

Projet module 151

Documentation de projet

Paccaud Samuel



Contenu

1	Analyse	3
1.1	Usecase	3
1.2	Les maquettes.....	4
1.3	Diagramme d'activités	6
1.4	Diagramme de séquence système	8
1.5	Base de données – entité-relation	9
2	Conception	9
2.1	Diagramme de classes partie clientes	9
2.2	Diagramme de classes partie serveur	10
2.3	Diagramme de séquence interactions	11
2.4	Base de données – schéma relationnel.....	13
3	Implémentation.....	15

1 Introduction

Le site web de mon projet sera une TODO-List, le principe de base est d'avoir différents projets avec différentes tâches qui ont des informations tels qu'un nom, ou une description. Les tâches ont un état et ils permettent de savoir l'avancer d'une tâche, ainsi que l'avancée du projet dans sa globalité.

Lorsqu'un visiteur arrivera sur l'accueil du site, il sera redirigé vers la page de connexion. Il n'aura pas le choix de se connecter pour accéder au site web. Si le visiteur ne possède pas de compte, il peut cliquer sur un lien en dessous pour se diriger vers la page de création de compte.

Une fois connecter, le client pourra accéder à tous les projets dont il fait partie, il peut aussi crée un nouveau projet dont il sera le seul à y être assigner. Dans chaque projet se trouve différentes tâches, nous pouvons crée une tâche, qui sera assigner par default dans l'état « TODO ». Il est possible de modifier le nom et la description de la tâche.

Les taches pourront changer d'état (TODO, In progress, Done, Validate). Chaque personne du projet peut créer ou, changer l'état des tâches, sauf pour l'état valider qui pourra être affecté par un administrateur.

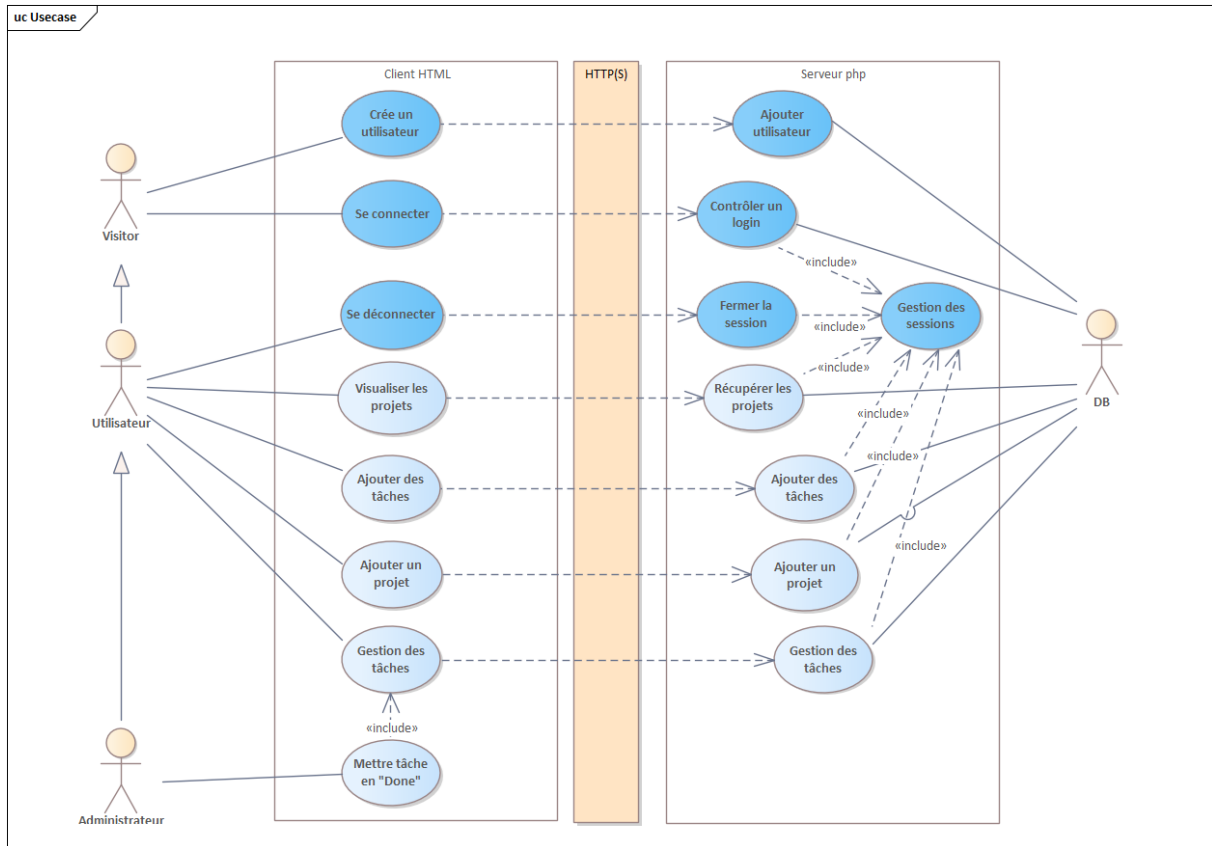
2 Analyse

Usecase

Voici le usecase de mon projet, voici à présent les actions de chaque acteur.

Acteur	Activité	Action
Visiteur	Un visiteur est une personne qui n'est pas connecter.	Se connecter à un compte utilisateur.
		Crée un compte utilisateur.
Utilisateur	Un visiteur qui se connecte à un compte deviens un utilisateur.	Se déconnecter de son compte, et donc revenir à la page de connexion.
		Visualiser les différents projets dont il fait partie.
		Ajouter une tâche au projet.
		Gérer l'état des tâches mais aussi leur description ou leur nom.
		Crée un nouveau projet
Administrateur	Un administrateur est un visiteur qui c'est connecter sur un compte admin.	Il possède les mêmes droits que l'utilisateur, sauf un droit en plus, qui est le pouvoir de mettre une tâche dans l'état valider.

Et voici le schéma Usecase.

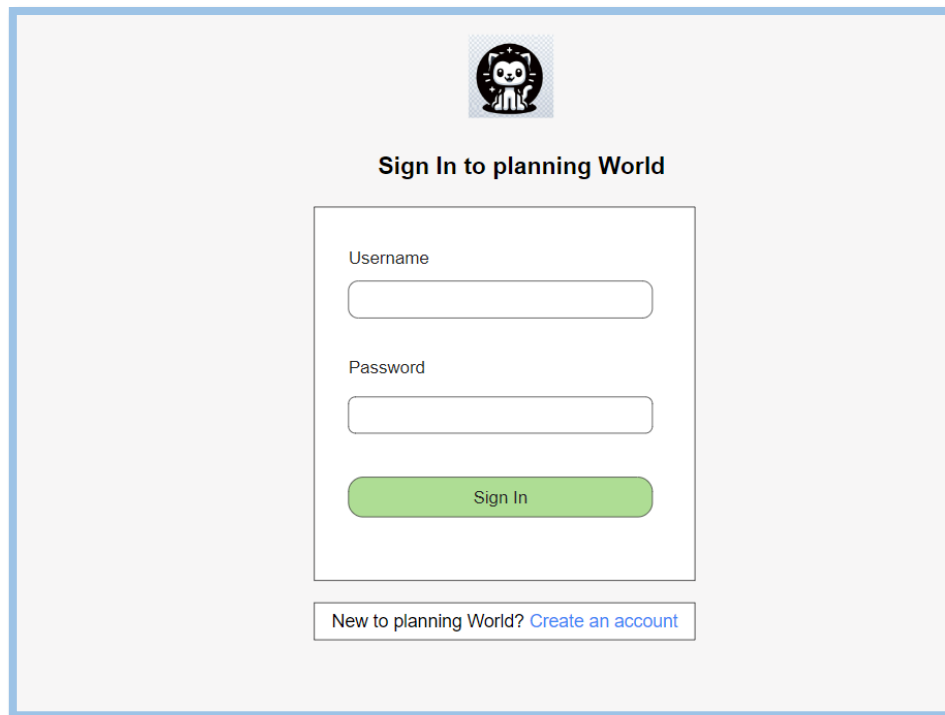


Les maquettes

1. Login

Cette page est une page de connexion, le visiteur doit spécifier son nom d'utilisateur et son mot de passe pour se connecter. Si le visiteur ne possède pas de compte utilisateur, il a la possibilité d'en créer un en cliquant sur « create an account ».

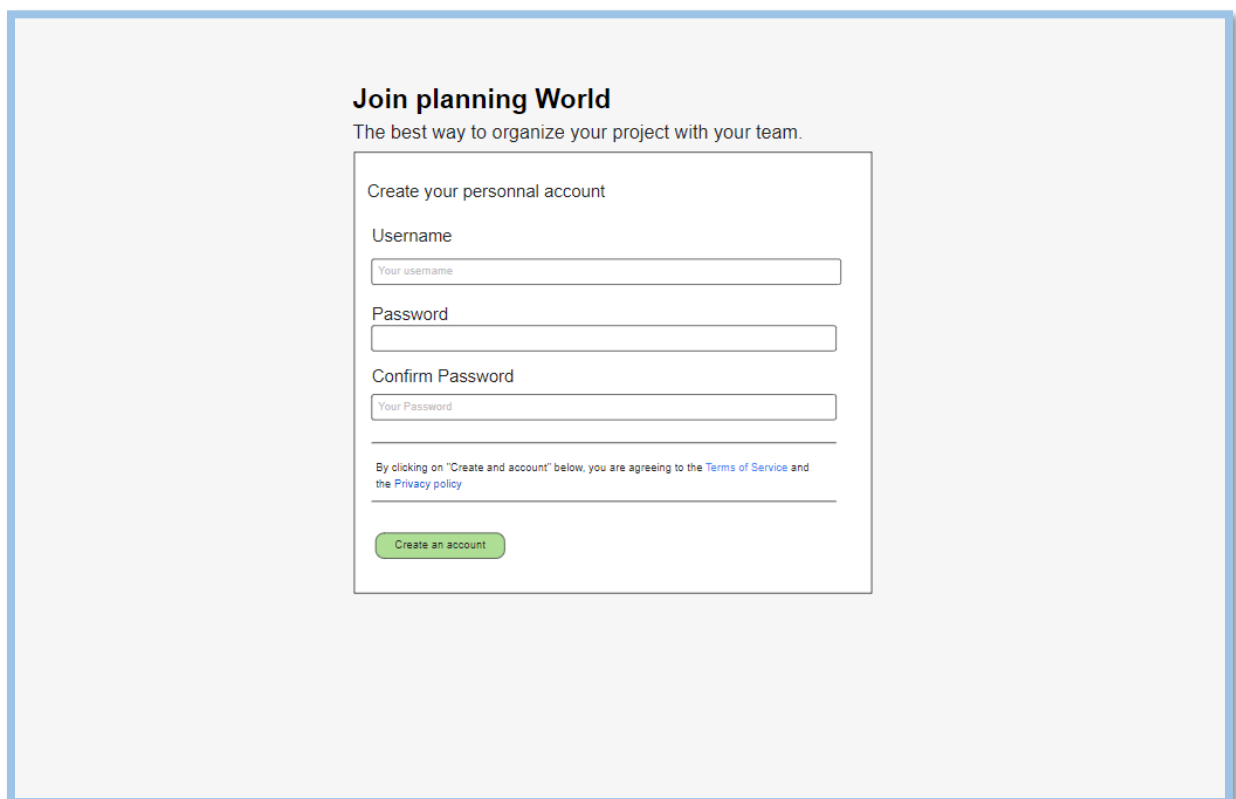
Pour se connecter à un utilisateur admin, il faut se connecter à un utilisateur possédant les permissions administrateur.



The image shows a sign-in form for 'planning World'. At the top center is a circular logo featuring a stylized cat with a star on its head. Below the logo, the text 'Sign In to planning World' is centered. The form itself is a white box with a thin border, containing two input fields: 'Username' and 'Password'. Below these fields is a green button with the text 'Sign In'. At the bottom of the form box, there is a link that says 'New to planning World? Create an account'.

2. Create account

Cette page permet à un visiteur de créer un compte utilisateur, il doit simplement spécifier le nom de l'utilisateur qu'il veut créer, ainsi que son mot de passe. Une fois les informations indiquées, il suffit de cliquer sur le bouton « Create an account ». Il est impossible de créer un utilisateur qui existe déjà.



The image shows a 'Join planning World' form. At the top, the text 'Join planning World' is followed by the tagline 'The best way to organize your project with your team.' Below this is a white form box with a thin border. Inside the box, the text 'Create your personal account' is at the top. There are three input fields: 'Username' (with placeholder text 'Your username'), 'Password', and 'Confirm Password' (with placeholder text 'Your Password'). Below the input fields, there is a line of text: 'By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy policy](#)'. At the bottom of the form box is a green button with the text 'Create an account'.

3. Gestion de projet

C'est ici que les utilisateurs peuvent accéder au gestionnaire de projets. En cliquant sur le menu déroulant, l'utilisateur peut voir tous les projets sur lesquelles ils travaillent. Il peut changer l'état d'une tâche en « drag and drop » vers un autre état. L'utilisateur peut aussi ajouter de nouvelles tâches en cliquant sur « add new task ». La nouvelle tâche apparaîtra dans l'état « TODO ».

Le nom ou la description des tâches peut aussi être modifié.

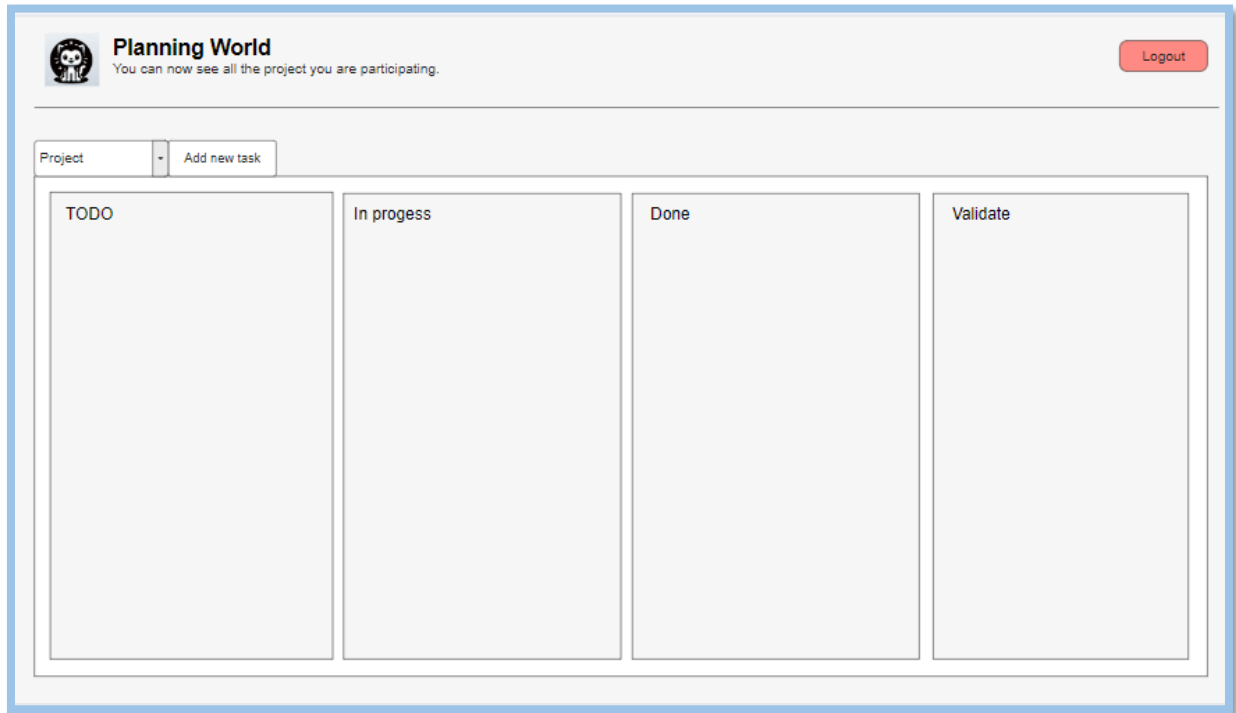
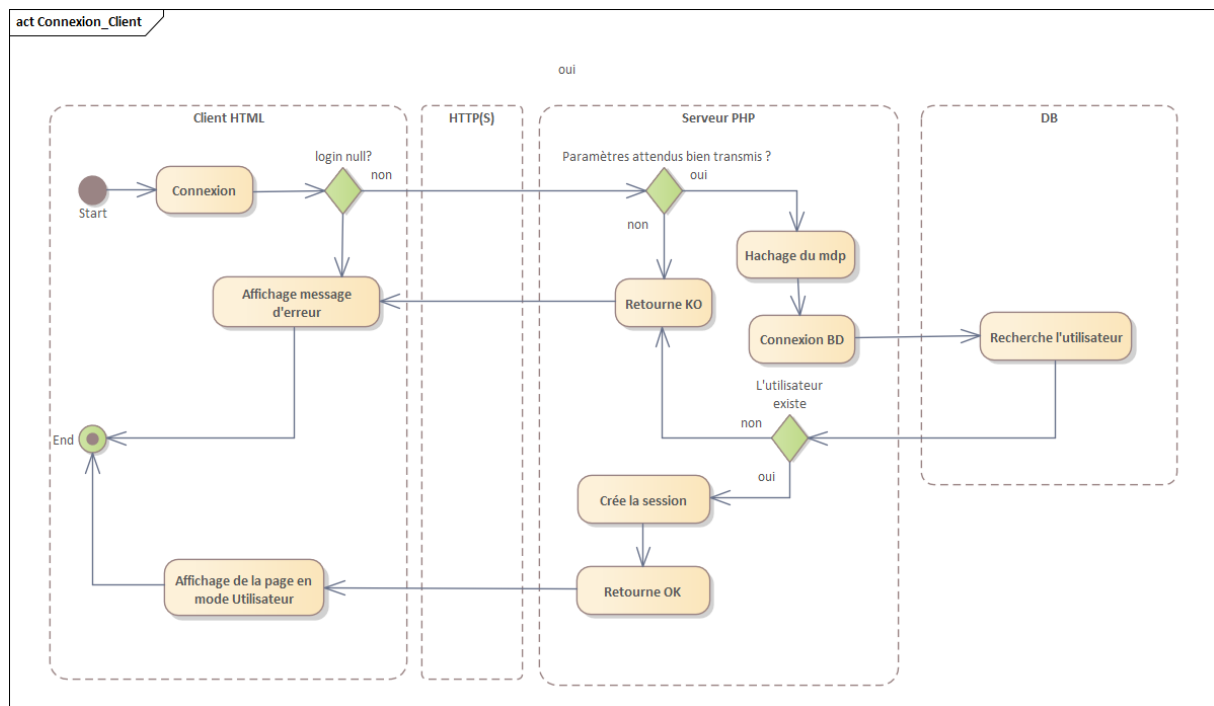
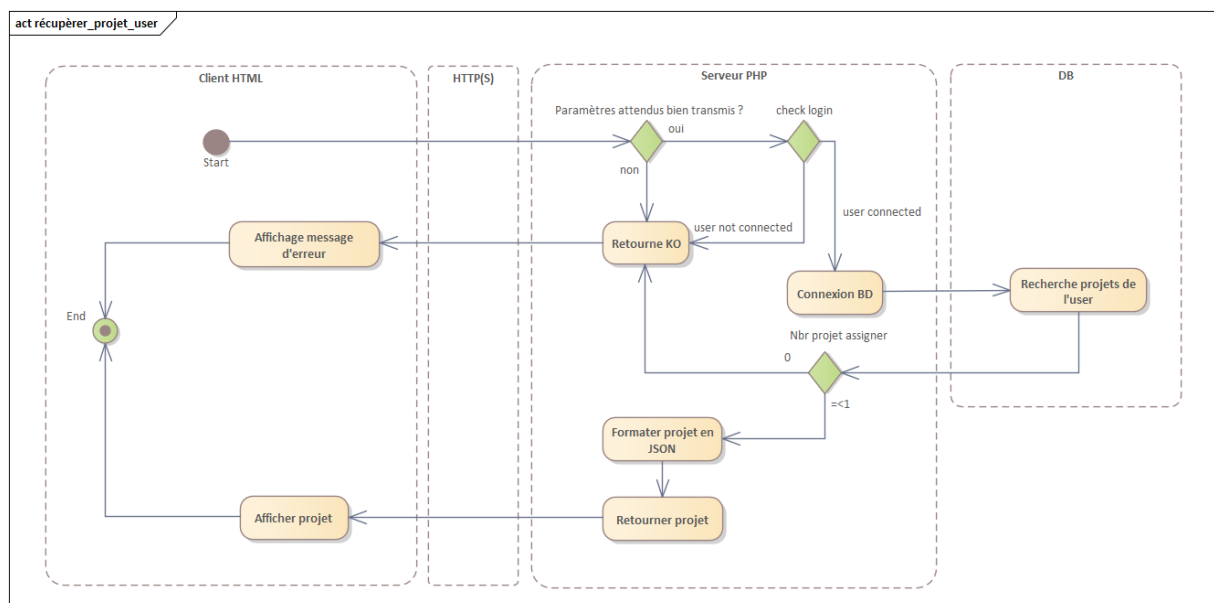


Diagramme d'activités

Voici le diagramme d'activité pour se connecter à un compte pour notre application. Nous vérifions d'abord si le login n'est pas vide (Null), puis nous envoyons au serveur PHP le nom d'utilisateur et le mot de passe. Ensuite nous hachons le mot de passe, et envoyons à la DB les informations pour vérifier si nous pouvons nous connecter. Si nous pouvons nous connecter à l'utilisateur, nous créons une session et affichons la page client. Sinon nous envoyons un message d'erreur au visiteur.



Pour récupérer les projets de l'utilisateur, nous allons d'abord envoyer au serveur PHP le nom de l'utilisateur connecter. Ensuite nous allons nous connecter à la DB, puis rechercher tous les projets sur lesquels l'utilisateur participe. La liste va être formater en JSON, envoyé au client et afficher sur l'interface web.



Un autre diagramme pour changer l'état d'une tâche, nous devons d'abord regarder dans qu'elle état nous allons déplacer notre tâche, si la tâche est déplacée dans l'état « DONE », il faut vérifier si notre utilisateur est admin. Ensuite nous allons envoyer au serveur dans qu'elle projet nous faisons les modifications, ainsi que dans qu'elle tâche nous déplaçons. Il faut aussi envoyer dans qu'elle était-elle se trouvait, et dans qu'elle état la tâche va se retrouver.

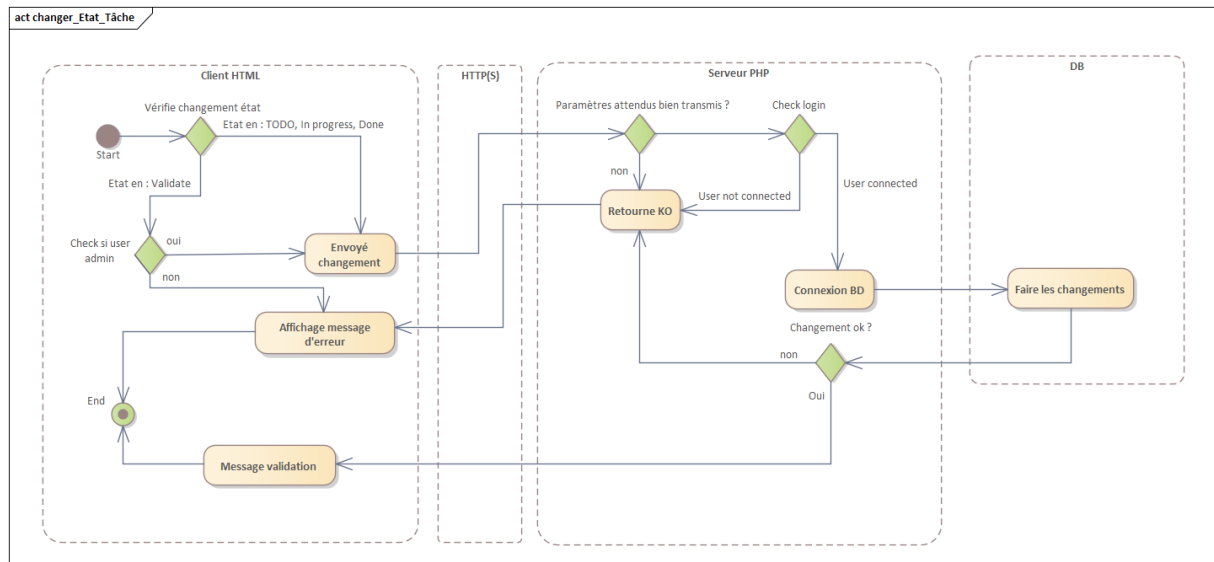
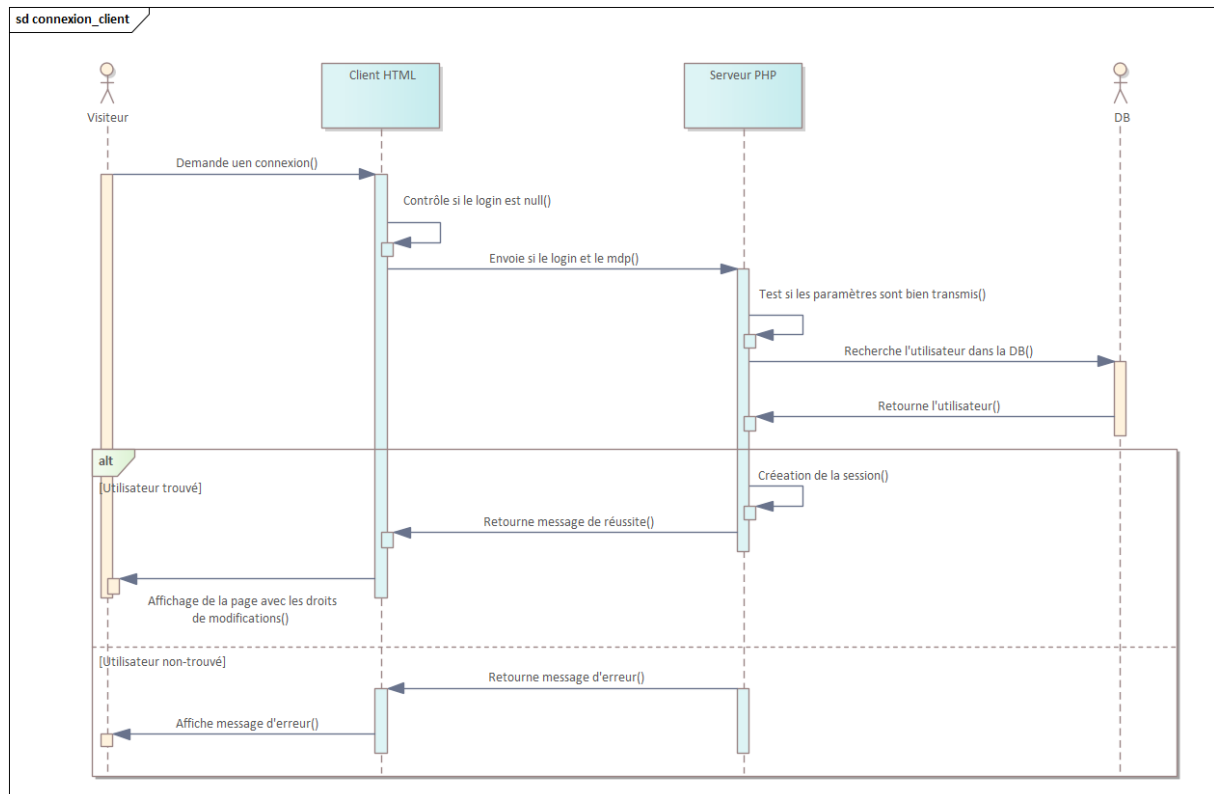
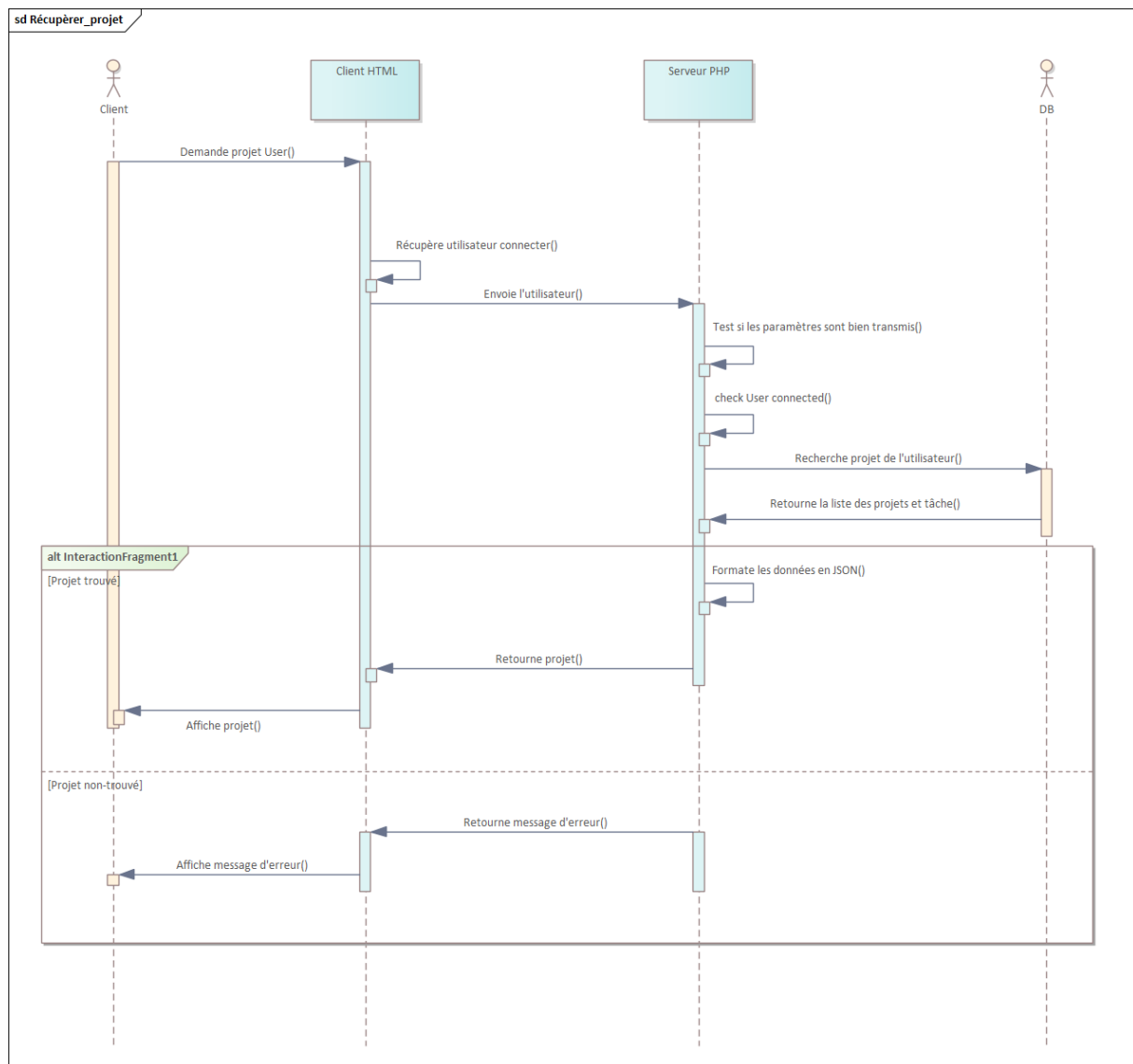


Diagramme de séquence système

Voici le diagramme séquence système pour la connexion à un utilisateur.

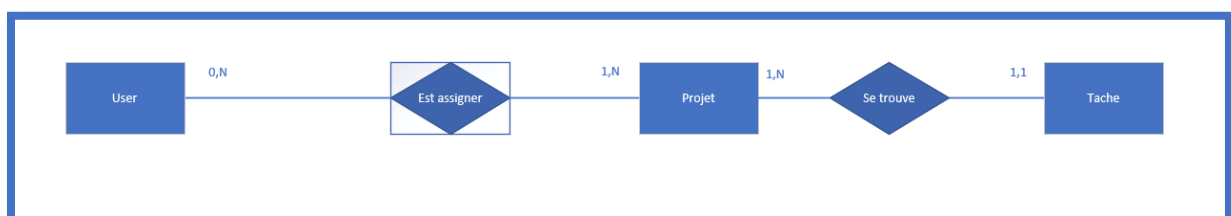


Voici le diagramme pour récupérer les projets d'un utilisateur.



Base de données – entité-relation

Voici le schéma entité relation de ma base de données. Une table de relation entre la table utilisateurs et la table projet a été ajoutée, car plusieurs utilisateurs peuvent travailler sur un projet. Cependant c'est fonctionnalité n'a pas encore mis en place, mais la table de relation a déjà été mis en place pour ajouter cette fonctionnalité dans un futur proche.



3 Conception

Diagramme de classes partie clientes

Voici le diagramme de classe pour la partie cliente, nous pouvons voir qu'il n'y a pas d'inex.HTML, car lorsque nous arriverons sur le siteWeb, nous serons dirigés directement sur connexion.html

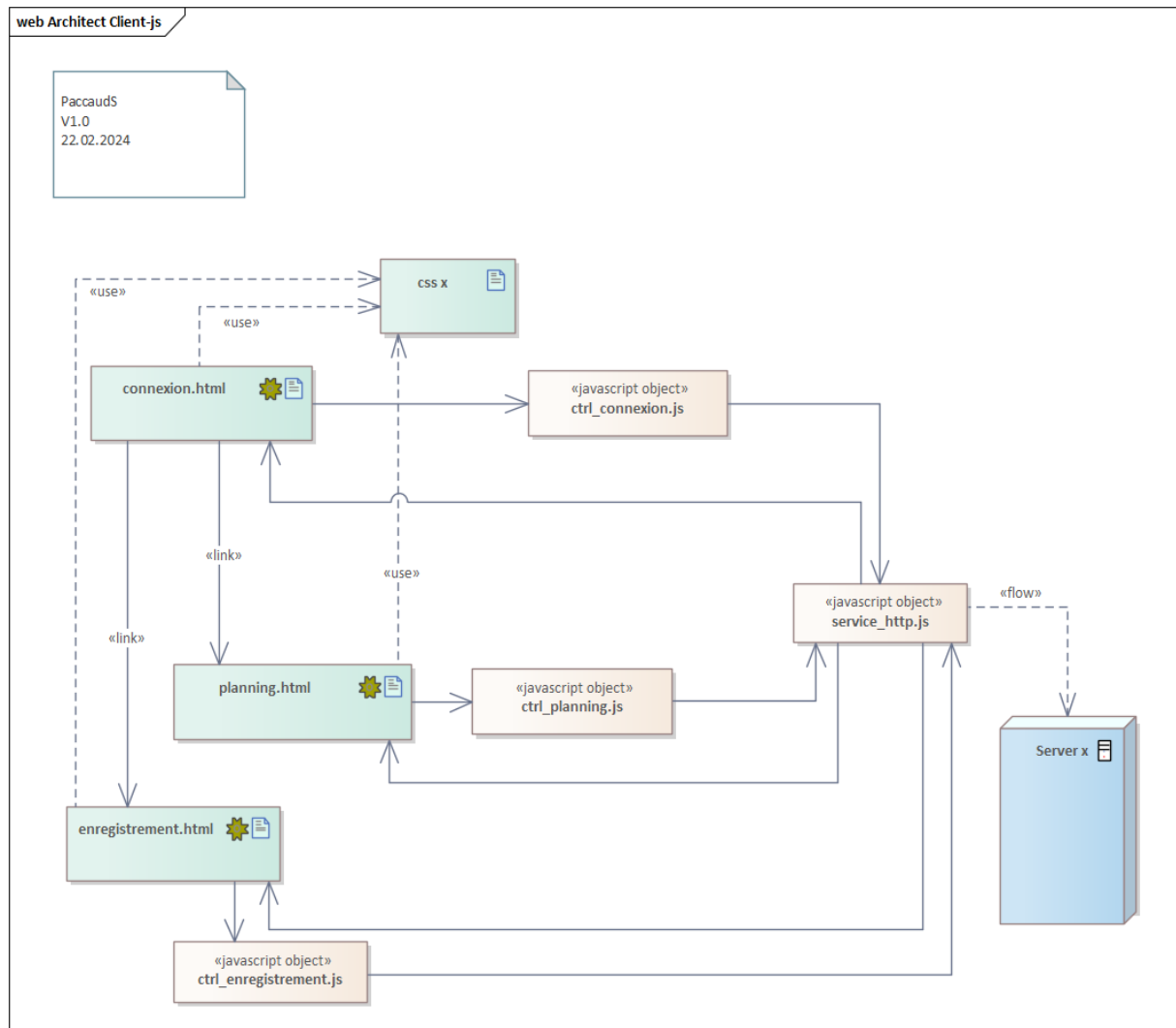


Diagramme de classes partie serveur

Et voici le diagramme du côté serveur.

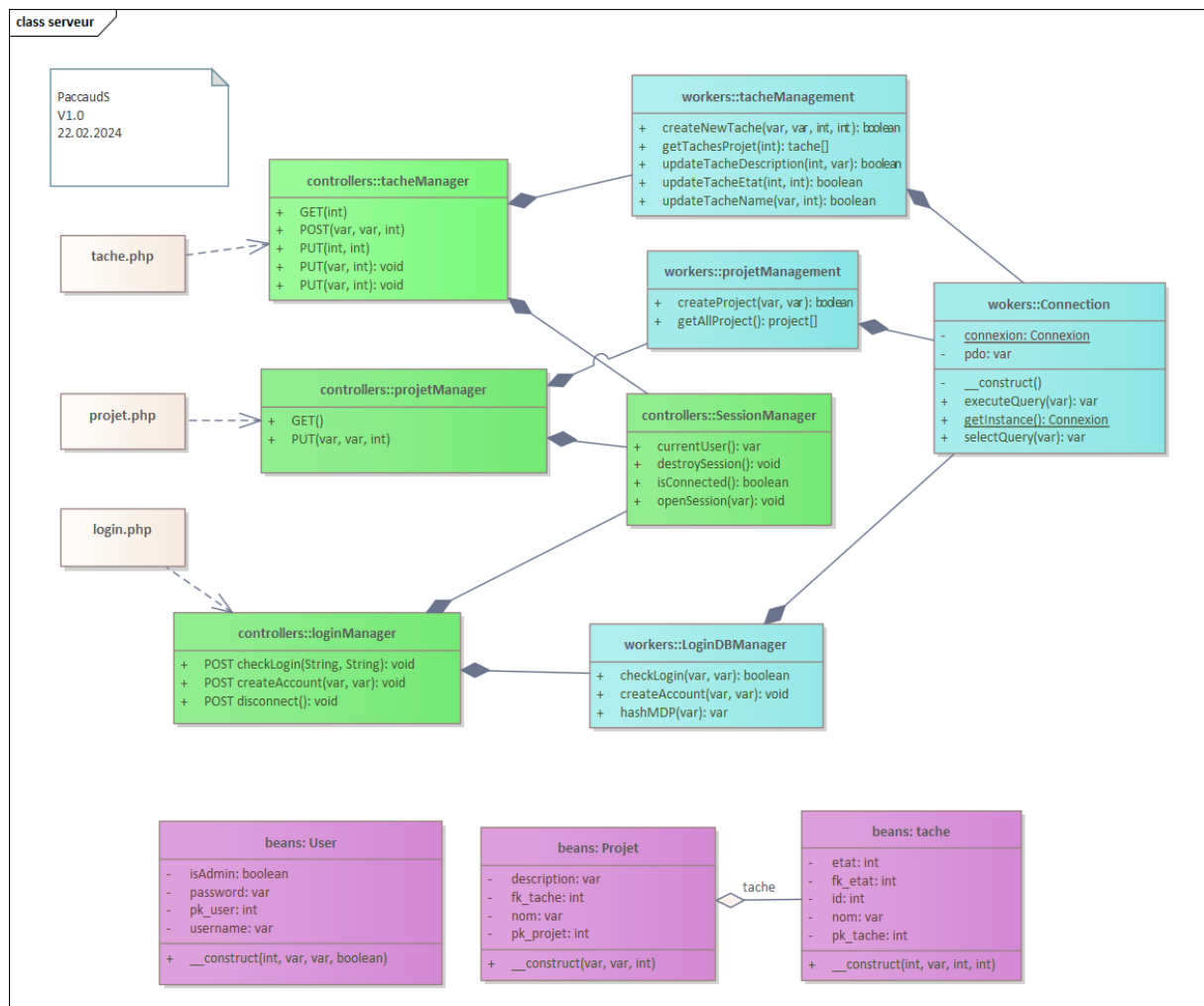
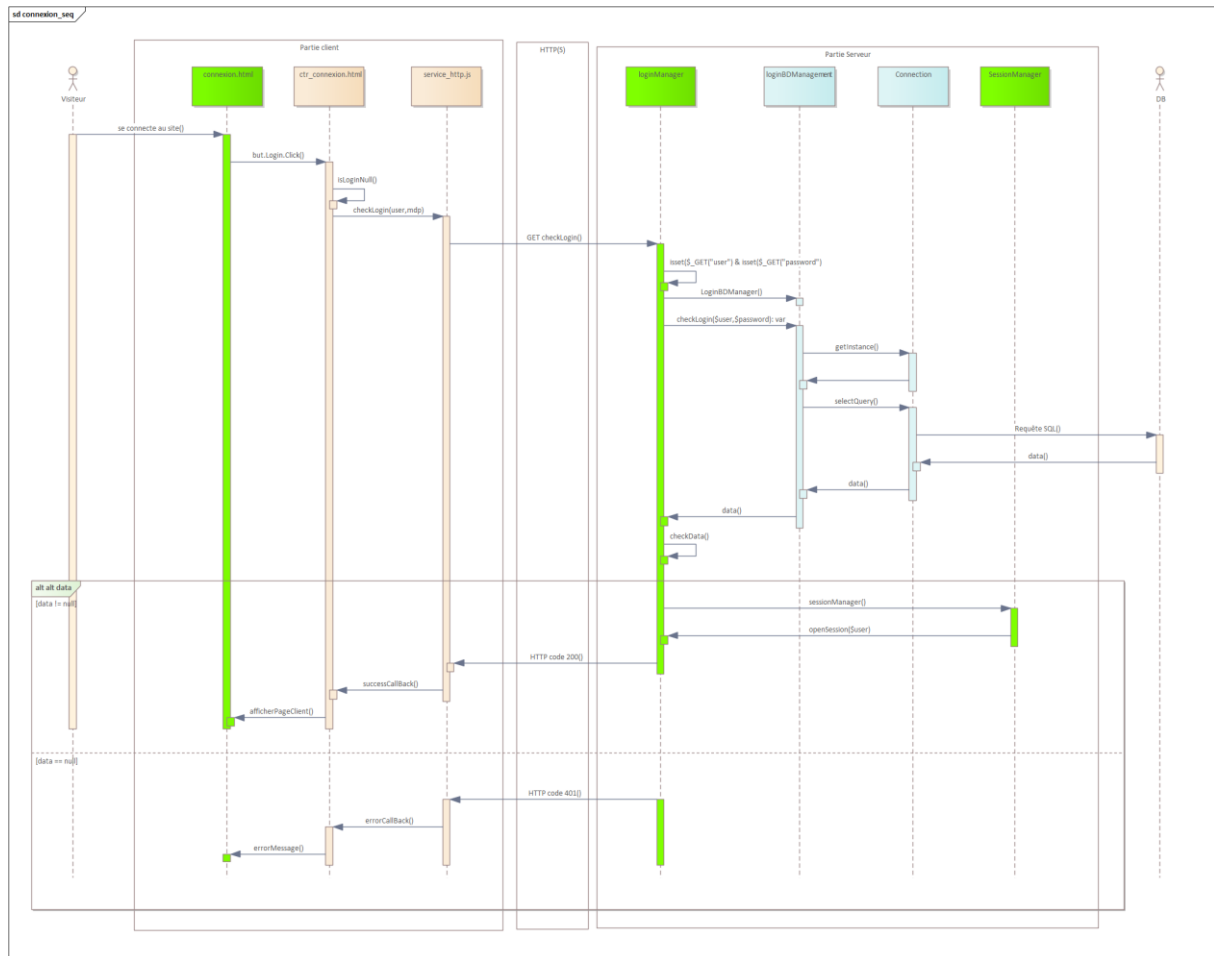
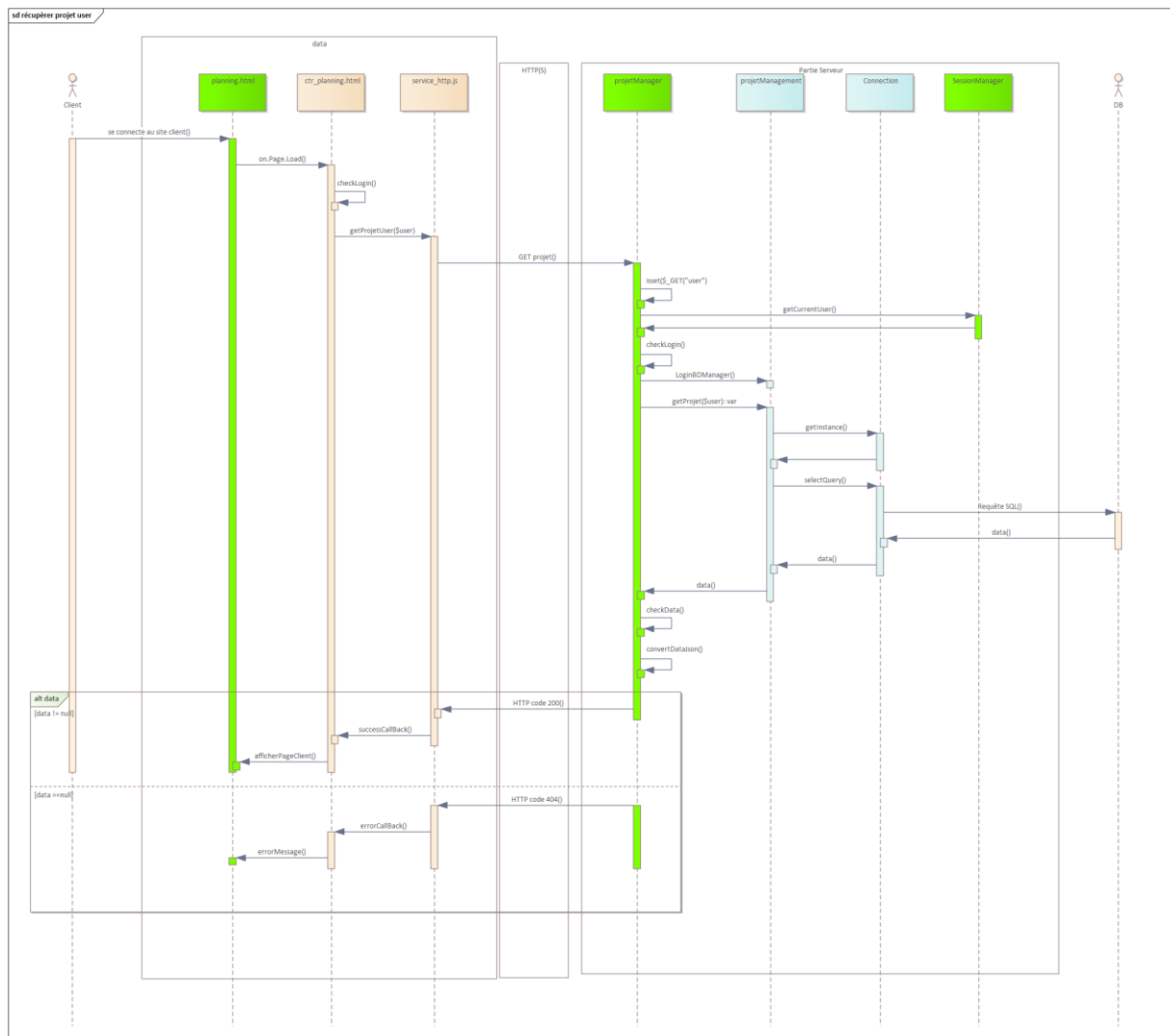


Diagramme de séquence interactions

Voici le diagramme de séquence d'interaction pour la connexion d'un visiteur.

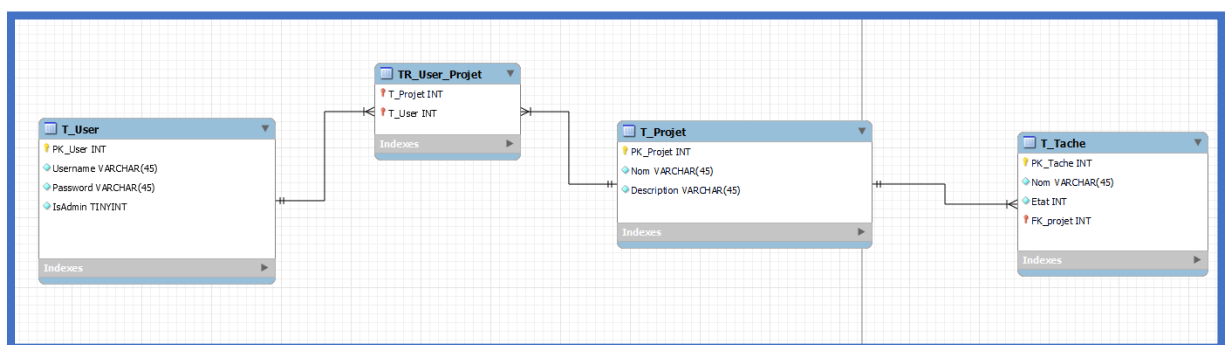


Et voici le diagramme pour la partie ou il faut récupérer les projets d'un utilisateur.



Base de données – schéma relationnel

Voici le schéma relationnel de la DB.



Conception des tests

Pour ce qui est des tests, voici les pas à effectuer pour vérifier la sécurité ainsi que le fonctionnement du site.

Voici les tests pour vérifier le fonctionnement de mon application :

Fonctionnement	Page ?	Résultat attendu	Résultat obtenu	Test validé ?
Crée un nouvel utilisateur	Register.ht ml	Un utilisateur devrait se crée et être stocker sur la DB,		
Crée un utilisateur qui est déjà existant	Register.ht ml	Un message d'erreur devrait apparaitre.		
Se connecter à un compte existant avec bon nom d'utilisateur et mot de passe.	Login.html	La connexion devrait réussir, et la page « planning.html » devrait s'ouvrir.		
Se connecter à un compte existant avec mauvais mot de passe ou nom d'utilisateur inexistant.	Login.html	Un message d'erreur devrait apparaitre.		
Recevoir tous les projets d'un utilisateur.	Planning.ht ml	Tous les projets qui sont assigner à l'utilisateur doivent être afficher dans le menu déroulant.		
Création d'une nouvelle tâche.	Planning.ht ml	Une tâche doit se crée dans l'état « todo ». Elle doit aussi être stocker dans la DB.		
Modifié le nom / la description d'une tâche.	Planning.ht ml	La tâche va changer de nom / description, le changement dois aussi changer dans la DB.		
Déplacer une tâche dans les états « TODO, In Progress, Done ».	Planning.ht ml	La tache doit changer d'état, le changement dois aussi changer dans la DB.		
Déplacer une tâche dans l'état « Validate » en tant qu'user non-administrateur.	Planning.ht ml	Un message d'erreur devrait apparaitre, et la tâche ne dois pas changer d'état.		
Déplacer une tâche dans l'état « Validate » en tant qu'user administrateur.	Planning.ht ml	La tache doit changer d'état, le changement dois		

		aussi changer dans la DB.		
Déconnection d'un utilisateur.	Planning.html	L'utilisateur doit se faire déconnecter.		

Et voici les testes pour tester la sécurité de mon site web.

Fonctionnement	Page ?	Résultat attendu	Résultat obtenu	Test validé ?
Vérifier si des injections SQL / HTML sont possible sur les champs de création de compte.	Register.html	Les injections de devraient pas fonctionner.		
Vérifier si des injections SQL / HTML sont possible sur les champs de connexion.	Connexion.html	Les injections de devraient pas fonctionner.		
Vérifier qu'avec un outil comme « POSTMAN », les requêtes SQL ne fonctionne pas.	Register.html / Connexion.html / Planning.html	Le site devrait remarquer qu'aucun user n'est connecter, et les requêtes ne devrait pas être prit en compte.		

4 Implémentation