18/02/2021

Ali-bis

Projet Web

Luke Cornaz & Gabriel Pereira

SI-C2a

Table des matières

[Analyse préliminaire 2](#_Toc66092222)

[Introduction 2](#_Toc66092223)

[Objectifs 2](#_Toc66092224)

[Planification initiale 3](#_Toc66092225)

[Analyse 4](#_Toc66092226)

[Cahier des charges 4](#_Toc66092227)

[Définition de l’audience 5](#_Toc66092228)

[Analyse concurrentielle 5](#_Toc66092229)

[Cas d’utilisation 5](#_Toc66092230)

[Home : 5](#_Toc66092231)

[Register : 7](#_Toc66092232)

[Login : 8](#_Toc66092233)

[Publier une annonce : 9](#_Toc66092234)

[Achter une annonce : 11](#_Toc66092235)

[Contacter un annonceur : 12](#_Toc66092236)

[MCD 13](#_Toc66092237)

[Nom du site et du domaine 14](#_Toc66092238)

[Stratégie de test 14](#_Toc66092239)

[Planification détaillée 15](#_Toc66092240)

[Conception 18](#_Toc66092241)

# Analyse préliminaire

## Introduction

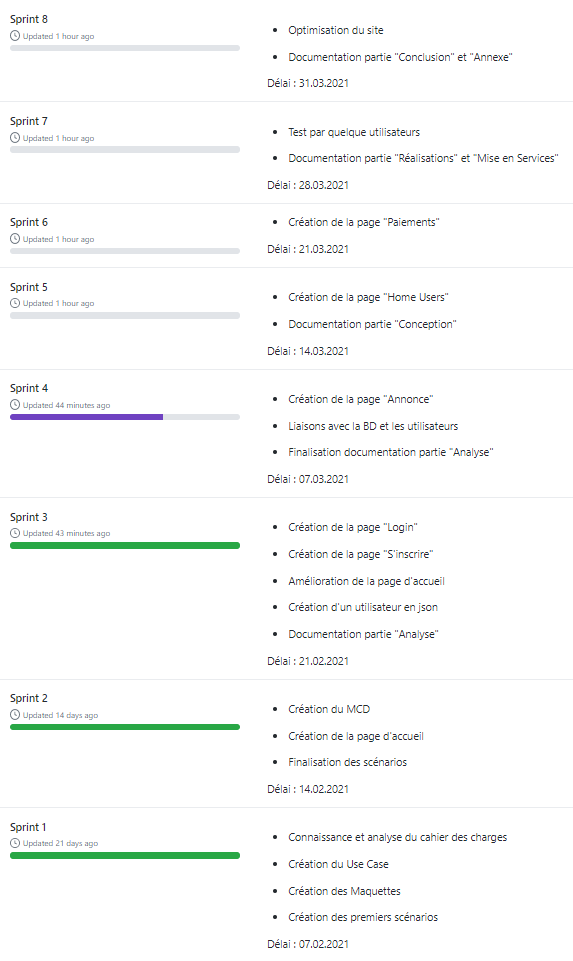
Pour ce projet web, nous avons dû créer un site web fonctionnel avec une petite base de données en JSON en 8 semaine soit un trimestre. Nous l’avons fait, pour nous familiariser avec l’utilisation du PHP sur un site internet, et aussi le fait de coder sur plusieurs fichiers que nous devons lier pour que chaque page du site puisse fonctionner correctement.

Chaque semaine, nous devons faire un rendu du travail effectué via un email contenant le journal de travail et le lien GitHub de notre projet, pour que le professeur puisse accéder facilement à notre code.

## Objectifs

Le but de ce projet est de pouvoir naviguer sur un site d’annonce et de contacter les différents annonceurs. L’utilisateur doit aussi pouvoir s’authentifier et s’inscrire. Au moment de l’inscription, les informations remplies par l’utilisateur son écrites sur la base de donnée JSON. Il aura aussi la possibilité de visiter, créer et acheter une annonce, mais il devra s’inscrire et s’authentifier pour pouvoir le faire. Chaque utilisateur dispose des mêmes droits.

## Planification initiale



# Analyse

## Cahier des charges

* Ce site devra permettre à des membres authentifiés de publier des annonces pour vendre des articles, mettre en location des articles, ou proposer des services (jardinage, nettoyage, cours…).
* Les utilisateurs pourront s’inscrire pour devenir membres.
* Les membres authentifiés pourront ajouter, modifier et supprimer leurs annonces.
* Les annonces vente et location auront un titre, une catégorie (liste déroulante), une description, un prix, et une image (uploadée).
* Les annonces de service auront des images génériques pour chaque activité.
* Les utilisateurs pourront parcourir les annonces par catégorie, ou avec une recherche.
* Sur la liste des annonces, les images des articles publiés seront des miniatures.
* Le détail de l’annonce affichera sa description et une image agrandie.
* Les utilisateurs du site pourront consulter les articles et contacter les membres via un formulaire. Un e-mail automatique avec les coordonnées de l’utilisateur et le titre du produit sera envoyé au membre concerné.
* Les tâches techniques devront être réparties de façon équitable, et apparaitre de façon détaillée sur le planning.
* Le site ne doit pas utiliser de base de données SQL, mais des fichiers JSON.
* Le site doit être responsive.
* Le site se base sur une structure MVC.
* Le site final devra contenir des données concrètes. (~ 5 membres et 10 annonces).

## Définition de l’audience

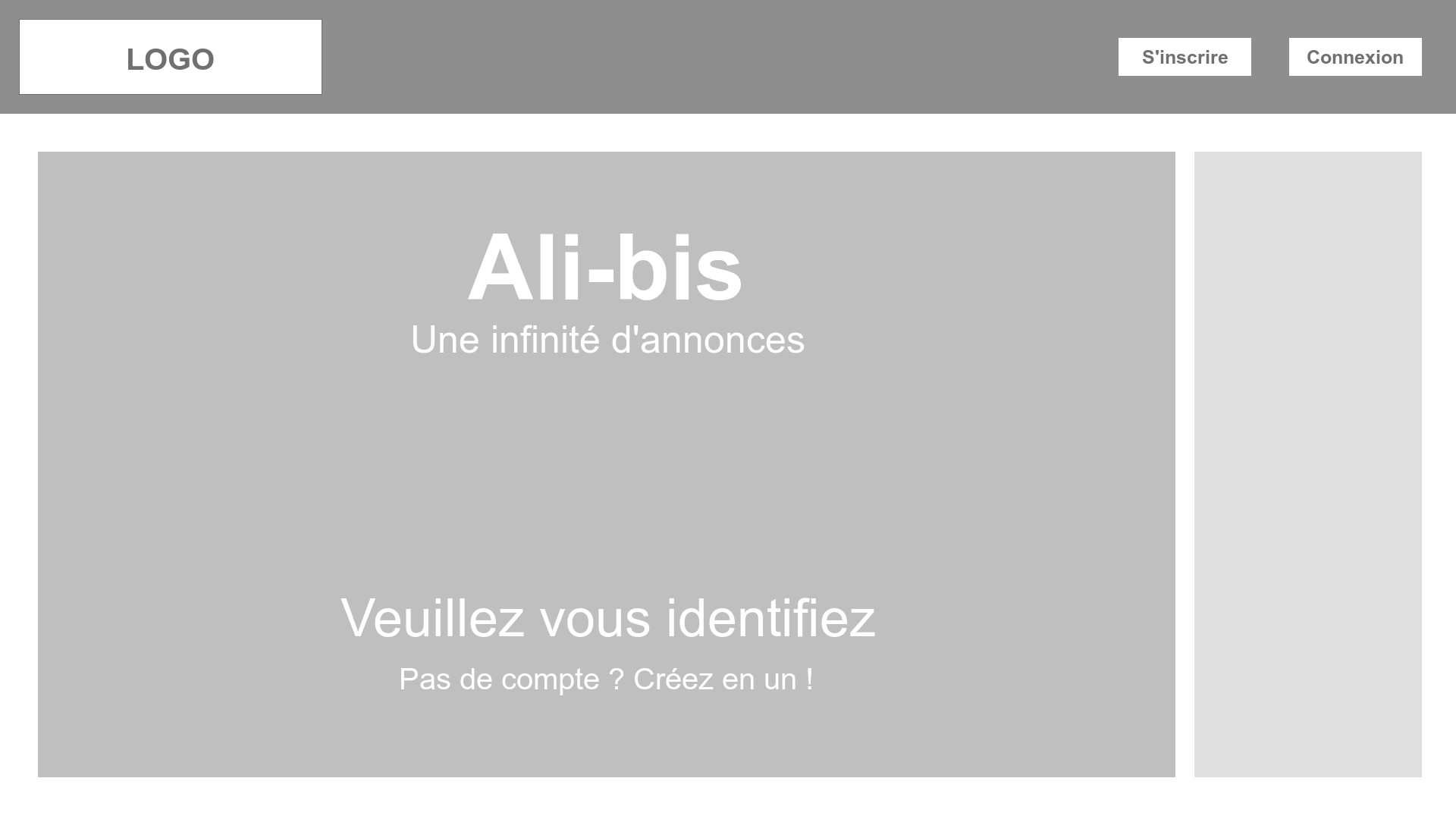
Nous cherchons à cibler un public qui veut utiliser notre site pour vendre ou louer ses biens en sécurité. Les personnes devront assurer les ventes eux-mêmes, nous ne garantissons pas la protection face au arnaqueur. Mais, s’ils se font signalés nous nous occuperons de les bannir pour le bien de nos utilisateurs. De ce fait, nous allons créer un site simple d’utilisations et assez sobre pour que tout le monde soit à l’aise avec son utilisation.

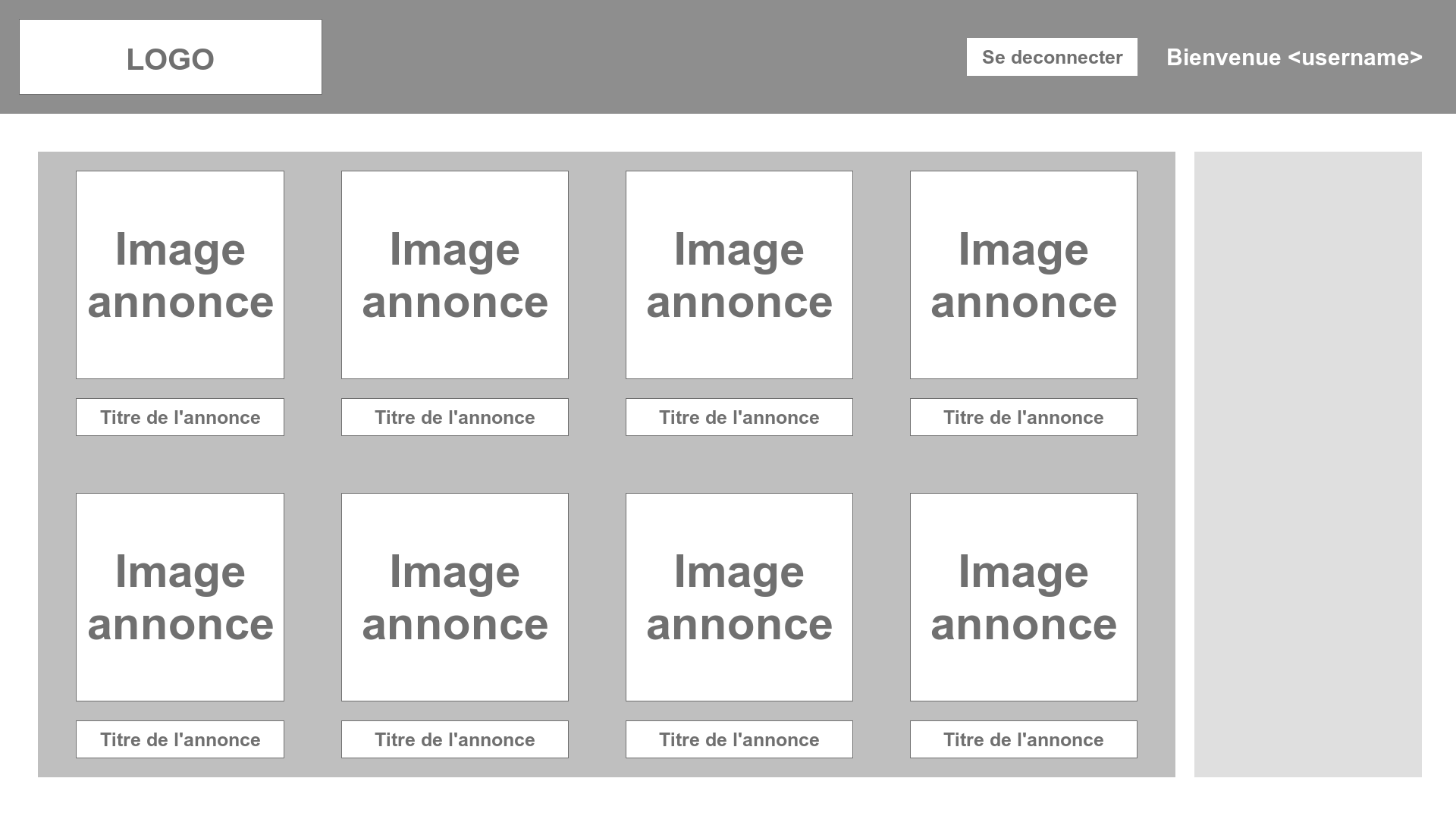
## Analyse concurrentielle

Nos principaux concurrents sont eBay de façon international et Anibis de façon local. Sur eBay, la façon pour contacter un vendeur sera assez similaire à la nôtre mais elle reste, néanmoins, moins facile d’accès que sur notre site. Comme Anibis nous aurons nous aussi un accueil avec les différentes annonces dessus. Mais le fait est que sur Anibis, il est moins ergonomique que le nôtre.

## Cas d’utilisation

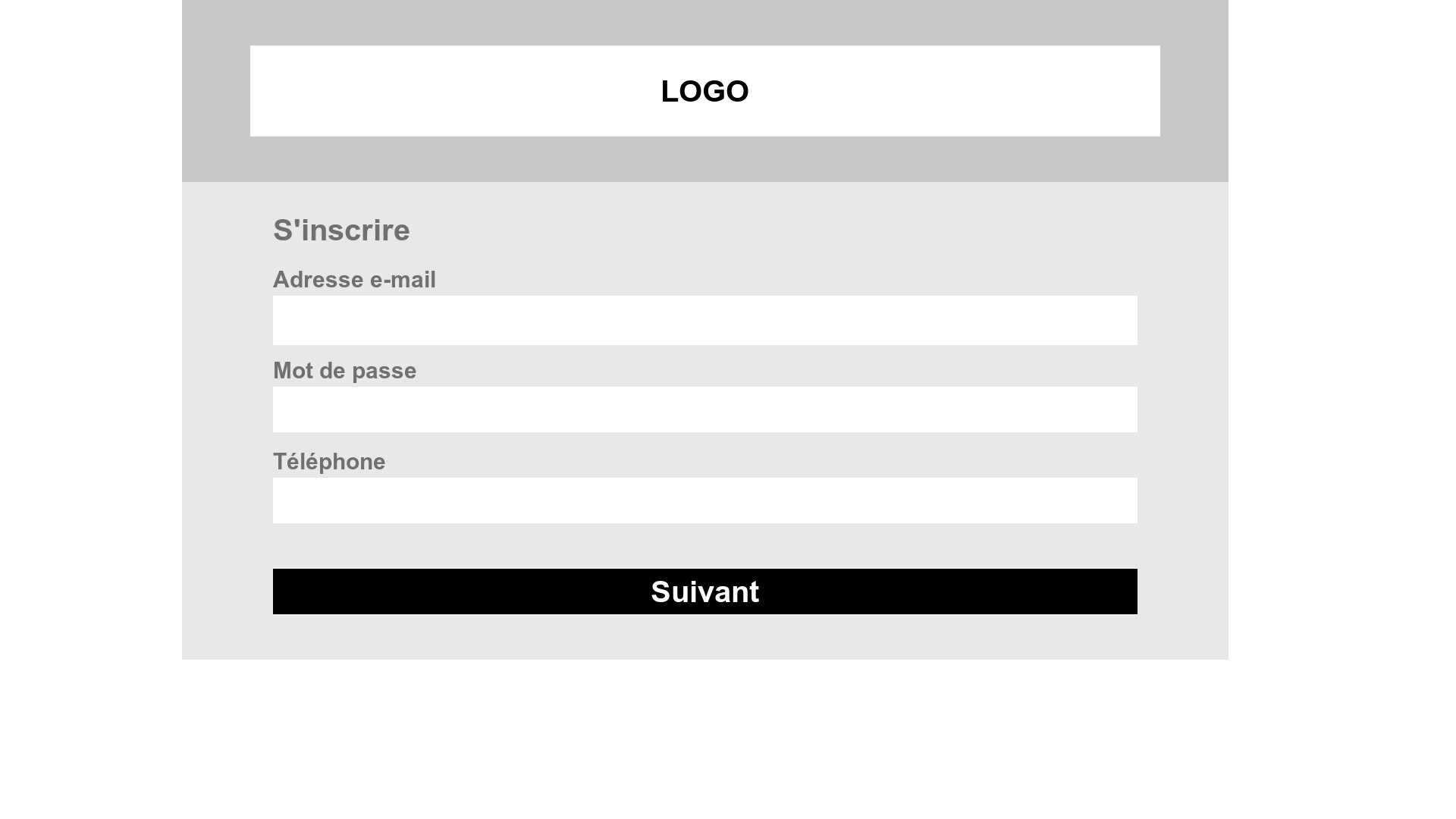
### Home :





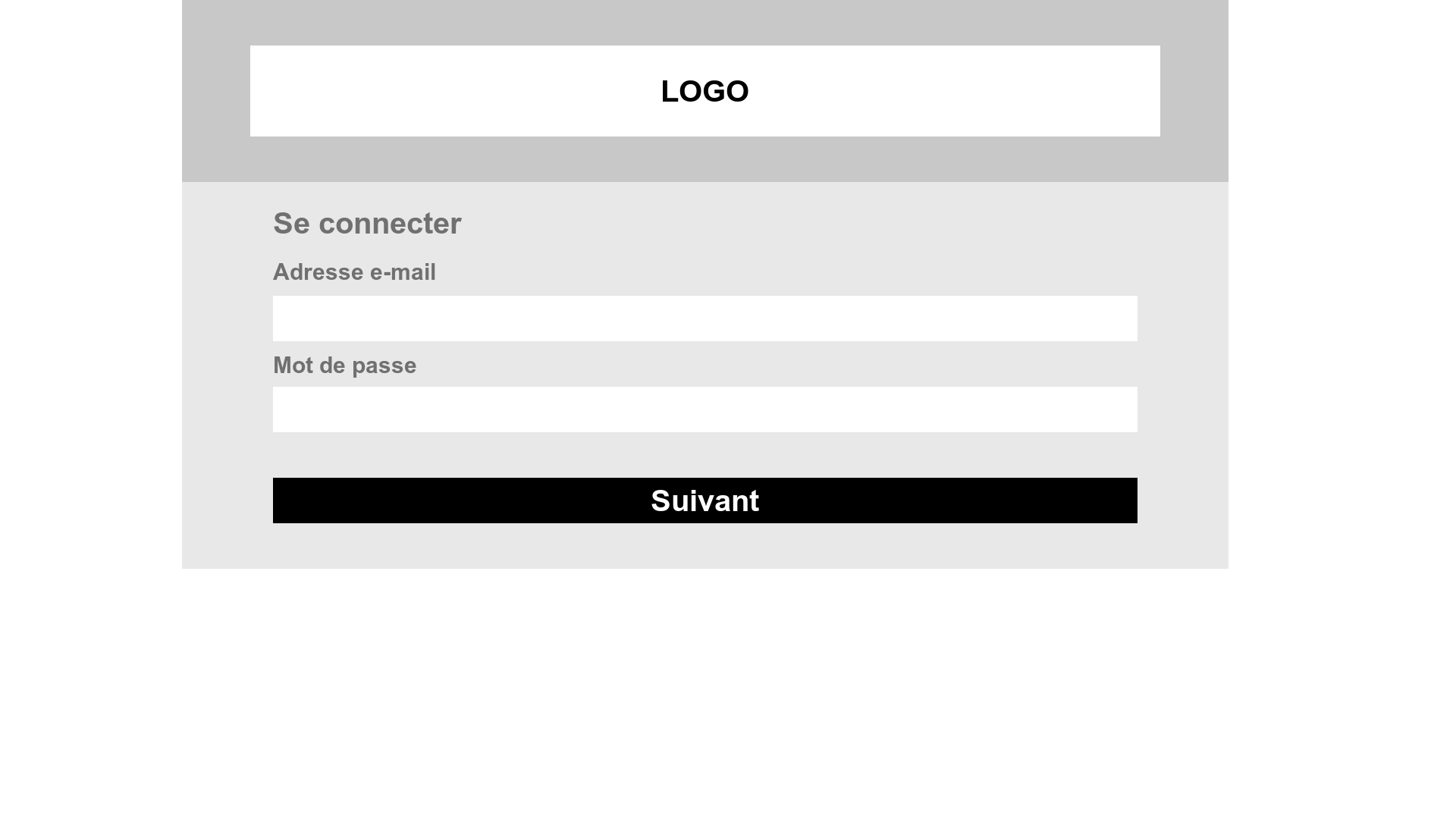


### Register :



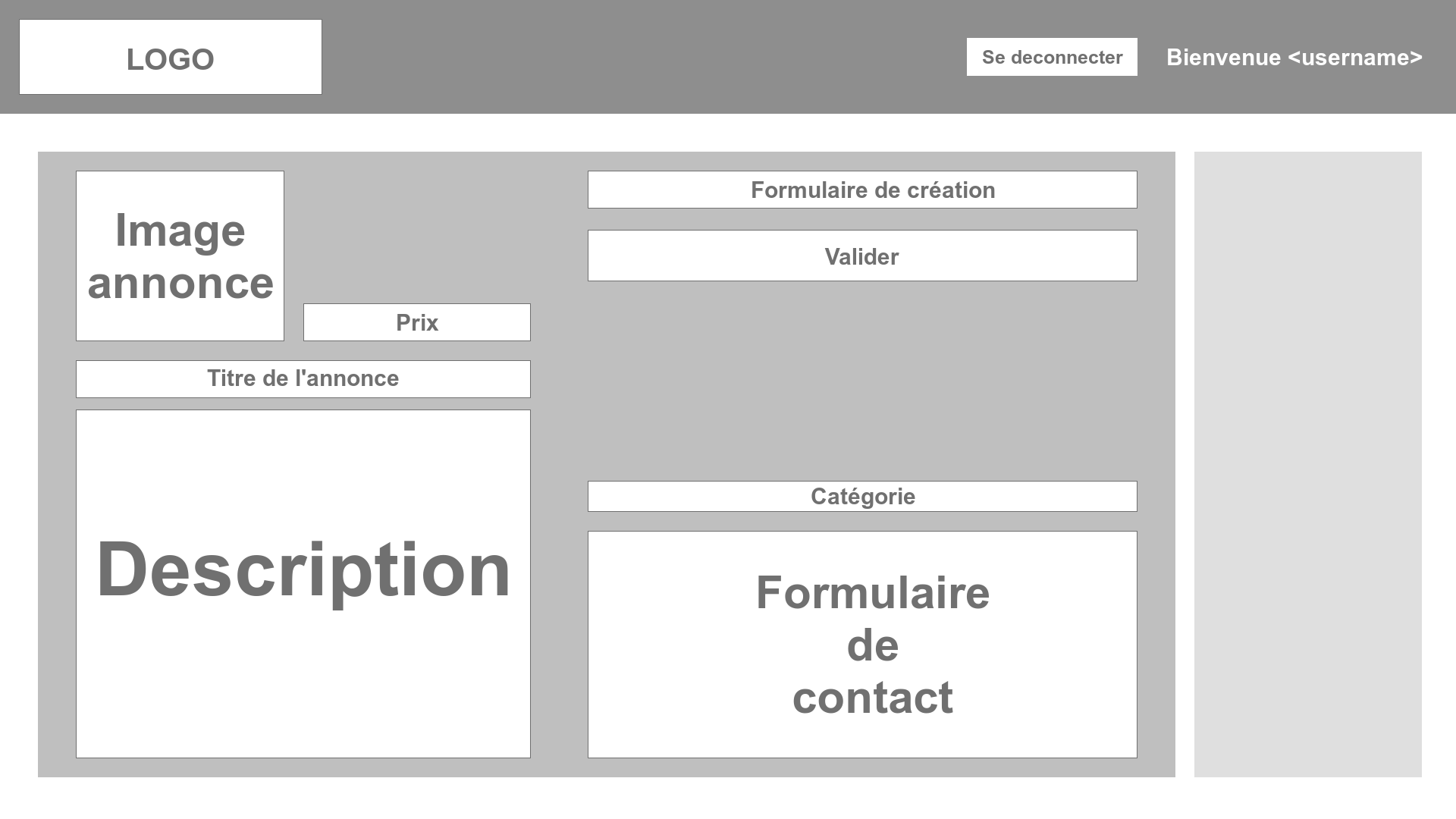


### Login :



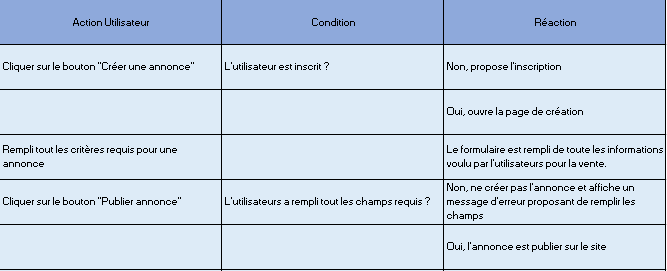


### Publier une annonce :

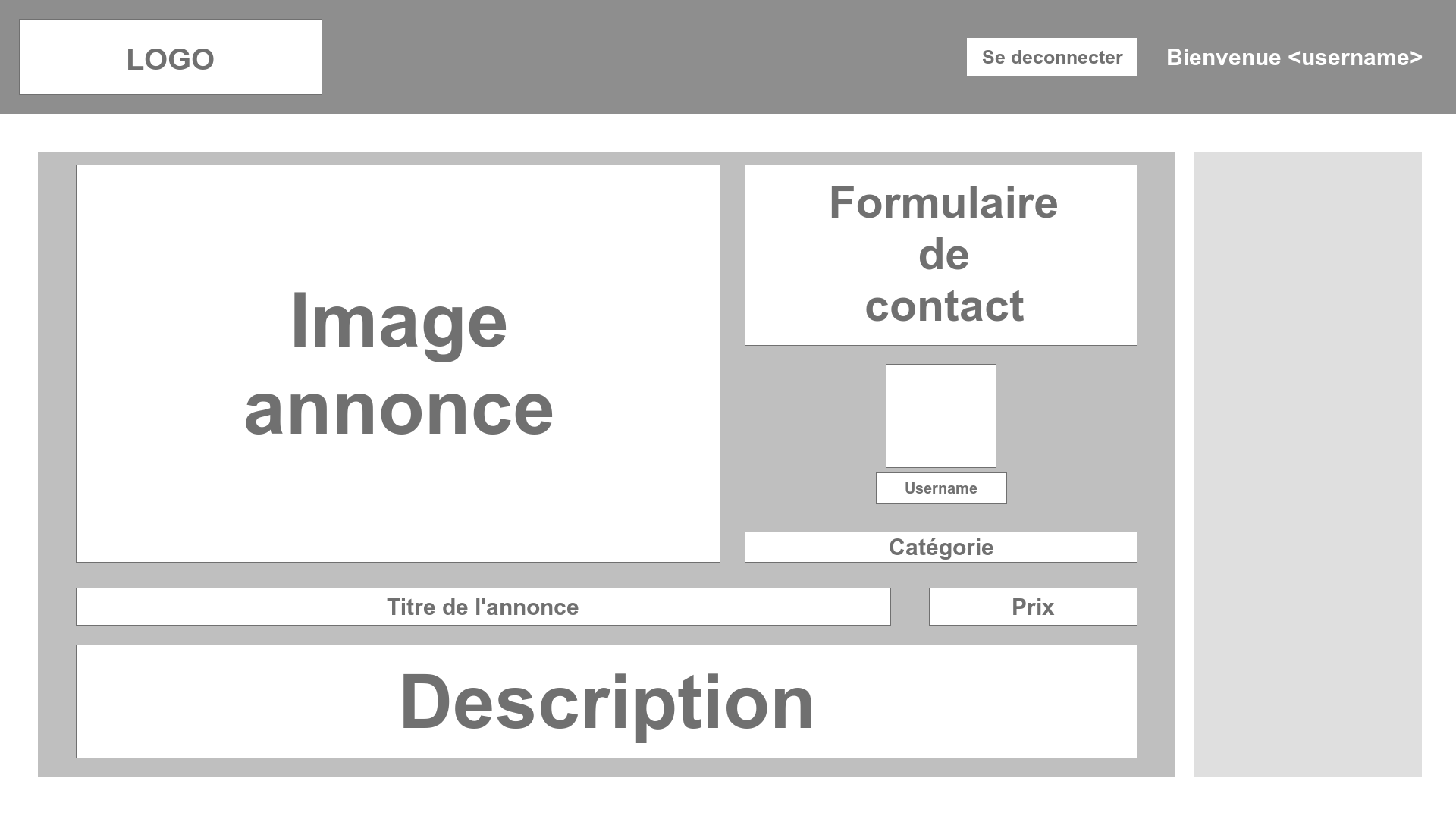






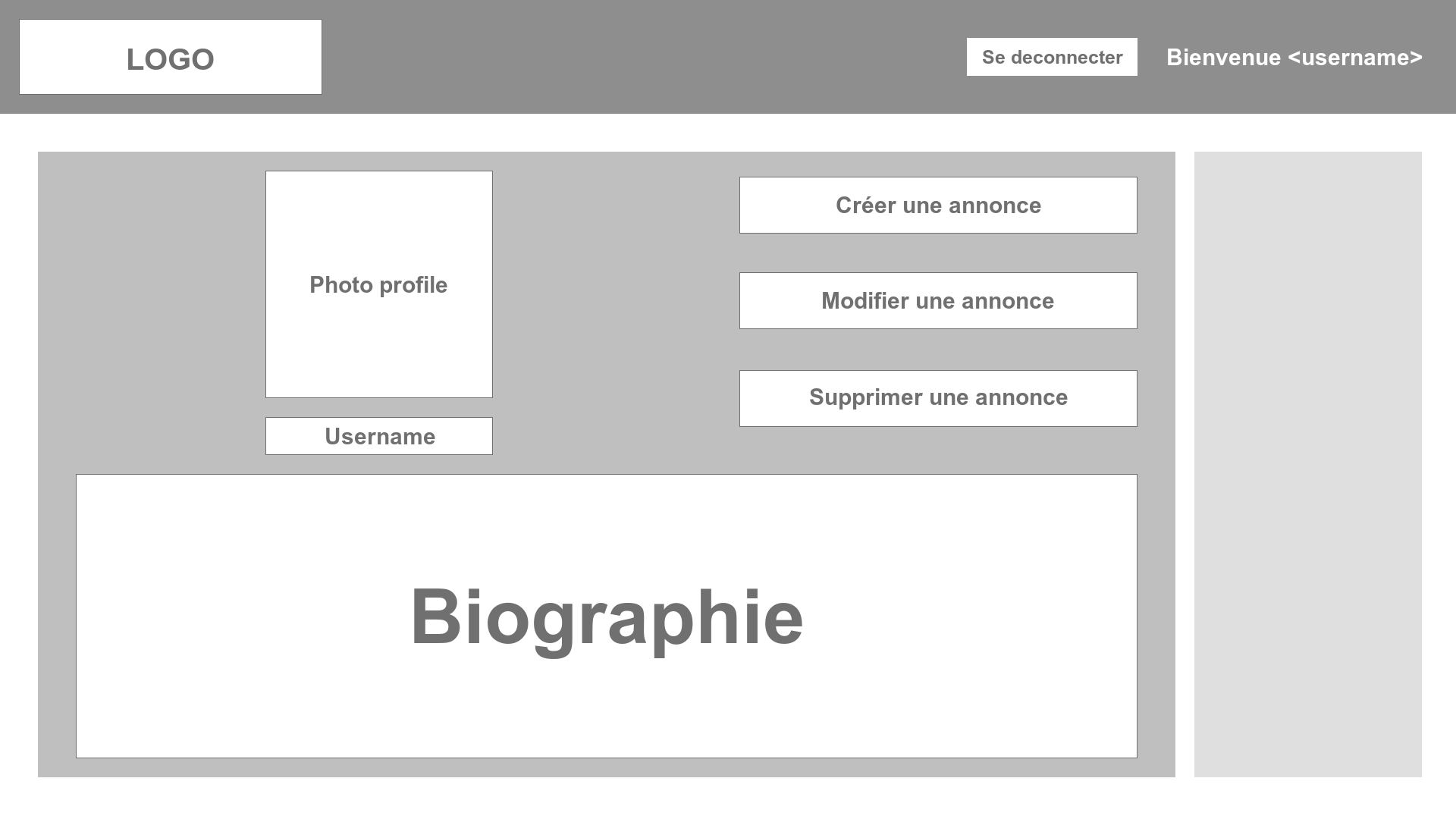


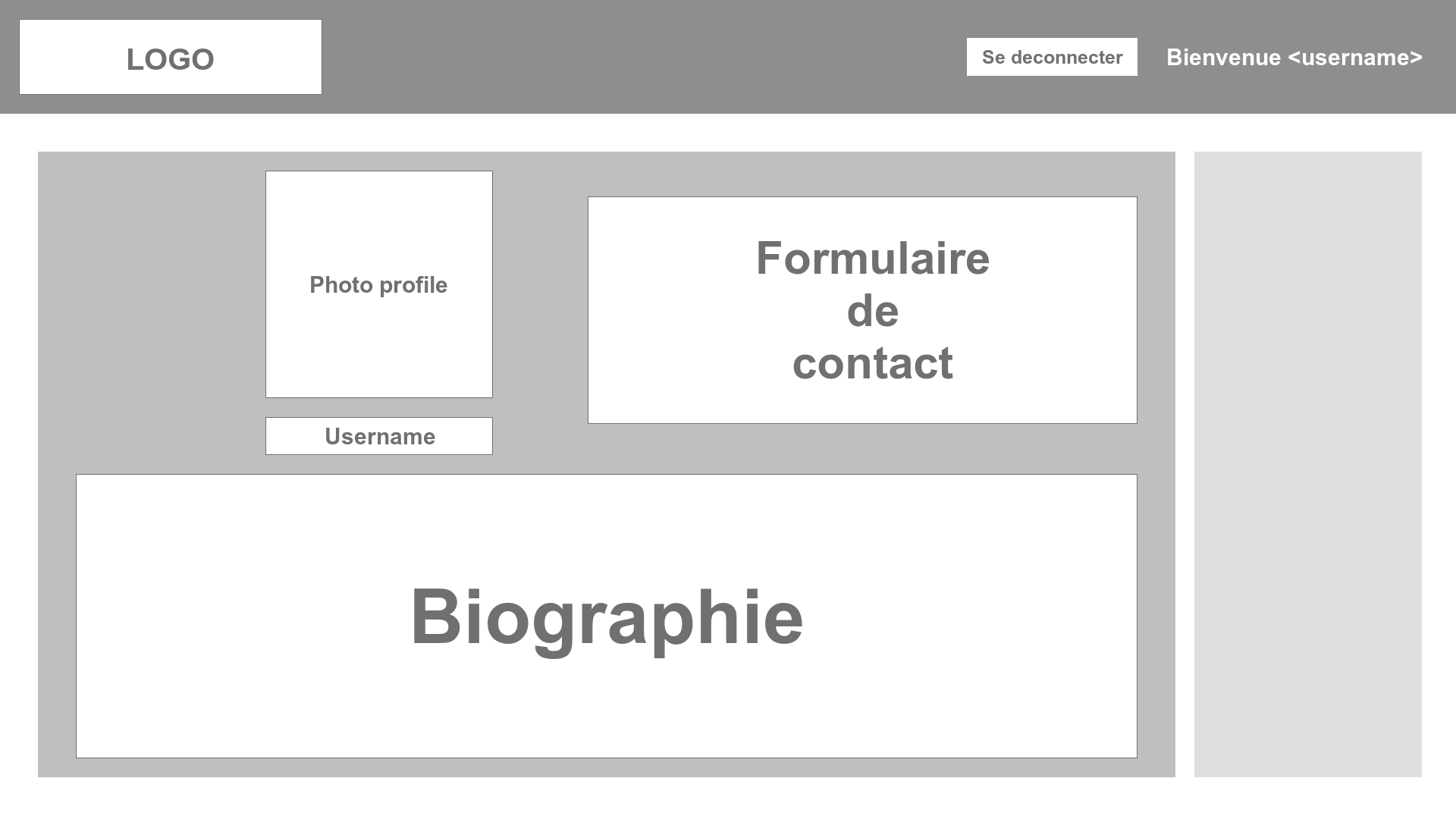
### Achter une annonce :

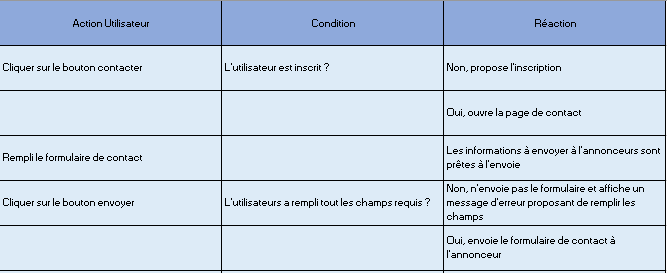




### Contacter un annonceur :

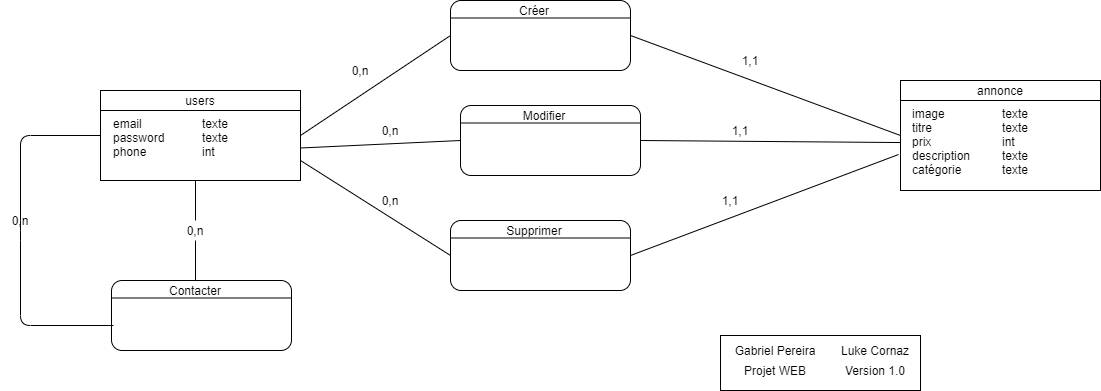






## MCD

Voici à quoi ressemble notre MCD représentant les possibilités que l’utilisateur aura avec notre site et les différente action faisable.



## Nom du site et du domaine

Notre site aura pour nom « Ali-Bis ». Ce nom a été décidé en équipe pour une entente entre chaque créateur du site. Le nom de domaine utilisé sera « ali-bis.ch » que nous aurons grâce à la licence fournit par le CPNV.

## Stratégie de test

Pour le développement de notre site internet nous utiliserons PHP Storm pour coder en même temps grâce à la fonctionnalité « CodeWithMe » que cette application propose. Nous utiliserons l’adresse 10.229.34.44 : 8888 pour nous connecter au site en local, temps qu’il n’est pas en ligne.

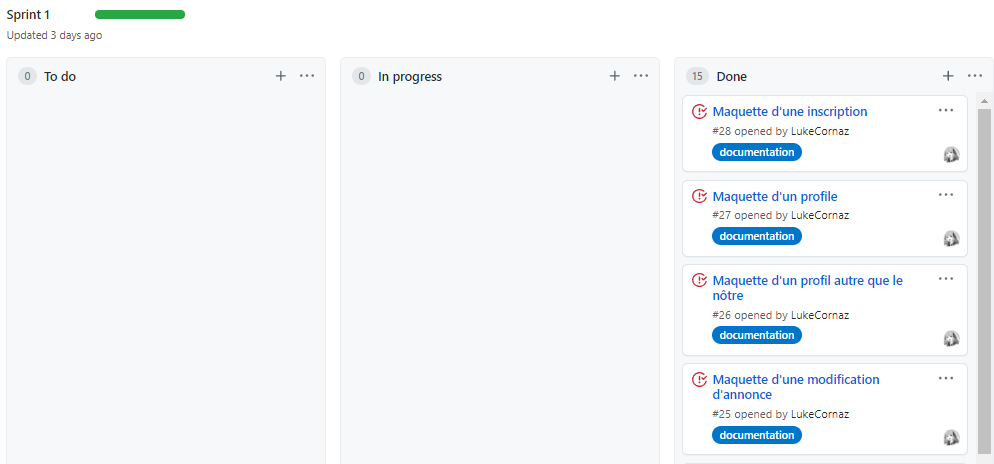
Nous allons créer un site relier à une base de donnée faite en JSON qui rassemble un ensemble de page navigable.

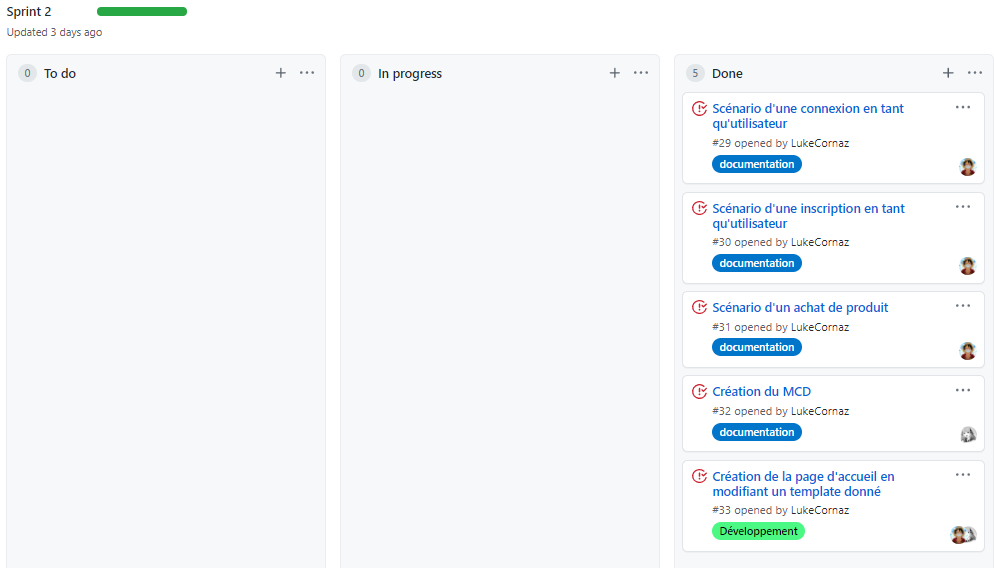
Nous utiliserons ensuite GitHub pour transmettre notre projet à notre professeur et à notre classe qui pourra essayer notre site et nous faire un retour sur les fonctionnalités. Nous demanderons à trois membres spécifiques de notre classe pour des teste plus poussée. Loik Meylan, Axel Pittet et Mahé Lavaud.

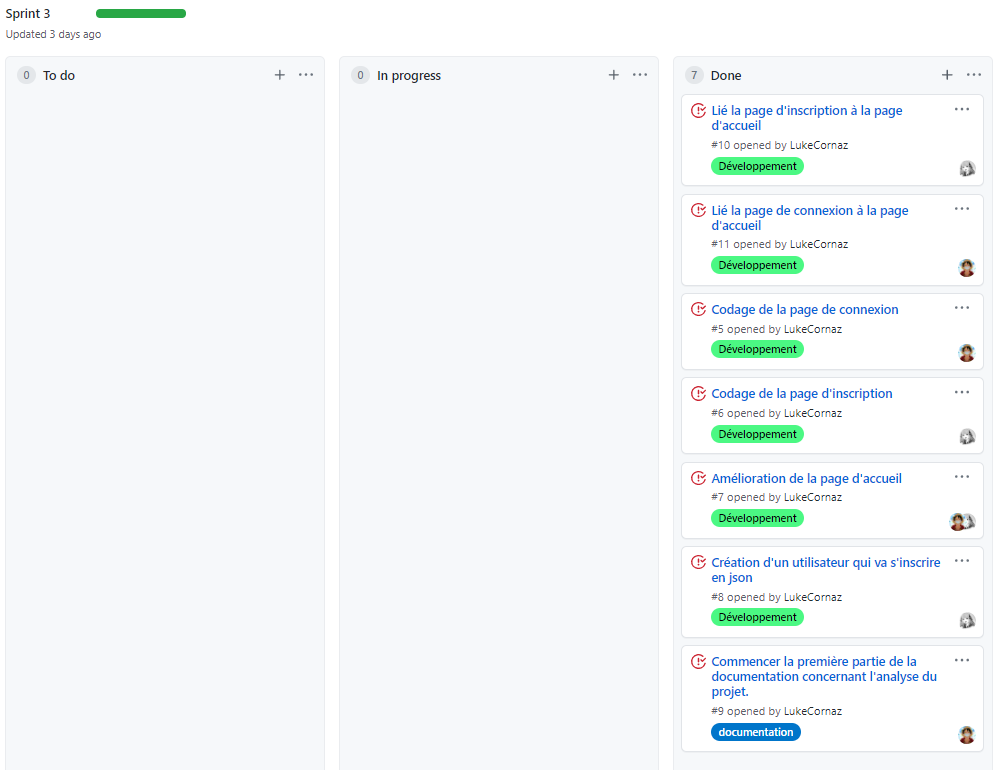
Pour les tests finaux nous ferons des tests de robustesse et de performance en inscrivant beaucoup d’utilisateur, pour voir si la base de donnée résiste bien.

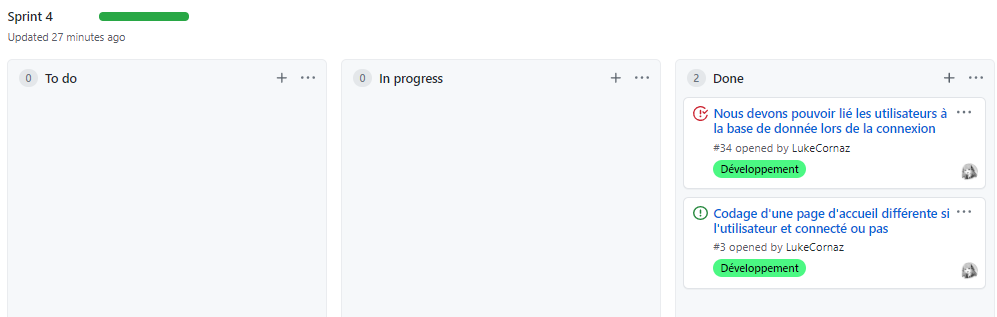
## Planification détaillée

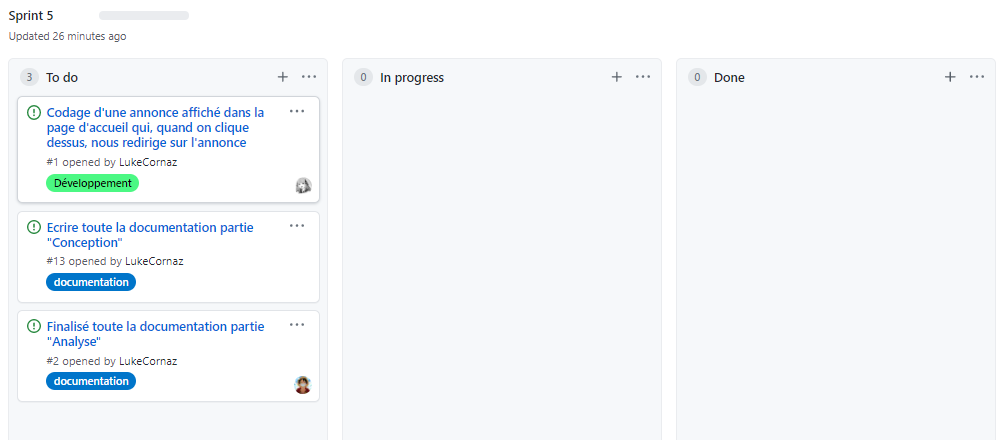
Voici la planification sous forme de sprint :

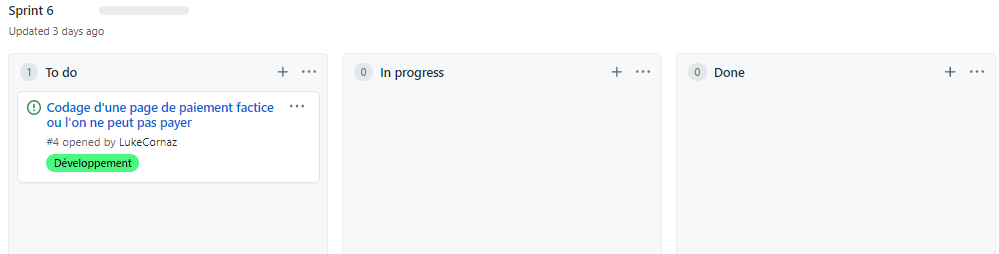


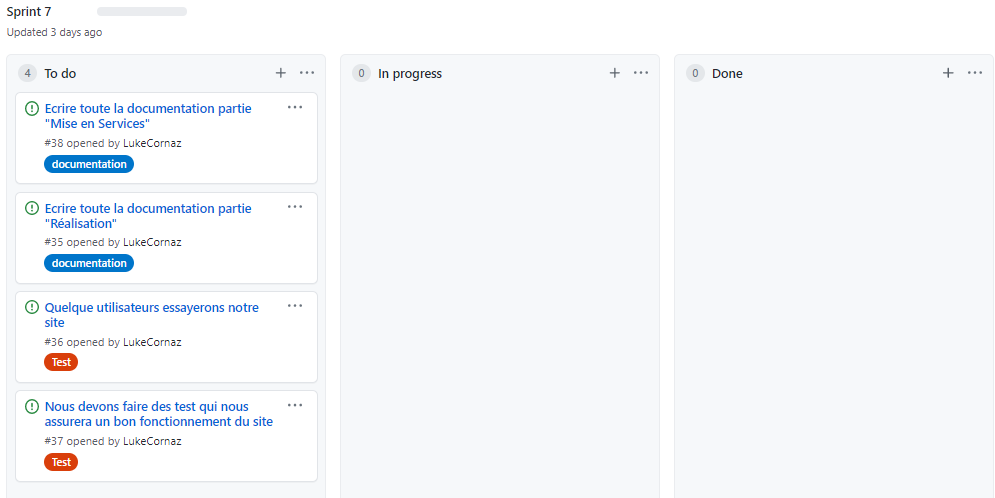


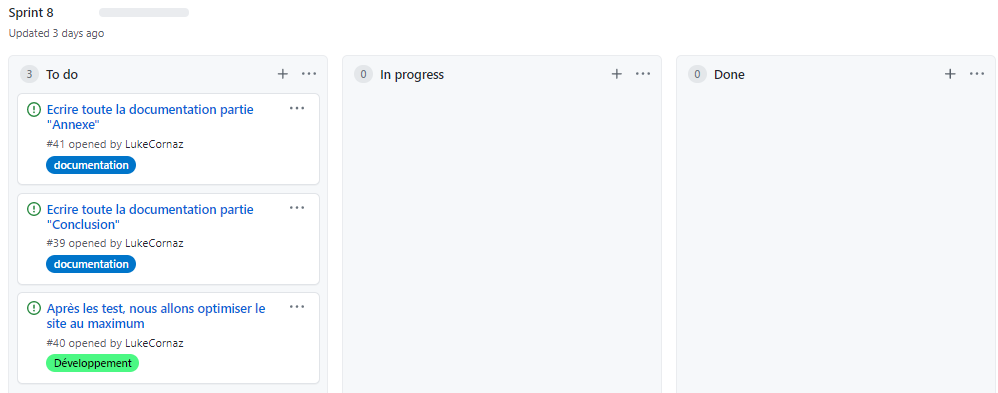












# Conception