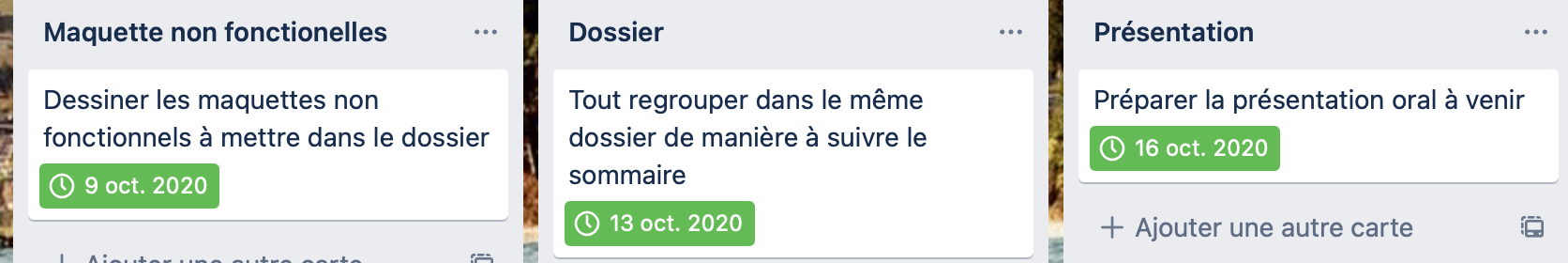


Figure 26



Figure 27

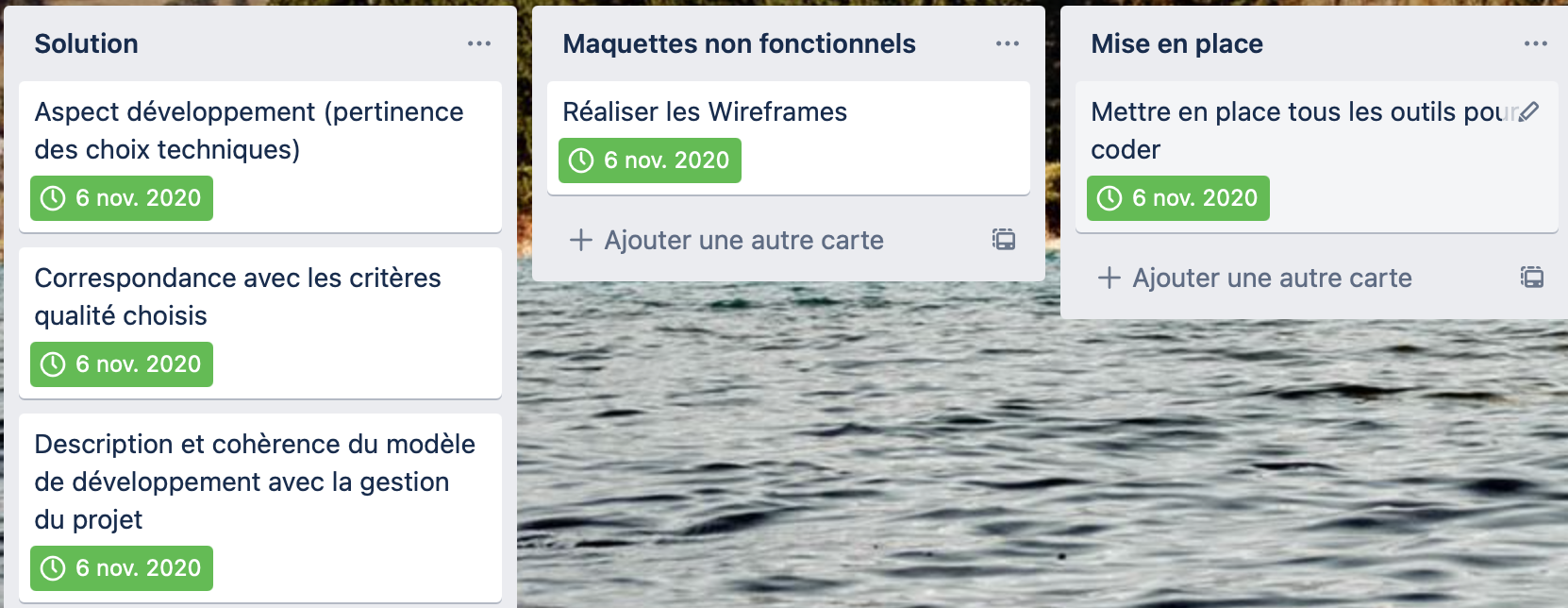


Figure 28

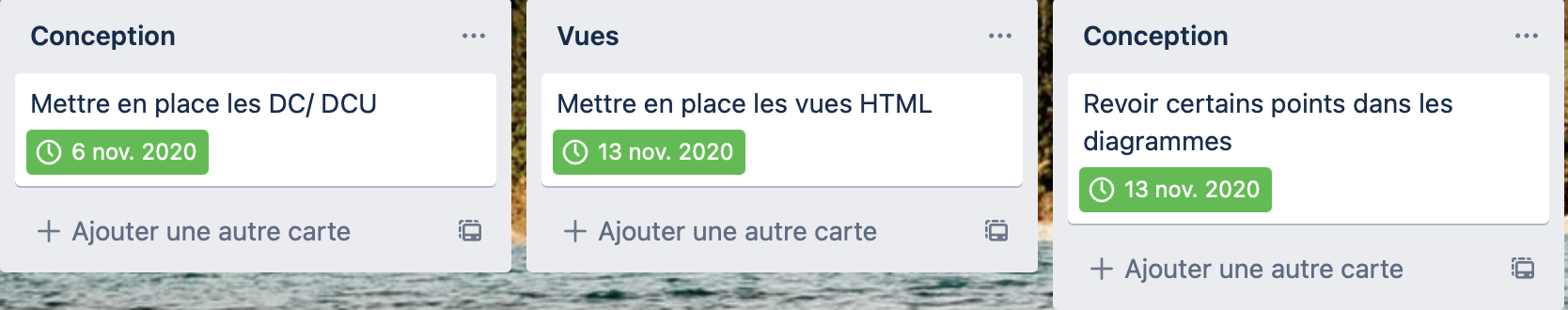


Figure 29

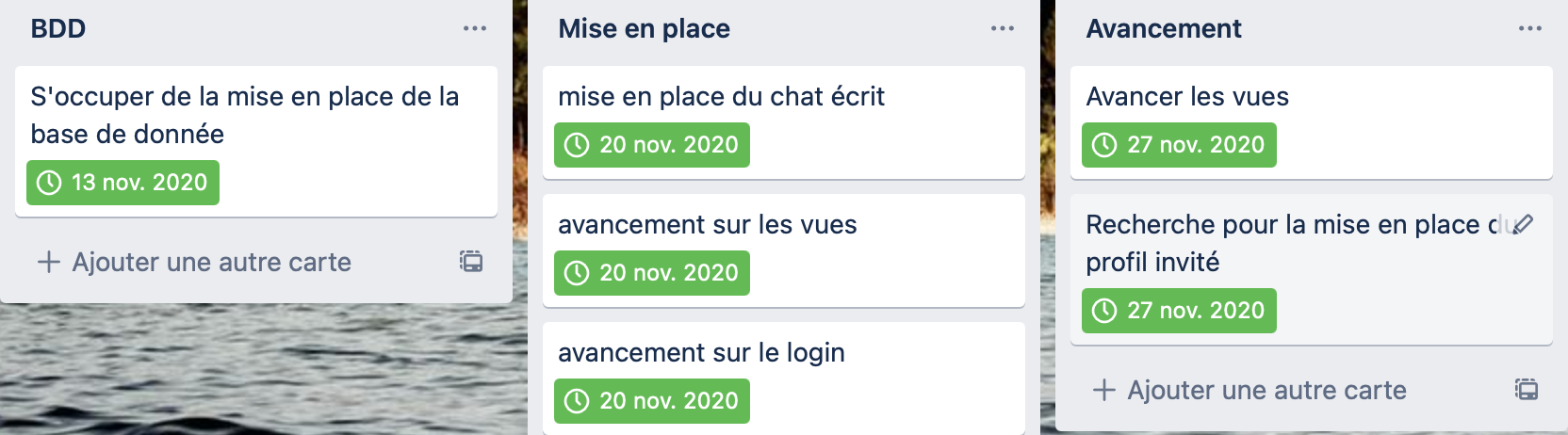


Figure 30

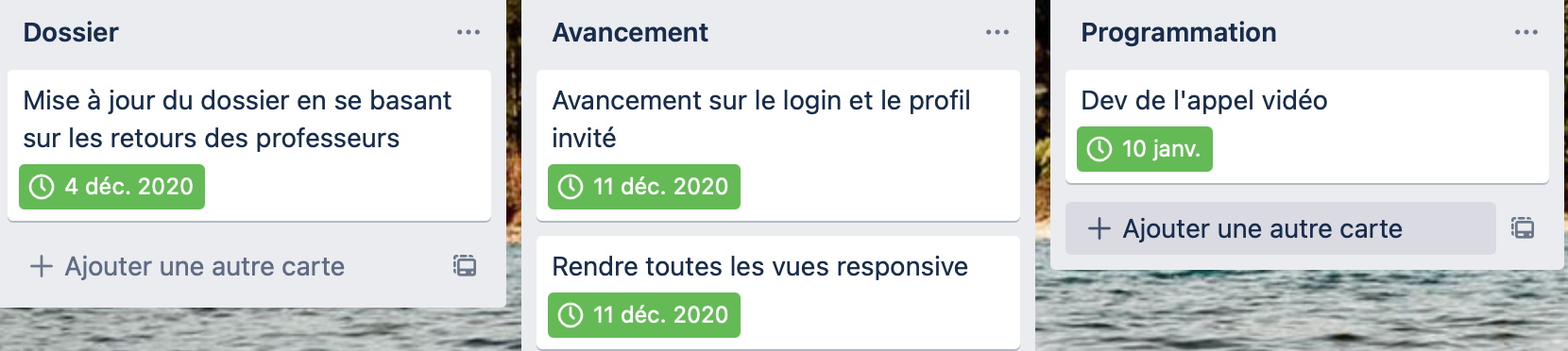


Figure 31

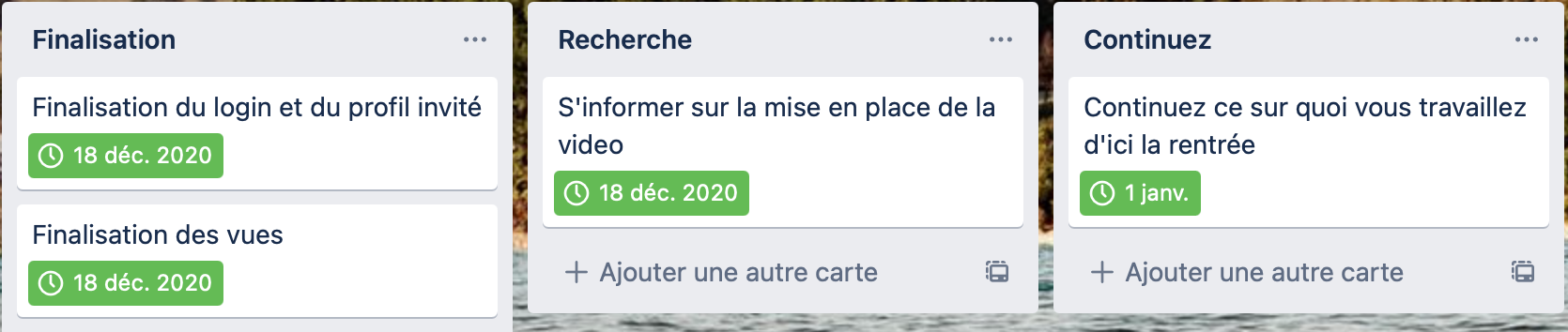


Figure 32



Figure 33