

A thick vertical teal bar on the left side of the page. A teal arrow points to the right from this bar, containing the date.

03/06/2022

Améliorez une application existante de ToDo & Co

Projet P8 - OpenClassRooms

Several thin, curved, light blue lines that originate from the bottom left and sweep upwards and to the right, creating a sense of movement or a stylized wave.

Serge Pillay

Table des matières

Introduction	2
Contexte	2
Description du besoin	2
Corrections d'anomalies	2
Implémentation de nouvelles fonctionnalités	2
Documentation technique	2
Audit de qualité du code & performance de l'application	3
Les acteurs du projet	4
Les fonctionnalités attendues	4
Diagrammes de cas d'utilisation et de séquences	5
Diagramme d'utilisation	5
Diagramme de séquence : ajouter une tâche et lister les tâches.	6
Diagramme de classes	7
Modèle de données	7
Base de données MySQL	7



Introduction

Contexte

L'entreprise **ToDo & Co** est une startup dont le corps de métier est une application permettant de gérer ses tâches quotidiennes.

L'application vient a dû être développée à toute vitesse pour permettre de montrer à de potentiels investisseurs que le concept est viable (on parle de Minimum Viable Product MVP).

Le choix du développeur précédent a été d'utiliser le Framework PHP Symfony.

L'entreprise ToDo & Co ayant réussi à lever des fonds pour permettre le développement de l'entreprise et surtout de l'application, vous êtes embauché en tant que développeur expérimenté en charge des tâches suivantes :

- L'implémentation de nouvelles fonctionnalités.
- La correction de quelques anomalies.
- L'implémentation de tests automatisés.

Description du besoin

Corrections d'anomalies

- Une tâche doit être attachée à un utilisateur :
 - A la sauvegarde de la tâche, l'utilisateur authentifié soit rattaché à la tâche nouvellement créée
 - Lors de la modification de la tâche, l'auteur ne peut pas être modifié.
 - Pour les tâches déjà créées, il faut qu'elles soient rattachées à un utilisateur "anonyme".
- Choisir un rôle pour un utilisateur : Lors de la création d'un utilisateur, il doit être possible de choisir un rôle pour celui-ci. Les rôles listés sont les suivants :
 - Rôle utilisateur (ROLE_USER) ;
 - Rôle administrateur (ROLE_ADMIN).

Implémentation de nouvelles fonctionnalités

- Autorisation
 - Seuls les utilisateurs ayant le rôle administrateur (ROLE_ADMIN) doivent pouvoir accéder aux pages de gestion des utilisateurs.
 - Les tâches ne peuvent être supprimées que par les utilisateurs ayant créé les tâches en question.
 - Les tâches rattachées à l'utilisateur "anonyme" peuvent être supprimées uniquement par les utilisateurs ayant le rôle administrateur (ROLE_ADMIN).
- Implémentation de tests automatisés : Il vous est demandé d'implémenter les tests automatisés (tests unitaires et fonctionnels) nécessaires pour assurer que le fonctionnement de l'application est bien en adéquation avec les demandes.

Documentation technique

Il vous est demandé de produire une documentation expliquant comment l'implémentation de l'authentification a été faite. Cette documentation se destine aux prochains développeurs juniors qui rejoindront l'équipe dans quelques semaines.

Dans cette documentation, il doit être possible pour un débutant avec le framework Symfony de :

- Comprendre quel(s) fichier(s) il faut modifier et pourquoi ;
- Comment s'opère l'authentification ;
- Où sont stockés les utilisateurs.

Par ailleurs, vous ouvrez la marche en matière de collaboration à plusieurs sur ce projet. Il vous est également demandé de produire un document expliquant comment devront procéder tous les développeurs souhaitant apporter des modifications au projet.

Ce document devra aussi détailler le processus de qualité à utiliser ainsi que les règles à respecter.

Audit de qualité du code & performance de l'application

Les fondateurs souhaitent pérenniser le développement de l'application. Cela dit, ils souhaitent dans un premier temps faire un état des lieux de la dette technique de l'application.

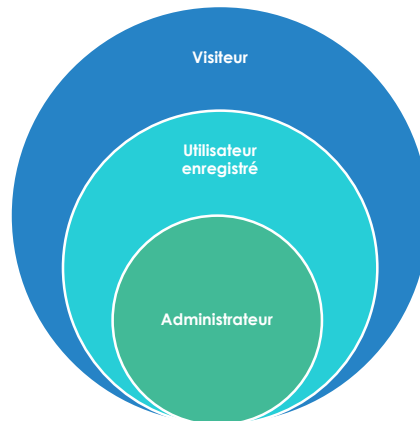
Au terme de votre travail effectué sur l'application, il vous est demandé de produire un audit de code sur les deux axes suivants : la qualité de code et la performance.

Les acteurs du projet

Les acteurs principaux :

- Les visiteurs
- Les utilisateurs enregistrés
- Les administrateurs

Remarque : dans ce projet, un acteur possède les fonctionnalités de son niveau plus les fonctionnalités des niveaux inférieurs. Schématiquement, nous avons :

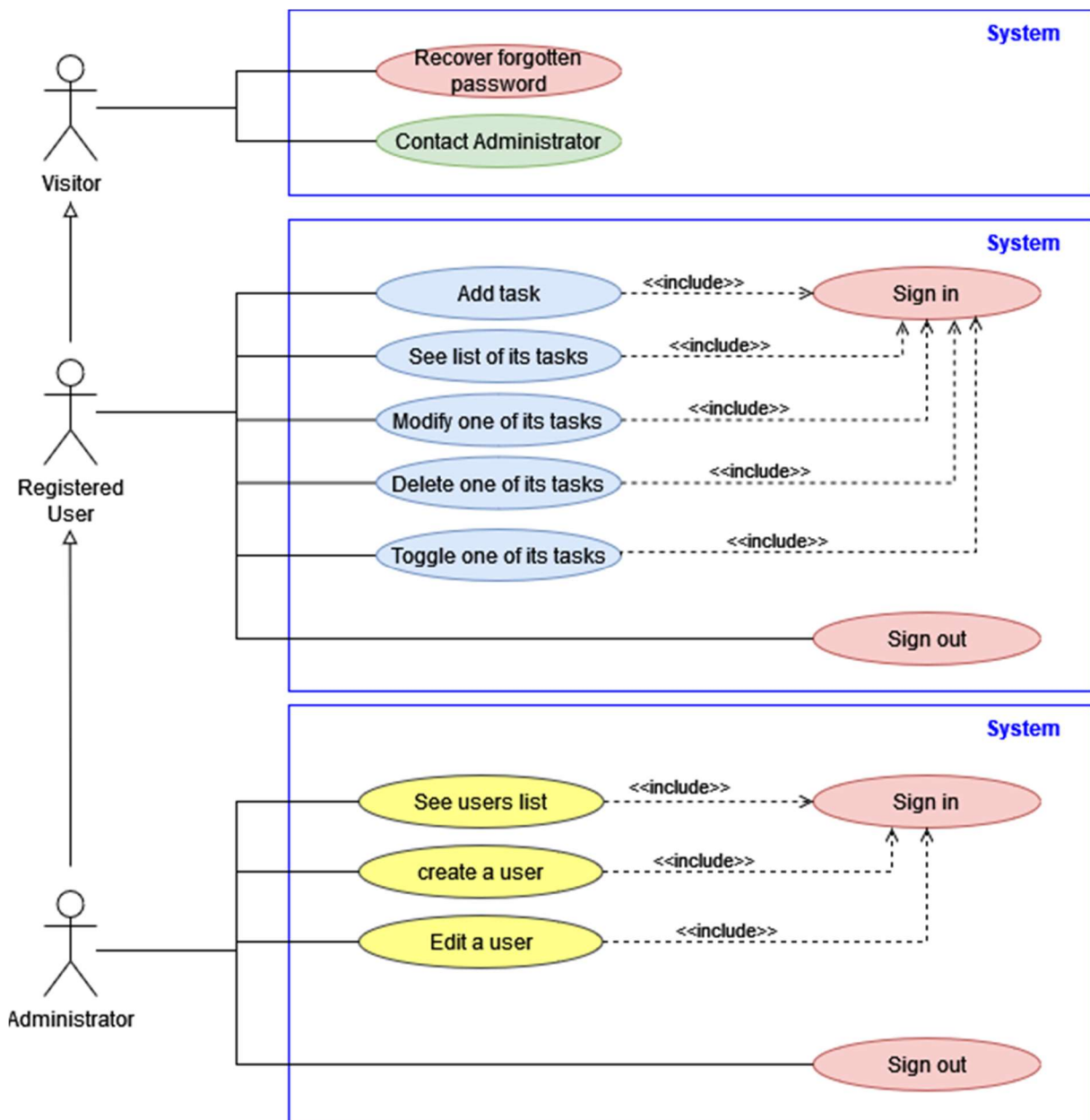


Les fonctionnalités attendues

Le visiteur doit pouvoir	<ul style="list-style-type: none">• Aucun accès
L'utilisateur enregistré doit pouvoir	<ul style="list-style-type: none">• Se connecter/déconnecter• (*) Créer/modifier une tâche• (*) Indiquer si une tâche est terminée ou non <p>(*) Si connecté</p>
L'administrateur doit pouvoir	<p>Comme l'utilisateur enregistré +</p> <ul style="list-style-type: none">• (*) Gérer les utilisateurs (dont les mots de passe) <p>(*) Si connecté</p>

Diagrammes de cas d'utilisation et de séquences

Diagramme d'utilisation



Les fonctionnalités du visiteur dans le schéma ci-dessus seront à implémenter dans une future version de l'application.

Diagramme de séquence : ajouter une tâche et lister les tâches.

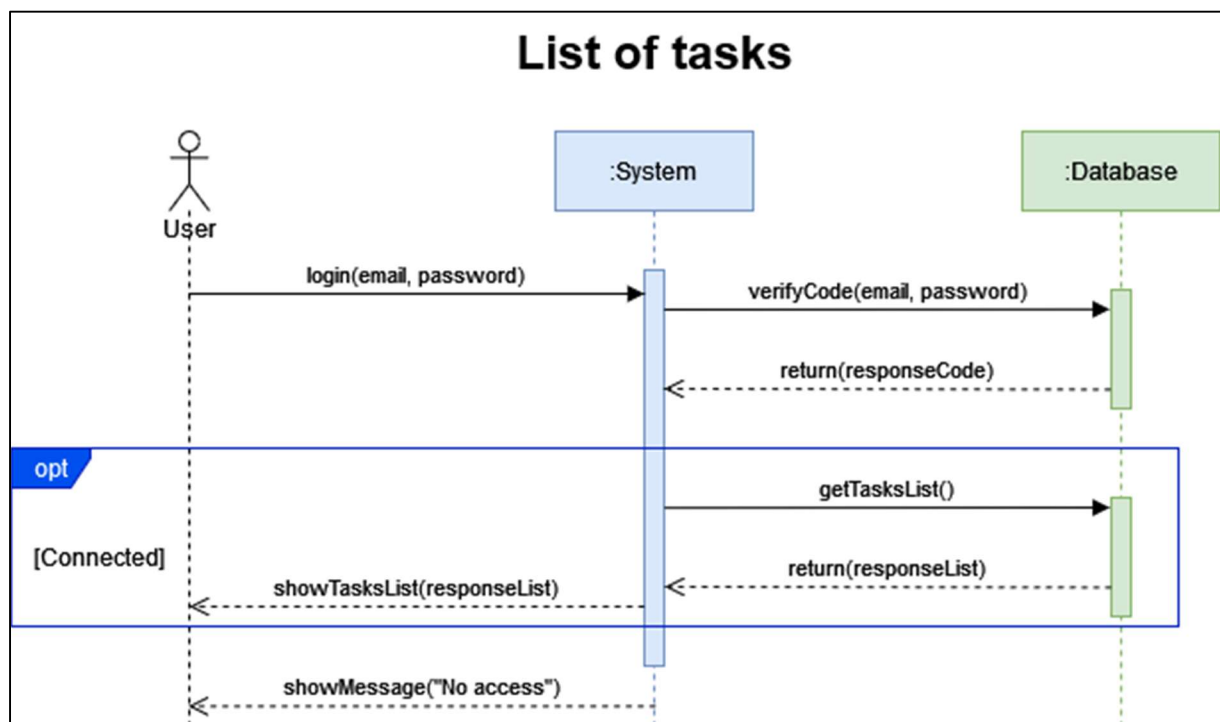
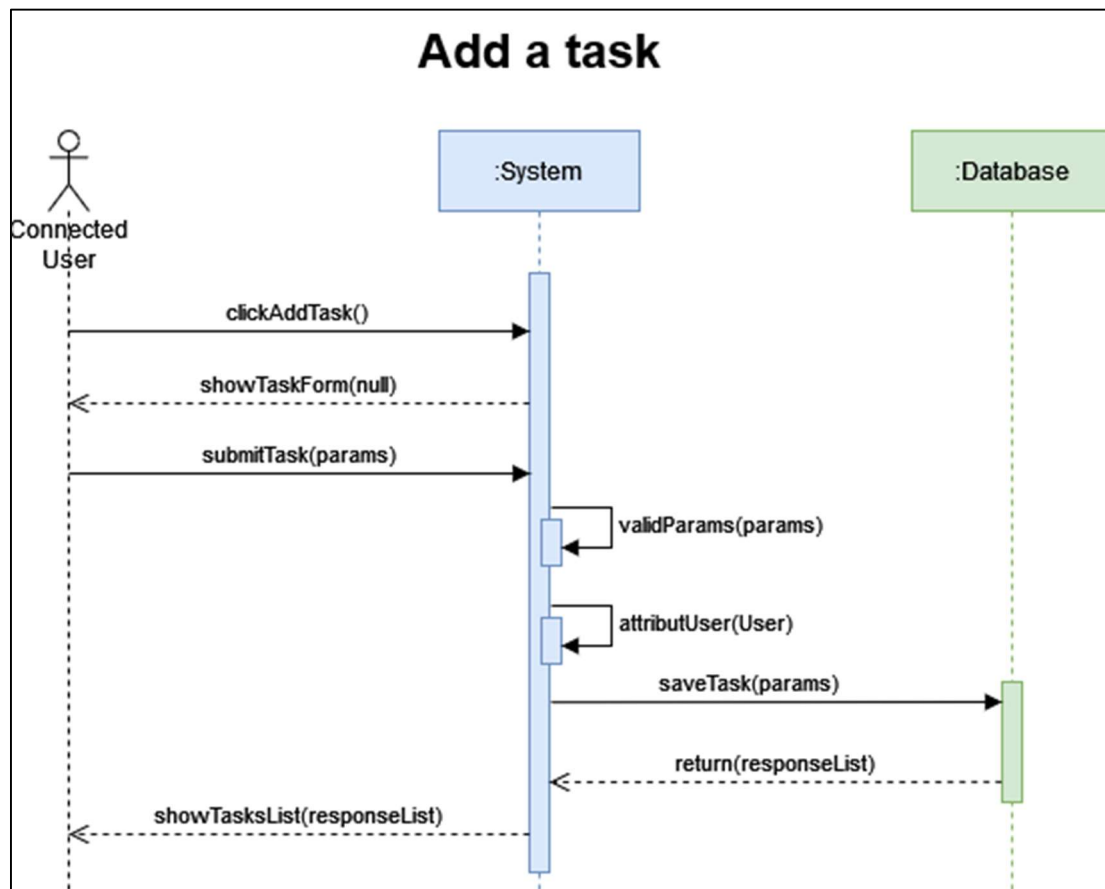


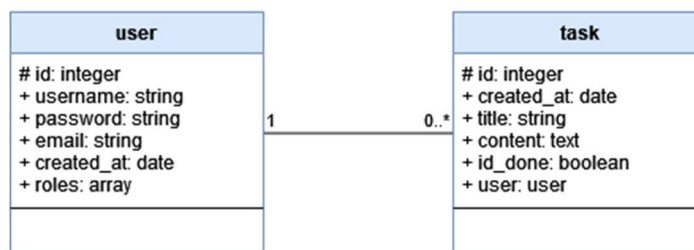
Diagramme de classes

Class diagram

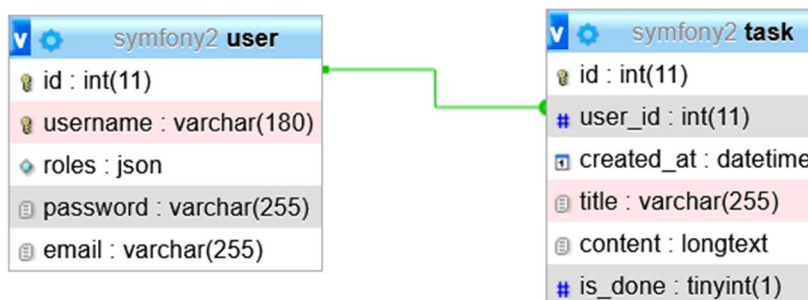
Original



Needed



Modèle de données



Base de données MySQL

Outils utilisés :

- La base de données fournie a été réalisée avec MySQL 5.7.31
- Doctrine 2 et le module supplémentaire doctrine/doctrine-fixtures-bundle en mode « dev » pour intégrer le jeu de données.