

151 - Intégrer des bases de données dans le Web

Rapport personnel

Kevin Martin

01.02.2024 au 08.03.2024

Date de création : 09.02.2024

Version 1 du 10.03.2024

Table des matières

1	Introduction	3
1.1	Objectifs du module	3
2	Analyse	4
2.1	Présentation du projet	4
2.1.1	Fonctionnalités :	4
2.2	Uses Case	5
2.3	Maquettes	6
2.3.1	Page d'accueil	6
2.3.2	Popup d'inscription	6
2.3.3	Popup de connexion	7
2.3.4	Page liste des projets	7
2.3.5	Page projet	8
2.3.6	Popup ajout de tâche	8
2.3.7	Popup ajout de poste	9
2.4	Diagramme activité	9
2.4.1	NewPost	9
2.4.2	NewTask	10
2.5	Diagramme de séquences systèmes	11
2.5.1	NewPost	11
2.5.2	NewTask	12
2.6	Schéma ER	13
3	Conception	14
3.1	Diagramme de classe	14
3.1.1	Client	14
3.1.2	Serveur	15
3.2	Schéma relationnel	16
3.3	Diagramme séquence interactions	17

3.4	Conception des tests	17
3.4.1	Injections	17
3.4.2	Postmann	18
3.4.3	Fonctionnalités	19
4	Implémentation	20
4.1	Descente de code	20
4.1.1	Interface	20
4.1.2	Client	20
4.1.2.1	<i>Project.tsx</i>	20
4.1.2.2	<i>NewTask.tsx</i>	21
4.1.2.3	<i>httpServices.tsx</i>	23
4.1.3	Serveur	23
4.1.3.1	<i>Server.php</i>	23
4.1.3.2	<i>TaskCtrl.php</i>	23
4.1.3.3	<i>TaskDBManager.php</i>	24
4.1.3.4	<i>Connexion.php</i>	24
4.1.3.5	<i>Task.php</i>	25
4.2	Problèmes rencontrés	26
4.2.1	Problème #1 base de données	26
4.2.2	Problème #2 cookies de session	26
4.3	Tests fonctionnels	26
4.3.1	Injections	26
4.3.2	Postmann	27
4.3.3	Fonctionnalités	28
4.4	Hébergement	28
5	Synthèse	29
5.1	Conclusion	29

1 Introduction

La compétence principale à acquérir durant ce module est la suivante :

- ⇒ Intégrer une base de données existante respectivement des éléments de données dans une application Web en tenant compte des exigences techniques et de sécurité.

1.1 Objectifs du module

Les objectifs de ce module sont les suivants :

- ⇒ Analyser les exigences d'une application Web et de la base de données, respectivement des éléments de données à lier, définir et documenter la technique de liaison.
- ⇒ Identifier les informations importantes de protection et de sécurité en tenant compte de la protection des données, et définir les mesures.
- ⇒ Réaliser l'intégration de l'application Web avec la base de données, respectivement aux éléments de données, en prêtant attention aux transactions, à la protection et la sécurité des données.
- ⇒ Mettre en œuvre les souhaits de modifications conformément au déroulement prescrit des modifications.
- ⇒ Définir et exécuter la procédure de test et de remise, la documenter dans un procès-verbal de tests. Si nécessaire, entreprendre les corrections.

2 Analyse

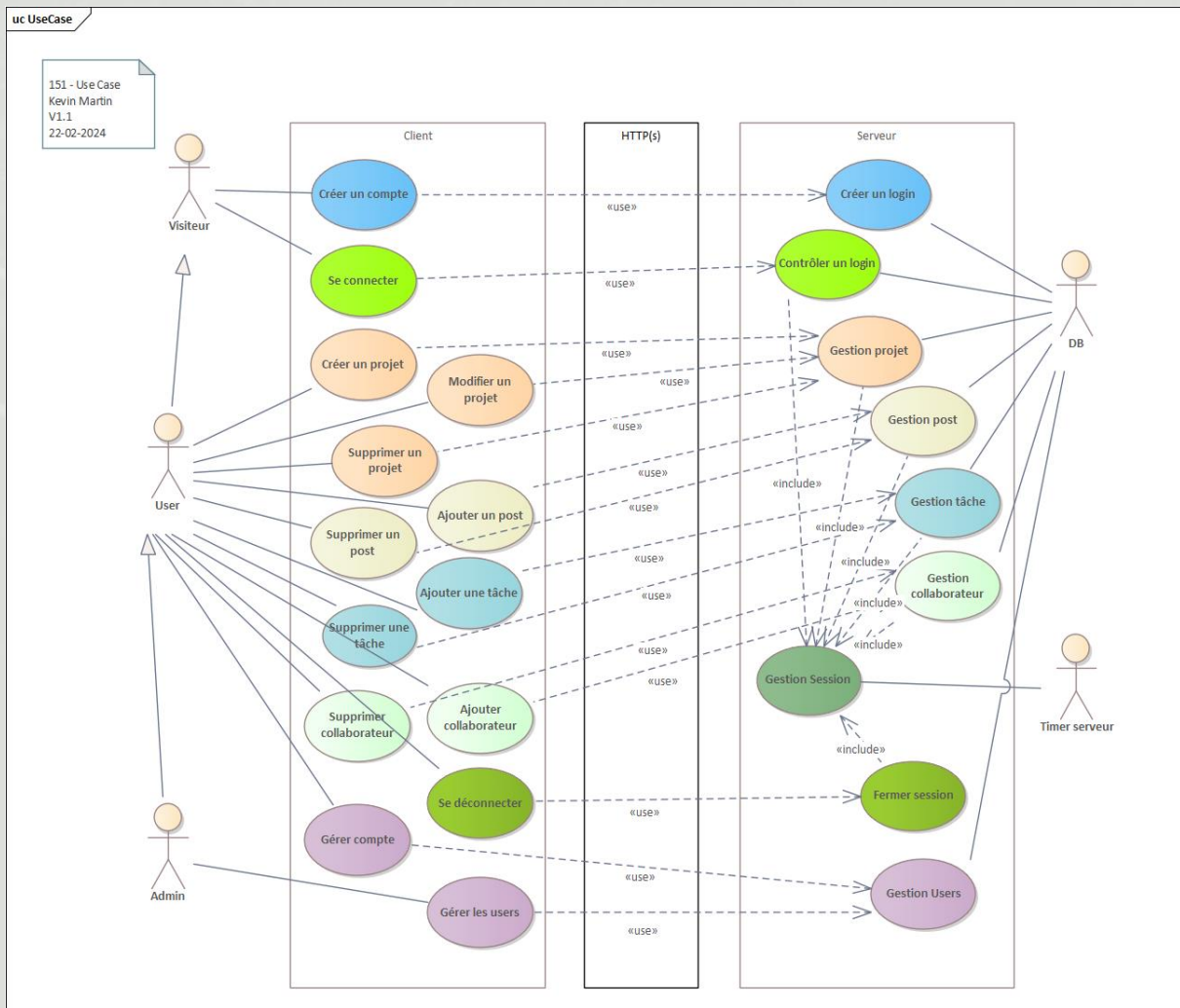
2.1 Présentation du projet

Le projet consiste en un site web permettant de gérer un projet (pro, événement, etc.). Le principe est de pouvoir créer un projet et lui affecter des tâches ainsi qu'avoir la possibilité de poster des commentaires soit en rapport avec le projet, soit en rapport avec une tâche précise. Nous avons également la possibilité de collaborer avec des autres utilisateurs en les invitant sur le projet afin que ceux-ci puissent créer des tâches et poster des commentaires.

2.1.1 Fonctionnalités :

- ⇒ Visiteurs
 - ⇒ Créer un compte
- ⇒ Utilisateurs
 - ⇒ Se connecter/déconnecter
 - ⇒ Gérer son compte
 - ⇒ Créer/modifier/supprimer un projet
 - ⇒ Créer/supprimer une tâche
 - ⇒ Poster/supprimer un commentaire en choisissant s'il est pour le projet ou pour une tâche précise en la sélectionnant
 - ⇒ Inviter/enlever des utilisateurs à collaborer sur son projet
- ⇒ Administrateur
 - ⇒ Gérer les utilisateurs

2.2 Uses Case

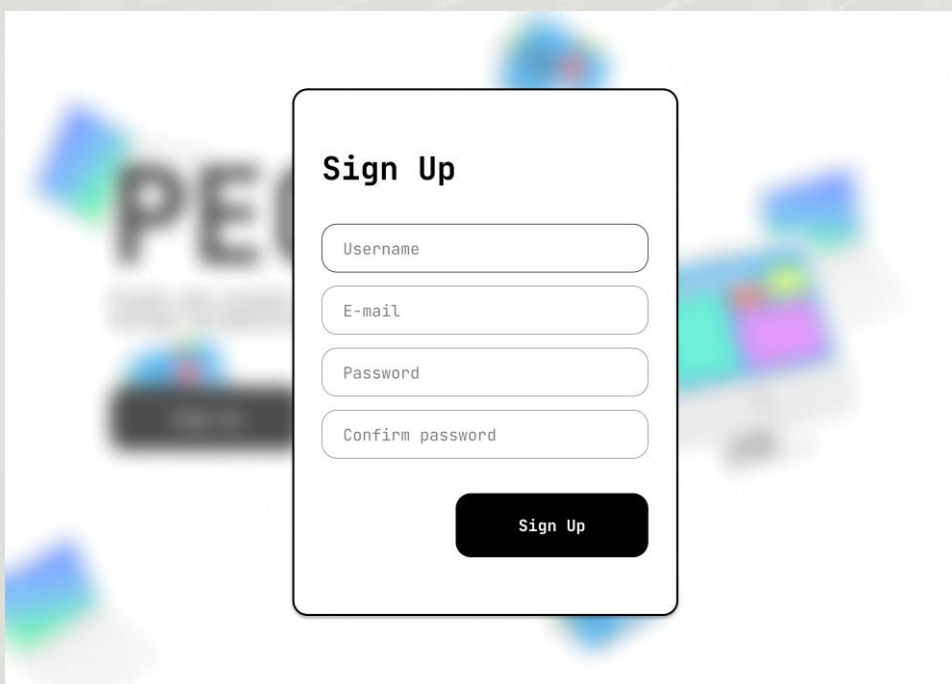


2.3 Maquettes

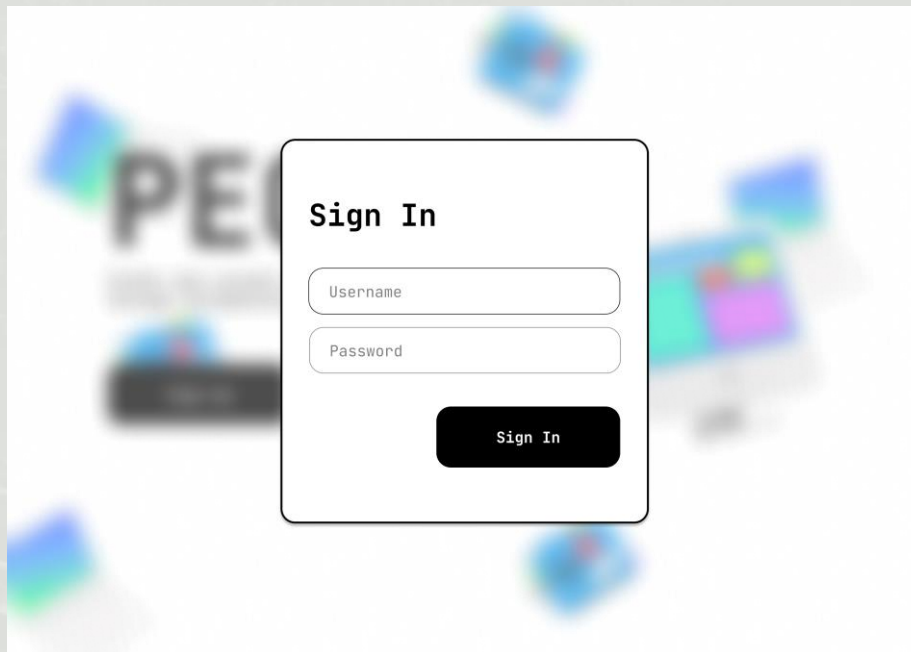
2.3.1 Page d'accueil



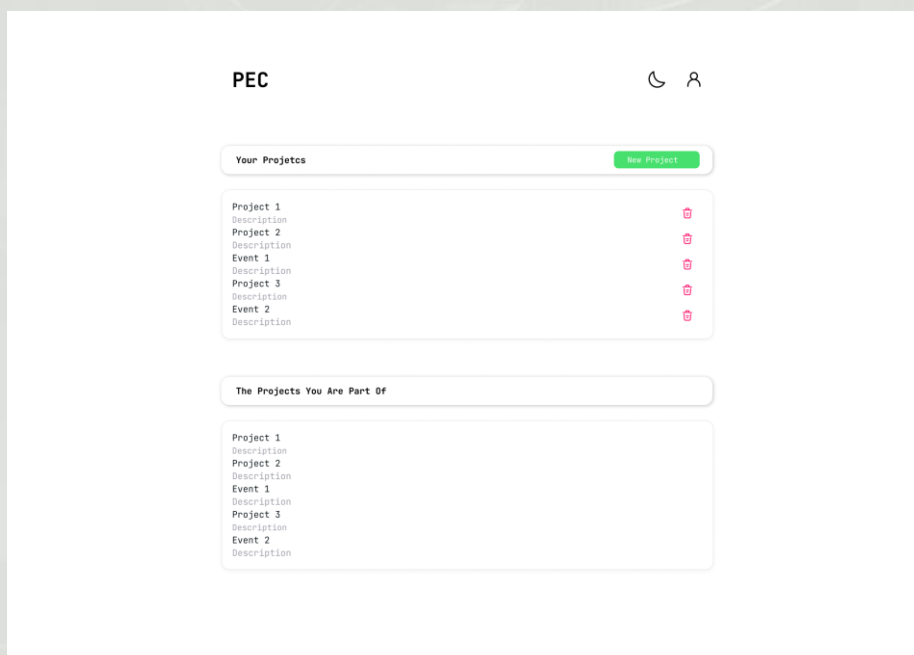
2.3.2 Popup d'inscription



2.3.3 Popup de connexion



2.3.4 Page liste des projets



2.3.5 Page projet

PEC

Participants :

+

Nom du projet/event

Description...

New Post

Your Task

New Task

Terminé

Titre De La Tâche

Petite Description

A Faire

Titre De La Tâche

Petite Description

Terminé

Titre De La Tâche

Petite Description

En Cours

Titre De La Tâche

Petite Description

Terminé

Titre De La Tâche

Petite Description

MichMich94

2 J.

Commentaire...

..

...

...

.

MichMich94

2 J.

Commentaire...

..

...

...

.

MichMich94

2 J.

Commentaire...

..

...

...

.

MichMich94

2 J.

Commentaire...

..

...

...

.

2.3.6 Popup ajout de tâche

New task

2.3.7 Popup ajout de poste

New post

Affecter à une tâche ou au projet/event



Commentaire...

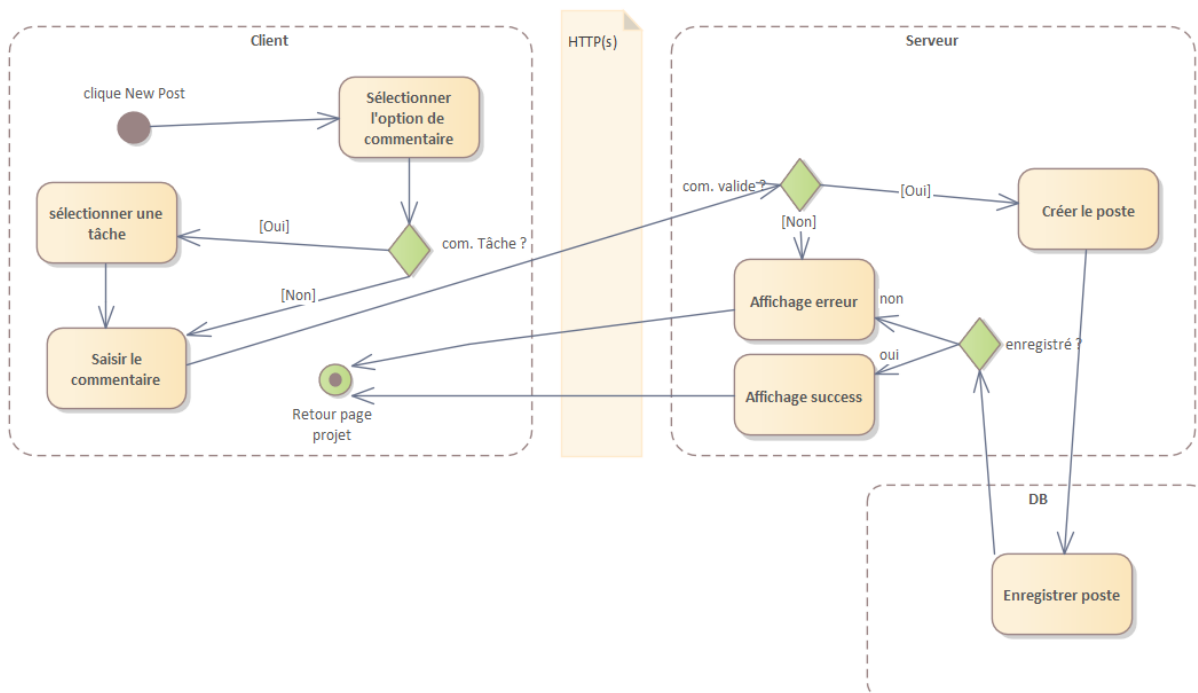
Poster

2.4 Diagramme activité

2.4.1 NewPost

act ActivityNewPost

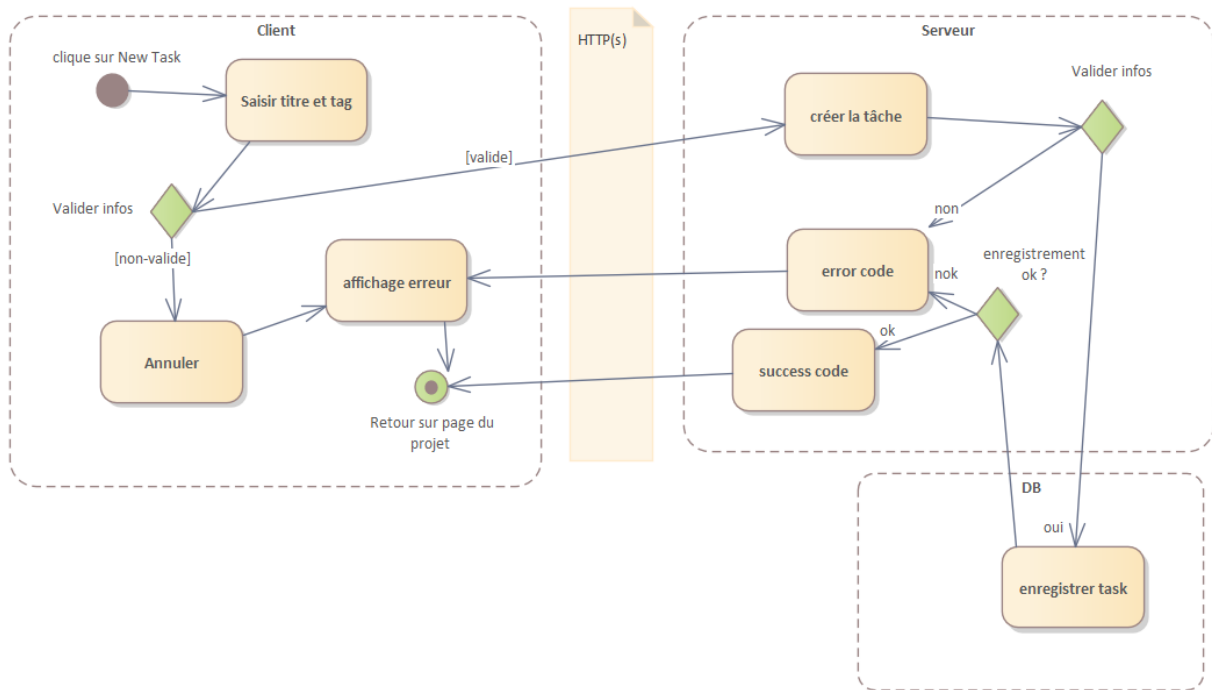
151 - diagramme
d'activité New Post
Kevin Martin
V2.0
03-03-2024



2.4.2 NewTask

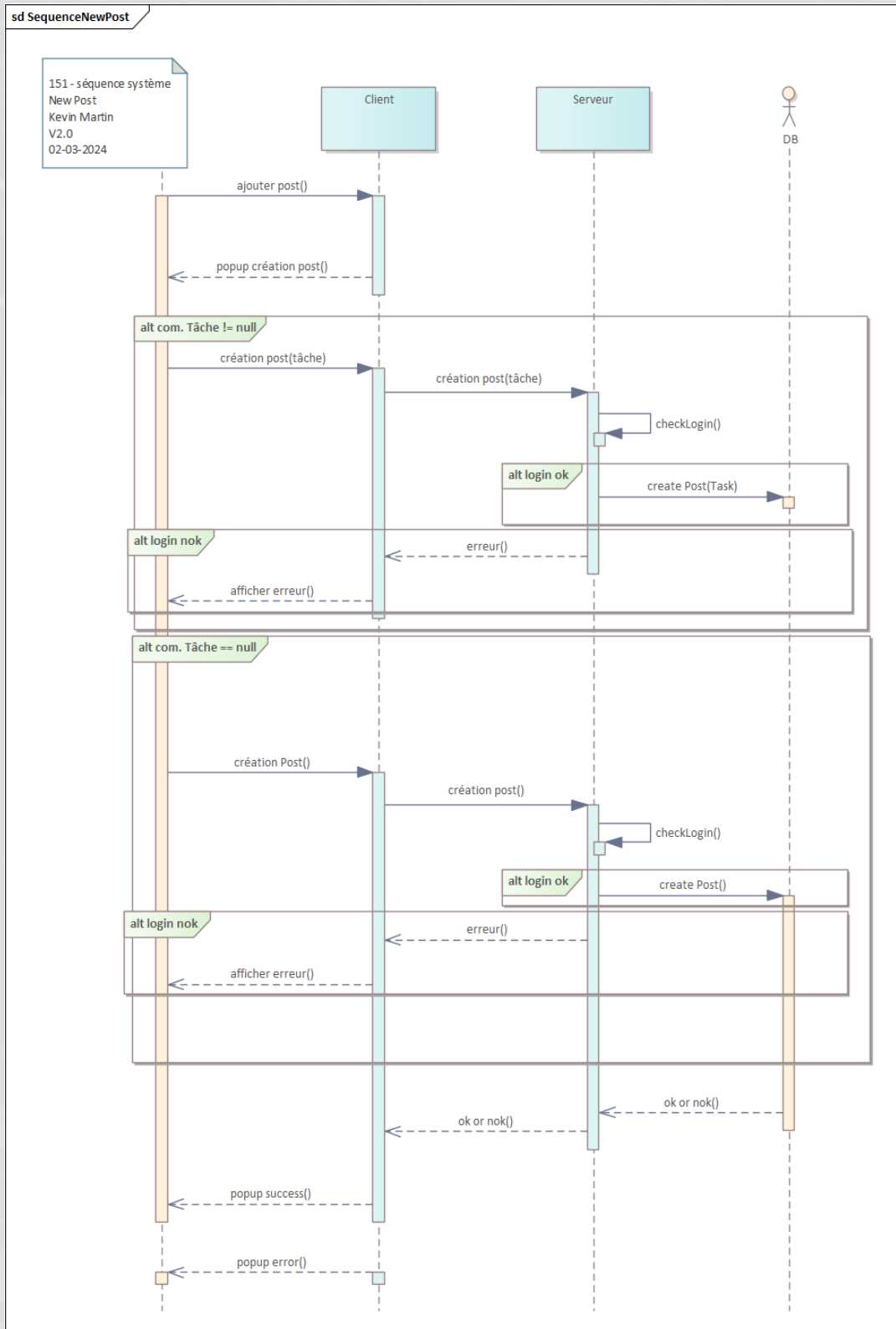
act ActivityNewTask

151 - diagramme
d'activité New Task
Kevin Martin
V2.0
03-03-2024



2.5 Diagramme de séquences systèmes

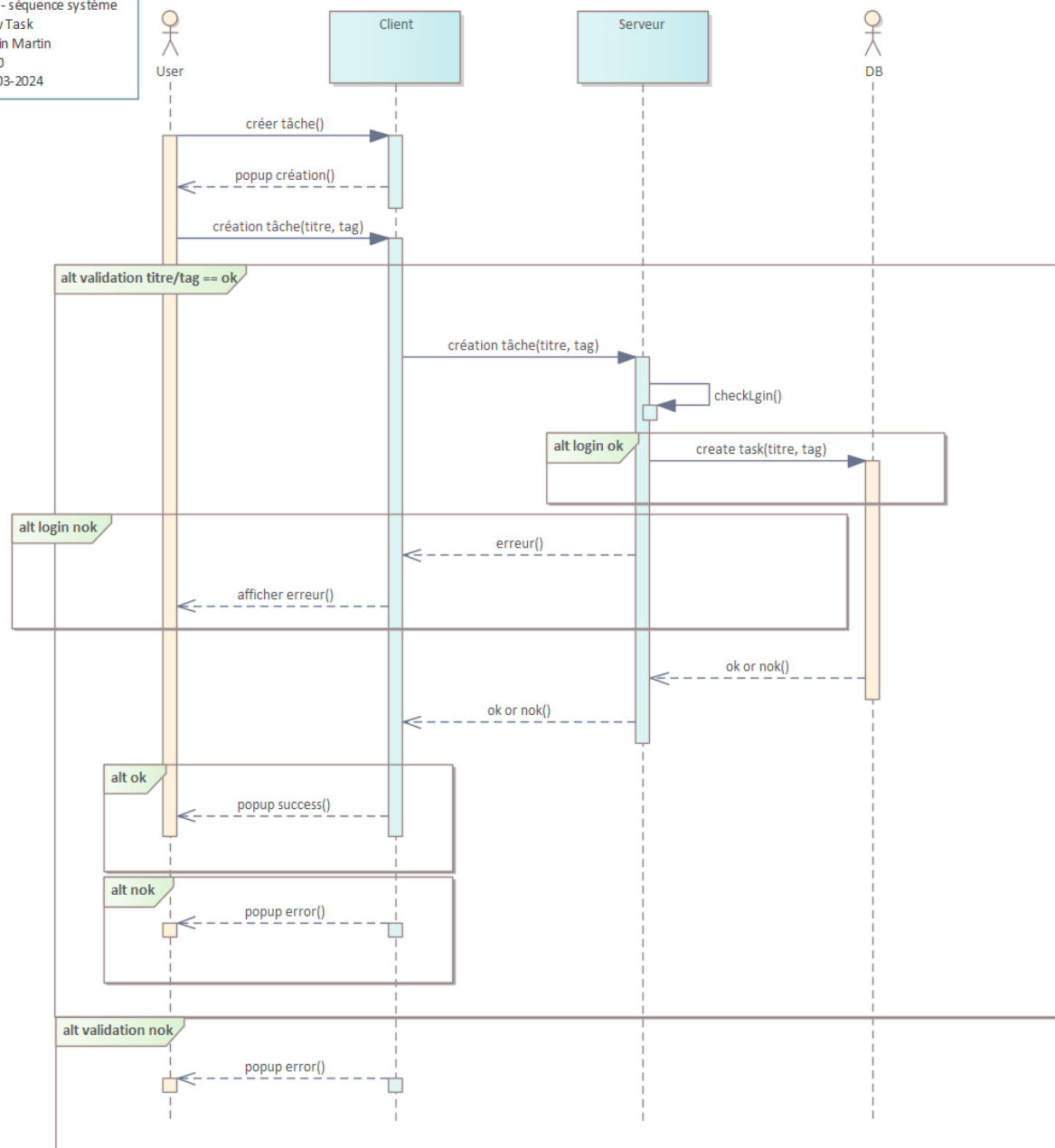
2.5.1 NewPost



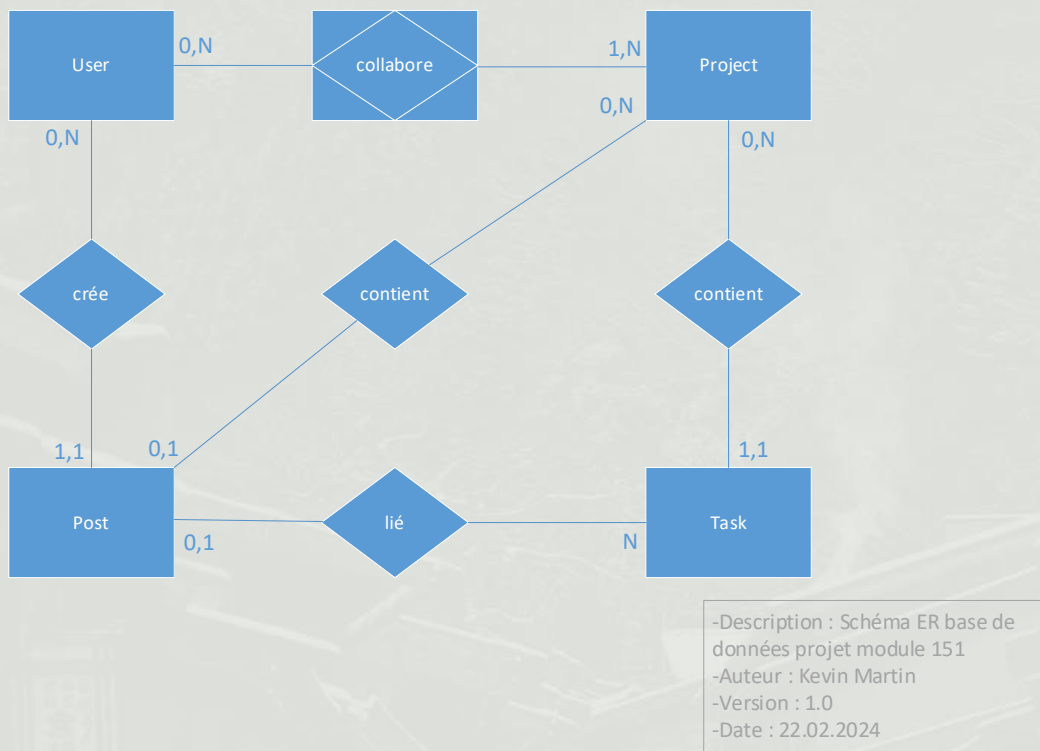
2.5.2 NewTask

sd SequenceNewTask

151 - séquence système
New Task
Kevin Martin
V2.0
02-03-2024



2.6 Schéma ER

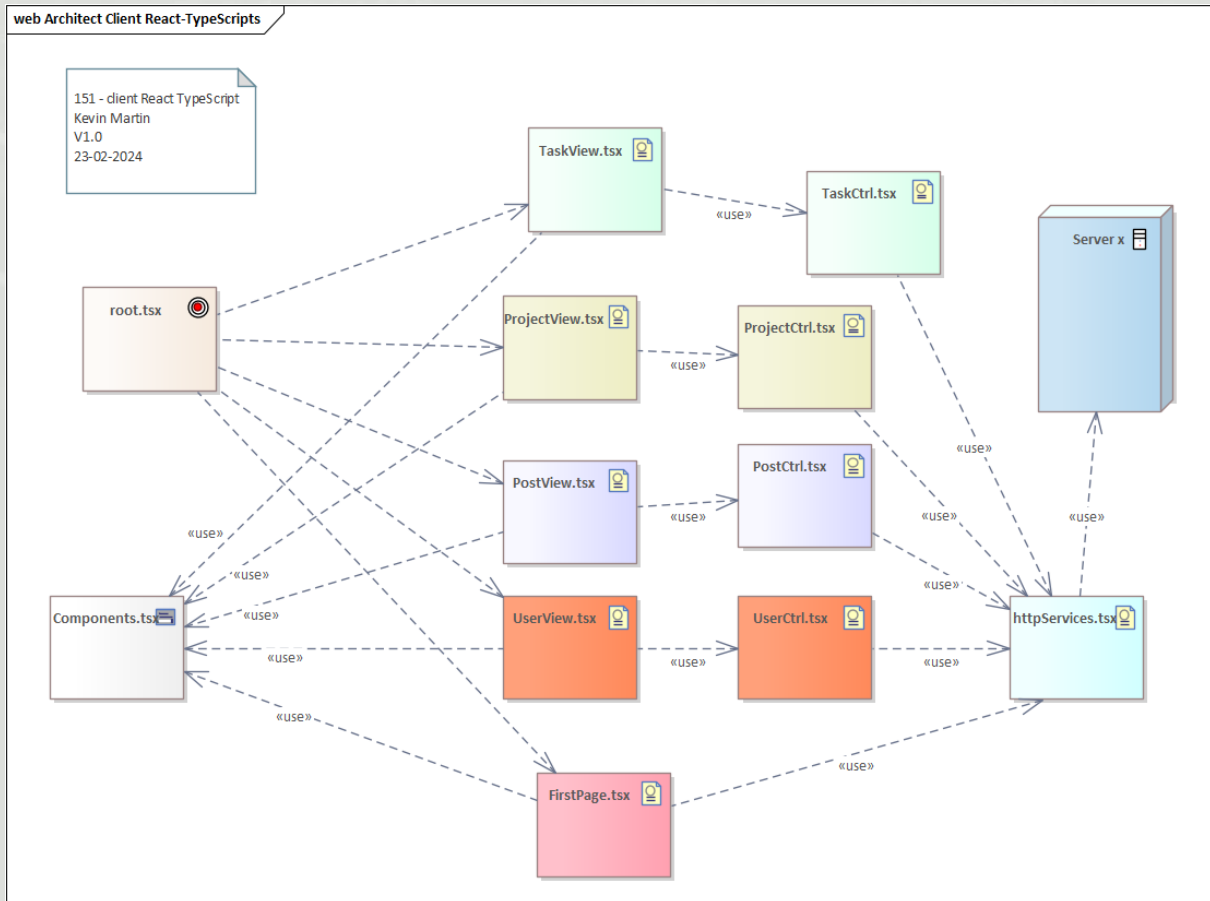


- ⇒ Un utilisateur peut créer plusieurs projets et un projet peut être lié à plusieurs utilisateurs. C'est pourquoi il va y avoir un champ permettant de savoir si l'utilisateur est le créateur ou juste un participant au projet.
- ⇒ Un projet peut contenir plusieurs tâches, mais une tâche est associée à un seul projet.
- ⇒ Un projet peut avoir plusieurs posts mais un post peut-être dans un seul projet.
- ⇒ Une tâche peut avoir plusieurs posts mais un post peut être lié à une seule tâche.

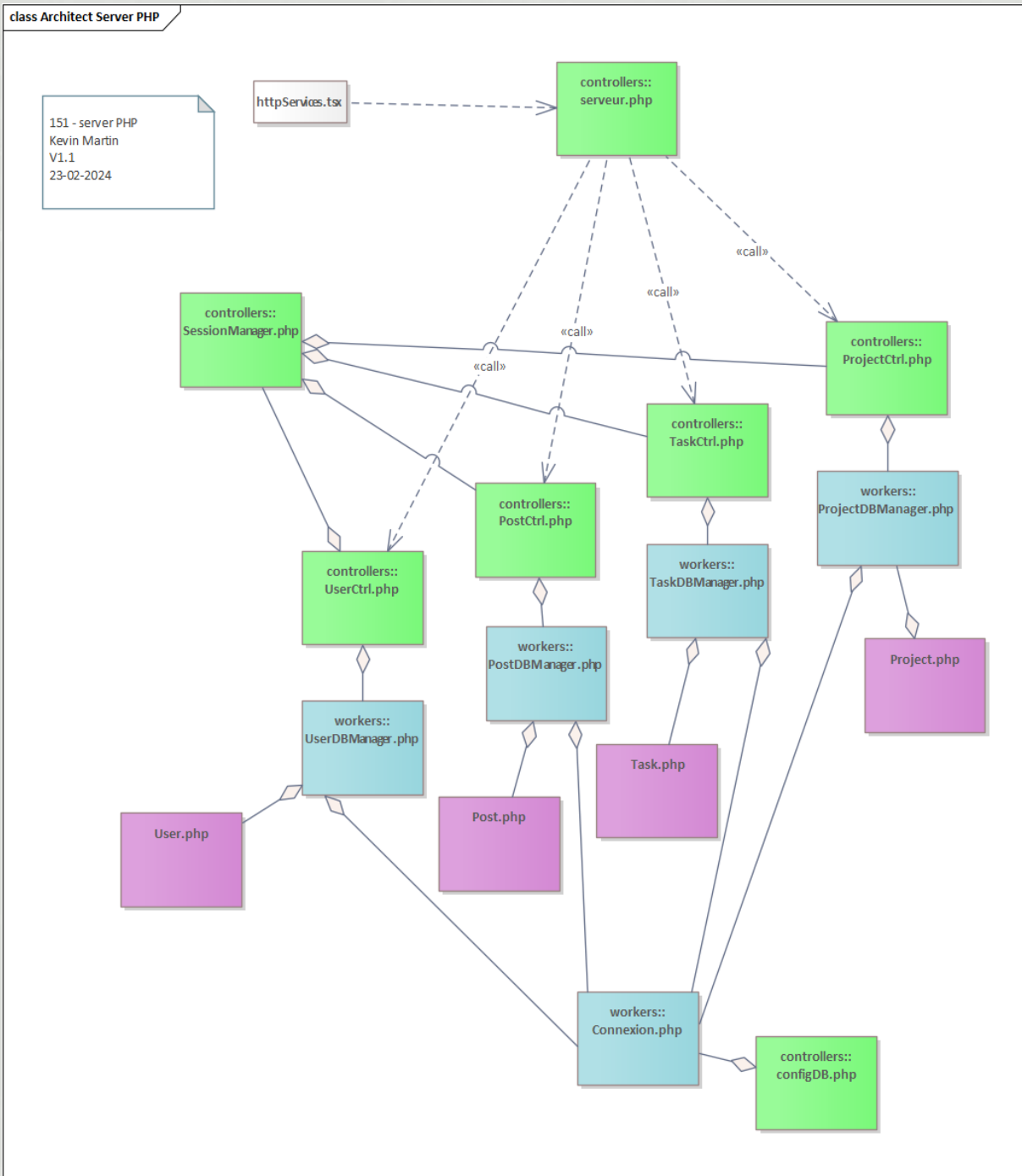
3 Conception

3.1 Diagramme de classe

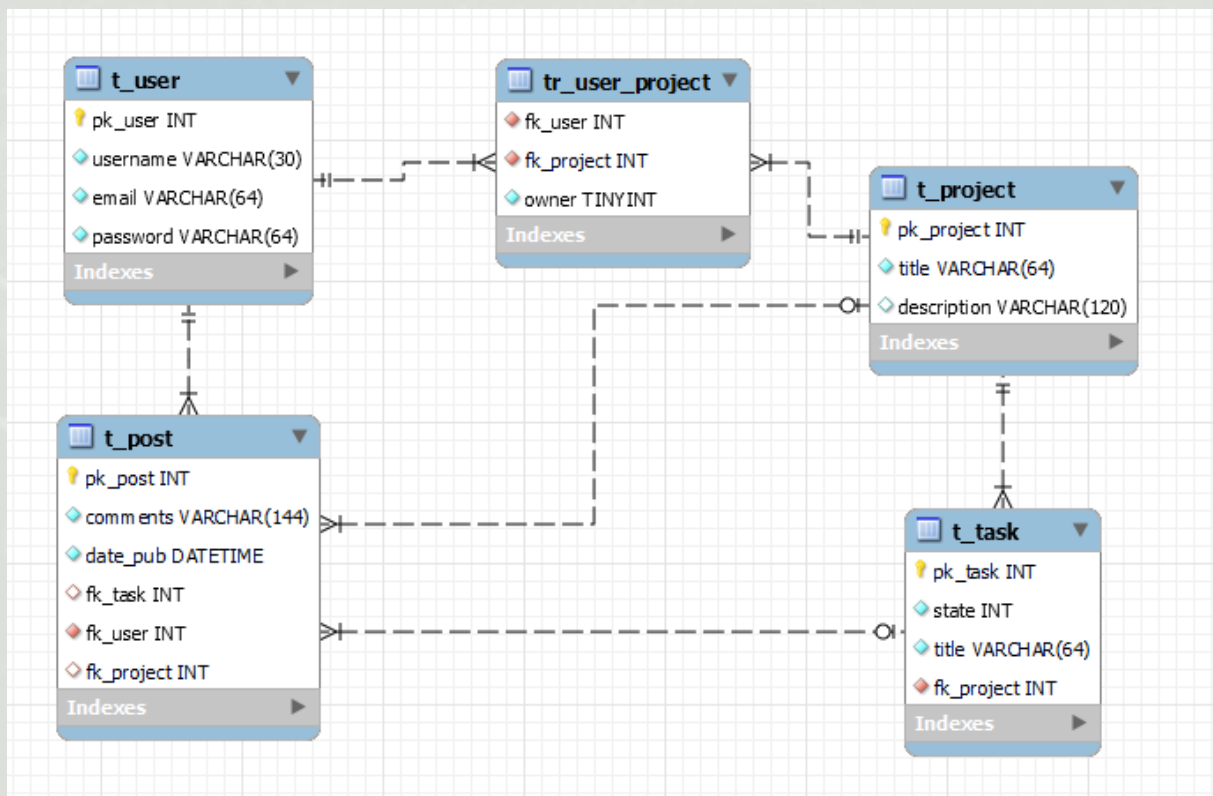
3.1.1 Client



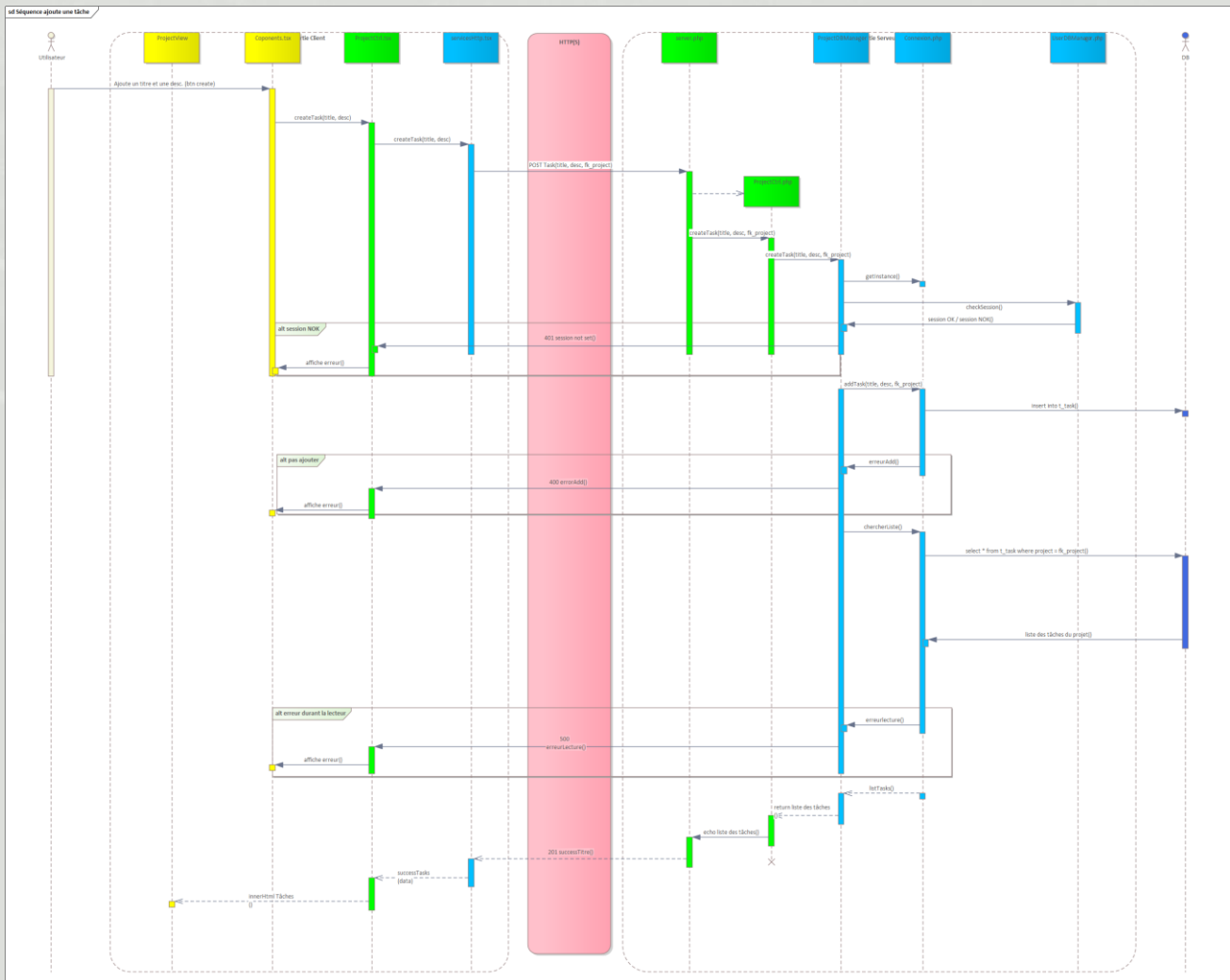
3.1.2 Serveur



3.2 Schéma relationnel



3.3 Diagramme séquence interactions



3.4 Conception des tests

3.4.1 Injections

Test	Description	Attendu	Obtenu
SignUp	Champs à remplir pour créer un compte	Aucune injections possible (client et serveur)	
SignIn	Champs à remplir pour se connecter à un compte	Aucune injections possible (client et serveur)	
CreateProject	Champs à remplir pour créer un projet	Aucune injections possible (client et serveur)	
CreateTask	Champs à remplir pour créer une tâche	Aucune injections possible (client et serveur)	

CreatePost	Champs à remplir pour créer un post	Aucune injections possible (client et serveur)	
-------------------	-------------------------------------	--	--

3.4.2 Postmann

Test	Description	Attendu	Obtenu
SignUp	Pouvoir créer un compte	Code success	
SignIn	Pouvoir se connecter	Code success	
CreatePost	Ne pas pouvoir créer de post (car non connecté)	Code error	
CreateTask	Ne pas pouvoir créer de tâche (car non connecté)	Code error	
CreateProject	Ne pas pouvoir créer de projet (car non connecté)	Code error	
GetListProjects	Pouvoir récupérer la liste des projets	Code success	
GetListTasks	Pouvoir récupérer la liste des tâches d'un projet	Code success	
GetListPosts	Pouvoir récupérer la liste des posts d'une tâche ou d'un projet	Code success	
GetProject	Pouvoir récupérer un projet	Code success	
GetListUsers	Ne pas pouvoir récupérer la liste des users (pas connecter en adm)	Code error	
GetUser	Ne pas pouvoir récupérer les infos du user (car pas connecté)	Code error	

3.4.3 Fonctionnalités

Test	Attendu	Obtenu
SignUp	Pouvoir créer un compte	
SignIn	Pouvoir se connecter	
CreateProject	Pouvoir créer un projet	
CreateTask	Pouvoir créer une tâche	
CreatePost	Pouvoir créer un post en relation avec une tâche ou le projet	
AddUserToProject	Pouvoir ajouter un utilisateur à notre projet pour collaborer	
DeleteProject	Pouvoir supprimer un projet	
DeleteTask	Pouvoir supprimer une tâche	

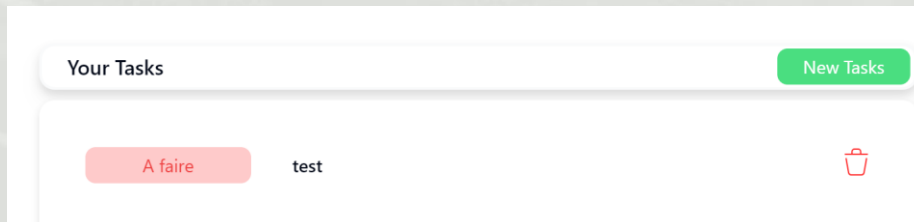
4 Implémentation

4.1 Descente de code

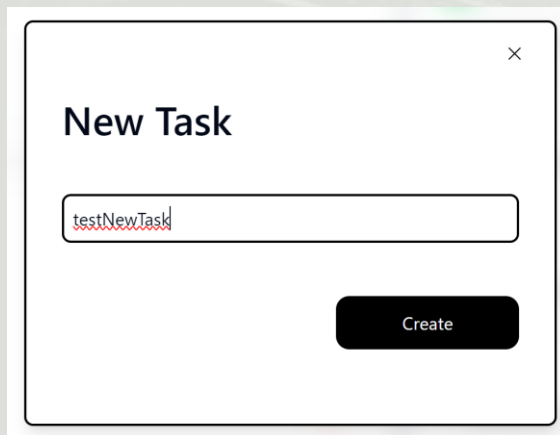
Descente de code pour l'ajout d'une tâche.

4.1.1 Interface

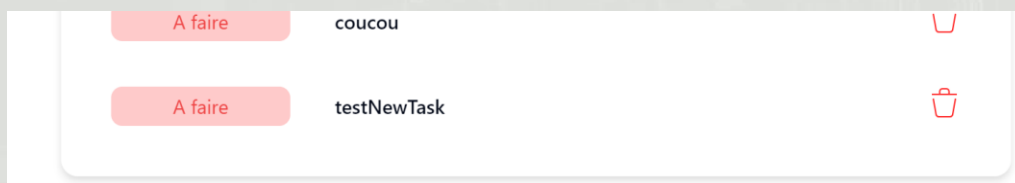
Pour créer une tâche, l'utilisateur doit appuyer sur le bouton vert « New Task ».



Ensuite, la popup s'affiche pour créer une nouvelle tâche, ou il faut ajouter un titre pour la tâche.



A la création d'une nouvelle tâche, le statut d'une tâche est « A faire » de base.



4.1.2 Client

4.1.2.1 Project.tsx

C'est ici que se trouve le code du bouton qui va ouvrir le popup.

```
<div className="w-full bg-white drop-shadow-lg rounded-xl px-5 py-1 border-gray-50 border-2 flex flex-row items-center">
  <p className="font-medium">Your Tasks</p>
  <button className="bg-green-400 py-1 px-5 text-white text-sm rounded-lg absolute right-1" onClick={togglePopupNewTask}>
    New Tasks
  </button>
</div>
```

```
const [showPopupNewTask, setshowPopupNewTask] = useState(false);

function togglePopupNewTask() {
  setshowPopupNewTask(!showPopupNewTask);
}
```

```
{showPopupNewTask && <NewTask onClose={() => setshowPopupNewTask(false)} />}
```

4.1.2.2 NewTask.tsx

C'est ce fichier qui est le popup (composant) qui est appelé lors du clic sur le bouton comme vu au-dessus. Tout d'abord on récupère l'id du projet se trouvant dans l'url et les infos du form de la popup (titre) afin d'appeler la méthode dans httpService qui va envoyer la requête fetch au serveur. Si la tâche a été créée, on reload la page afin d'actualiser la liste des tâches car cela se fait au chargement de la page.

```
import { motion } from "framer-motion";
import { useState } from "react";
import { createTask } from "@httpServices.ts";
import { useParams } from "react-router-dom";

const NewTask: React.FC<{ onClose: () => void }> = ({ onClose }) => {

  const { id } = useParams<{ id: string }>();

  // Définir les états pour les valeurs du formulaire
  const [title, setTitle] = useState<string>('');

  // Gérer la soumission du formulaire
  const handleSubmit = async (event: React.FormEvent) => {
    event.preventDefault();
    if (!title) {
      alert('Veuillez remplir un titre pour votre projet');
    } else {
      try {
        const taskCreate = await createTask(title, id);
        if (taskCreate) {
          alert('Votre tâche a été créée.');
```

```
          window.location.reload();
        } else {
          alert('Un problème est survenu lors de la création de la tâche. Veuillez réessayer.');
```

```
        } catch (error) {
          console.error('Erreur lors de la création de la tâche : ',
error);
          alert('Un problème est survenu lors de la création de la tâche
(try). Veuillez réessayer.');
```

```
        }
        // Réinitialiser les champs du formulaire après la soumission
        setTitle('');
      }
    }
  };
};
```



```

    return (
      <div className="fixed top-0 left-0 w-full h-full flex items-center
justify-center bg-white bg-opacity-30 backdrop-blur-md">
        <motion.div
          className="bg-white border-2 border-black h-2/4 w-2/6 rounded-lg
shadow-lg flex flex-col justify-center p-8 relative"
          initial={{ opacity: 0, y: -50 }}
          animate={{ opacity: 1, y: 0 }}
          exit={{ opacity: 0, y: -50 }}
        >
          <div className="absolute top-0 right-0 m-2">
            <button
              className="bg-white hover:bg-gray-100 text-white font-
bold py-2 px-4 rounded-lg"
              onClick={onClose}
            >
              <svg width="24px" height="24px" viewBox="0 0 24 24"
fill="none" xmlns="http://www.w3.org/2000/svg" color="#000000"><path d="M6.75827
17.2426L12.0009 12M17.2435 6.75736L12.0009 12M12.0009 12L6.75827 6.75736M12.0009
12L17.2435 17.2426" stroke="#000000"></path></svg>
            </button>
          </div>
          <div className="flex flex-col space-y-12">
            <h1 className="text-4xl font-medium">New Task</h1>
            <form onSubmit={handleSubmit} className="space-y-3">
              <div>
                <input
                  value={title}
                  onChange={(e) => setTitle(e.target.value)}
                  placeholder="Title"
                  className="border-2 border-gray-400 rounded-lg p-
2 w-full focus:outline-none focus:border-black"
                />
              </div>
            </form>
            <div className="flex justify-end">
              <motion.button
                className="bg-black px-12 py-3 w-2/5 text-white
rounded-xl"
                whileHover={{ scale: 1.1 }}
                transition={{ type: "spring", stiffness: 400,
damping: 10 }}
                onClick={handleSubmit}
              >
                Create
              </motion.button>
            </div>
          </div>
        </motion.div>
      </div>
    );
  };
export default NewTask;

```

4.1.2.3 httpServices.tsx

C'est ici que se trouve la méthode qui va envoyer la requête au serveur afin de créer et ajouter la nouvelle tâche dans la db.

```
export const createTask = async (title: string, fk_project: string | undefined) => {
  try {
    const response = await fetch(BASE_URL, {
      method: "POST",
      headers: {
        "ContentType": "application/json"
      },
      body: JSON.stringify({
        action: "newTask",
        title: title,
        fk_project: fk_project
      }),
      credentials: "include"
    });
    const data = await response.json();
    return !!data.newTaskOk;
  } catch (error) {
    console.error("Erreur lors de la création de la tâche : ", error);
    throw error;
  }
};
```

4.1.3 Serveur

4.1.3.1 Server.php

Lors de l'appelle, on vérifie si toutes les données sont présentes et on appelle le taskCtrl afin de l'ajouter à la db.

```
case 'newTask':
  if (isset($data['title'], $data['fk_project'])) {
    $title = $data['title'];
    $fk_project = $data['fk_project'];

    $result = $taskCtrl->newTask(1, $title, $fk_project);

    echo $result;
  } else {
    echo 'Certains champs sont manquants dans les données JSON !';
  }
  break;
```

4.1.3.2 TaskCtrl.php

Ce fichier permet de gérer la logique entre la communication avec la db et le retour au client.

```
class TaskCtrl{

  private $taskDBManager;
```

```
public function __construct() {
    $this->taskDBManager = new TaskDBManager;
}

public function newTask($state, $title, $fk_project){

    if($state != null && $title != null && $fk_project != ''){
        $task = new Task(null, $state, $title, $fk_project);
        $result = $this->taskDBManager->saveTask($task);
    }else{
        $result = 'Une erreur est survenue pour cause de manque de données.';
    }

    return json_encode($result);
}
```

4.1.3.3 TaskDBManager.php

C'est dans cette méthode que la requête sql afin d'ajouter la tâche à la db est faite.

```
class TaskDBManager{

    public function saveTask(Task $task){
        $connexion = Connexion::getInstance();
        $sql = "INSERT INTO t_task (state, title, fk_project) VALUES (:state, :title, :fk_project)";
        $params = array(
            ':state' => $task->getState(),
            ':title' => $task->getTitle(),
            ':fk_project' => $task->getFKProject()
        );
        try {
            $stmt = $connexion->executeQuery($sql, $params);
            if ($stmt != 0){
                return array("newTaskOk" => true);
            } else {
                return array("newTaskOk" => false);
            }
        } catch (PDOException $e) {
            return "Une erreur s'est produite lors de l'ajout d'une tâche : " . $e->getMessage();
        }
    }
}
```

4.1.3.4 Connexion.php

C'est se fichier qui nous permet de ce connecter à la db et de lancer la requête pdo adéquate.

```
public function executeQuery($query, $params) {
    try {
        $queryPrepared = $this->pdo->prepare($query);
        $queryRes = $queryPrepared->execute($params);
        return $queryRes;
    } catch (PDOException $e) {
        print "Erreur !: " . $e->getMessage() . "<br/>";
        die();
    }
}
```

4.1.3.5 Task.php

Ce fichier est le Modele afin de gérer les objet tâche.

```
<?php
class Task{

    private $pk;
    private $state;
    private $title;

    private $fk_project;

    public function __construct($pk, $state, $title, $fk_project){
        $this->pk = $pk;
        $this->state = $state;
        $this->title = $title;
        $this->fk_project = $fk_project;
    }

    public function getPk()
    {
        return $this->pk;
    }
    public function getState()
    {
        return $this->state;
    }
    public function getTitle()
    {
        return $this->title;
    }
    public function getFKProject()
    {
        return $this->fk_project;
    }

    public function toArray(){
        return array(
            'pk' => $this->getPk(),
            'state' => $this->getState(),
            'title' => $this->getTitle(),
            'fk_project' => $this->getFKProject()
        );
    }

    public function __toString(){
        return $this->title;
    }
}
?>
```

4.2 Problèmes rencontrés

4.2.1 Problème #1 base de données

J'ai eu des problèmes pour accéder à ma base de données depuis le php, j'avais soit un time out soit il ne reconnaissait pas le nom de la db. Pour résoudre ce problème, j'ai dû créer à la main la db depuis phpmyadmin et créer chaque table en sql directement.

4.2.2 Problème #2 cookies de session

J'ai rencontré un problème avec la session, au login le php me donnais bien la session en la stockant dans les cookies, mais à chaque requête vers le php, je recevais une nouvelle session uniquement pour cette requête ce qui fait que je ne pouvais pas utiliser les variables de session. Pour résoudre ce problème, j'ai dû ajouter dans chaque requête du client une option spéciale qui permet de stocker et d'utiliser les credential donc les cookies.

```
export const createTask = async (title: string, fk_project: string | undefined)
=> {
  try {
    const response = await fetch(BASE_URL, {
      method: "POST",
      headers: {
        "ContentType": "application/json"
      },
      body: JSON.stringify({
        action: "newTask",
        title: title,
        fk_project: fk_project
      }),
      credentials: "include"
    });
    const data = await response.json();
    return !!data.newTaskOk;
  } catch (error) {
    console.error("Erreur lors de la création de la tâche : ", error);
    throw error;
  }
};
```

4.3 Tests fonctionnels

4.3.1 Injections

Test	Description	Attendu	Obtenu
SignUp	Champs à remplir pour créer un compte	Aucune injections possible (client et serveur)	OK
SignIn	Champs à remplir pour se connecter à un compte	Aucune injections possible (client et serveur)	OK
CreateProject	Champs à remplir pour créer un projet	Aucune injections possible (client et serveur)	OK

CreateTask	Champs à remplir pour créer une tâche	Aucune injections possible (client et serveur)	OK
CreatePost	Champs à remplir pour créer un post	Aucune injections possible (client et serveur)	Post pas encore opérationnel

4.3.2 Postmann

Test	Description	Attendu	Obtenu
SignUp	Pouvoir créer un compte	Code success	Code success
SignIn	Pouvoir se connecter	Code success	Code success
CreatePost	Ne pas pouvoir créer de post (car non connecté)	Code error	Post pas opérationnel
CreateTask	Ne pas pouvoir créer de tâche (car non connecté)	Code error	Code error tant qu'on n'est pas loggé
CreateProject	Ne pas pouvoir créer de projet (car non connecté)	Code error	Code error tant qu'on n'est pas loggé
GetListProjects	Pouvoir récupérer la liste des projets	Code success	Code success
GetListTasks	Pouvoir récupérer la liste des tâches d'un projet	Code success	Code success
GetListPosts	Pouvoir récupérer la liste des posts d'une tâche ou d'un projet	Code success	Post pas opérationnel
GetProject	Pouvoir récupérer un projet	Code success	Code success
GetListUsers	Ne pas pouvoir récupérer la liste des users (pas connecter en adm)	Code error	Pas encore opérationnel
GetUser	Ne pas pouvoir récupérer les infos du user (car pas connecté)	Code error	Pas encore opérationnel

4.3.3 Fonctionnalités

Test	Attendu	Obtenu
SignUp	Pouvoir créer un compte	OK
SignIn	Pouvoir se connecter	OK
CreateProject	Pouvoir créer un projet	OK
CreateTask	Pouvoir créer une tâche	OK
CreatePost	Pouvoir créer un post en relation avec une tâche ou le projet	Post pas opérationnel
AddUserToProject	Pouvoir ajouter un utilisateur à notre projet pour collaborer	Pas opérationnel
DeleteProject	Pouvoir supprimer un projet	Pas opérationnel
DeleteTask	Pouvoir supprimer une tâche	Pas opérationnel

4.4 Hébergement

Le projet est hébergé sur le lien suivant : martink.emf-informatique.ch/151/Client

Cependant, j'ai un problème d'hébergement sur le cpanel avec react. Je vais donc essayer de le résoudre ou d'héberger mon site sur mon propre hébergeur.

5 Synthèse

5.1 Conclusion

Pour conclure j'ai beaucoup aimé ce module, même si nous avons eu peu de temps allouer au projet. J'ai appris beaucoup de chose en termes de gestion de projet ou de technologie notamment avec docker et le client en react.