



Billed

Mise en place de la fonctionnalité “notes de frais”

Description pratique des besoins

Contexte

- Départ imprévu de Garance qui travaillait initialement sur le projet et remplacement au pied levé.
- Fonctionnalité très attendue et timing serré car lancement dans deux semaines.
- Deux parcours sur la fonctionnalité : 1. Employé et 2. Administrateur RH.
- Où en sommes-nous actuellement ?

	Back-end	Front-end
Parcours employé	✓ Prêt en version alpha	✗ À tester ✗ À débbugger
Parcours admin	✓ Prêt en version alpha	✓ Testé ✗ À débbugger

Équipe et missions

Matthieu

Lead developer
Supervise le projet

Garance

Feature team
A développé, débbuggé
et testé une partie de la
fonctionnalité
A quitté l'entreprise

Remplaçant(e) de Garance

Front-end developer
Débbugge et teste la
fonctionnalité

Leïla

Quality assurance
Identifie les bugs et
réalise les tests
End-to-End

Tâche

Description de la tâche

Règles/contraintes

1. [Bug - report]

Fixer les bugs identifiés dans le rapport de bug fourni par Jest.
Une copie est disponible dans le [kanban Notion](#).

Utiliser **Chrome Debugger**.

2. [Bug - hunt]

Fixer les bugs identifiés par Leila sur le parcours employé.
Ils sont décrits dans le [kanban Notion](#).

Utiliser **Chrome Debugger**.

3. [Tests unitaires]

Ajouter des tests unitaires et

Il faut assurer un **taux de**

et d'intégration]	<p>d'intégration pour les fichiers Bills et NewBill.</p> <p>Ils vont permettre d'éliminer les bugs et d'éviter toute régression lors des prochaines évolutions de la solution.</p> <p>Certains tests sont déjà développés (pour le Login et pour le Dashboard côté administrateur RH) : ils sont déjà cochés sur le kanban. Il faut s'en inspirer pour les restants.</p>	<p>couverture global des containers de 80% minimum en testant les appels réseaux.</p> <p>Conseils :</p> <ul style="list-style-type: none"> ⇒ S'appuyer sur le rapport de couverture de Jest Lancer l'application avec live-server pour pouvoir le lire et aller à l'adresse http://127.0.0.1:8080/coverage/lcov-report/ (tout est indiqué dans le readme). ⇒ S'appuyer sur les fichiers Login.js et Dashboard.js pour réaliser un mock de l'API. ⇒ S'assurer d'utiliser des matchers pertinents, afin de bien tester l'application et pas simplement obtenir une bonne couverture. ⇒ Ne pas oublier de tester les erreurs 404 et 500.
4. [Test End-to-End]	<p>Rédiger un plan de test End-to-End (E2E) sur le parcours employé pour guider Leïla.</p>	<p>Manque de temps pour automatiser les tests (E2E). Ils seront effectués manuellement par Leïla.</p> <p>S'inspirer du plan E2E que Garance a déjà rédigé sur le parcours administrateur RH.</p>
Autres informations		

- L'application contient déjà des données test mais il est nécessaire d'en créer de nouvelles.
- Des comptes administrateur et employé ont été créés pour les tests dans [le readme du code front-end](#). Il faut les utiliser pour pouvoir charger une note de frais côté employé et la consulter côté administrateur RH.