

TASK MANAGER

Table des matières

Énoncé	2
Synthèse	2
Développement de solution	3
Diagramme des interacteurs	3
Décomposition des fonctions	3
FP1 : gérer la tâche	3
FC1 : vérifier la tâche	3
FC2 : informer l'utilisateur en temps réel	4
Résumé des fonctions	4
Solutions	4
CRUD	4
Entité	4
Mercure	5

Énoncé

Un client aimerait un premier jet d'une page web permettant de gérer des tâches.

Le dictionnaire de données comprend :

- Une date obligatoire, qui doit s'afficher au format français sur le site ;
- Un nom, qui doit être rempli ;
- Une description, qui peut être vide et contenir des textes jusqu'à 500 caractères.

Les tâches sont affichées dans un tableau, on peut les modifier et les supprimer.

Le client souhaite que tout se passe sur la même page et voir les changements en temps réel. Si le client va sur le site et qu'une ou des tâches ont lieu aujourd'hui, cela doit lui être notifié dans la zone en haut de la page.

Le design n'est pas important pour cette version, tu es libre de ne développer que les fonctionnalités ou d'implémenter une librairie de ton choix. Des instructions doivent être également indiquées pour pouvoir lancer cette page web sur le serveur du client ou chez PERIG.

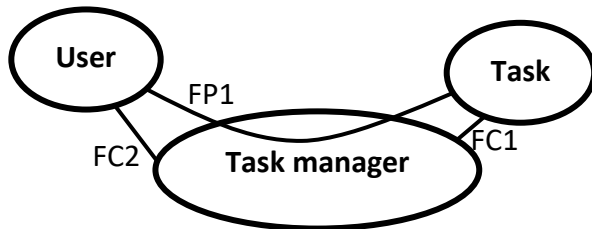
Le projet doit être fait majoritairement en PHP et/ou Javascript, le rendu géré en HTML et CSS. Le plus important est un code fonctionnel, la qualité du code, du commentaire et le respect des bonnes pratiques.

Synthèse

Créer un outil de gestion de tâches (CRUD) dont les changements seront visibles en temps réel pour l'ensemble des personnes présente sur la page.

Développement de solution

Diagramme des interacteurs



Fonction	Définition
FP1	Gérer la tâche
FC1	Vérifier la tâche
FC2	Informer l'utilisateur en temps réel

Décomposition des fonctions

FP1 : gérer la tâche

La fonction principale 1 peut être décomposée en plusieurs fonctions de services.

Fonction	Définition
FS1	Créer la tâche
FS2	Modifier la tâche
FS3	Supprimer la tâche
FS4	Lire la tâche

FC1 : vérifier la tâche

La fonction de contrainte 1 peut être décomposée en fonctions en se basant sur la définition de l'entité donnée dans le briefing.

Fonction	Définition
FS5	Vérifier la présence d'une date de rendu pour la tâche
FS6	Vérifier la présence d'un nom de tâche
FS7	Permettre une description de tâche vide
FS8	Vérifier que la description de tâche ne dépasse pas les 500 caractères

FC2 : informer l'utilisateur en temps réel

La fonction de contrainte 2 peut être subdivisée en trois fonctions distinctes.

Fonction	Définition
FS9	Notifier en temps réel l'ensemble des utilisateurs de l'action en cas de succès
FS10	Notifier l'utilisateur si une des tâches possède une date correspondant au jour en cours

Résumé des fonctions

Fonction	Définition
FS1	Créer la tâche
FS2	Modifier la tâche
FS3	Supprimer la tâche
FS4	Lire la tâche
FS5	Vérifier la présence d'une date de rendu pour la tâche
FS6	Vérifier la présence d'un nom de tâche
FS7	Permettre une description de tâche vide
FS8	Vérifier que la description de tâche ne dépasse pas les 500 caractères
FS9	Notifier en temps réel l'ensemble des utilisateurs de l'action en cas de succès
FS10	Notifier l'utilisateur si une des tâches possède une date correspondant au jour en cours

Solutions

CRUD

Les fonctions de service 1 2 3 et 4 peuvent être gérées par un controller (TaskController) admettant plusieurs routes pour définir un système CRUD (Create, Read, Update, Delete). Elles peuvent être créées sous forme de simple route d'affichage ou par des routes API dont les données renvoyées seront gérées en front par JavaScript.

Certaines vérifications seront à prendre en compte (exemple : « La tâche existe-t-elle ? » pour les fonctions Update et Delete). Si cette vérification échoue le système devra renvoyer une réponse d'erreur exploitable en front. Si en revanche, l'action s'est déroulée sans problème, il faudra alors renvoyer une réponse de succès exploitable en front.

Entité

Les fonctions de service 5 6 7 et 8 doivent être directement gérées par l'entité Task. Cela sera fait à l'aide de constraints mais également dans le constructeur de l'entité. Si certaines informations envoyées sont manquantes ou ne respectent pas la définition de la Task, le système devra renvoyer une réponse exploitable en front.

Mercure

La fonction de service 9 permettant d'informer en temps réel d'un changement sur les tâches devra être gérée par un système permettant du SSE (Server-Sent Events). Ces événements seront ensuite gérés en front par du JavaScript. Symfony propose l'utilisation de Mercure pour effectuer cette tâche. Une communication en websocket pourrait également être utilisée, mais limiterait l'acquisition de connaissances.

Liens :

- <https://symfony.com/blog/symfony-gets-real-time-push-capabilities>
- <https://symfony.com/doc/current/mercure.html>