



Sorbonne Université
Faculté de Science et d'ingénierie
Département Informatique

Rapport Informatique

*Spécialité :
Science et Technologie Logiciel*

GPSTL

Rapport des séances

Tabellout Yanis

le : 26/02/2025

CHAPITRE 1

RAPPORT

1. Gestion d'arrivée de Fréddie

1.1. email

De : Alice

À : Bob

Objet : Proposition d'organisation TDD à l'arrivée de Fréddie - Ticket 1664

Date : 05 mars 2025

Cher Bob,

Suite à notre dernière réunion de synchronisation et à l'arrivée de Freddie en tant qu'apprenti QA, voici ma proposition de plan d'action orienté TDD pour le ticket 1664. Ce plan vise à structurer nos tests et à assurer une couverture optimale en intégrant les principes du développement piloté par les tests (TDD).

- **Plan de développement et UI/UX** : Le design a été complété, le projet consistera principalement de deux écrans : un écran d'accueil ainsi qu'un écran pour la recherche. J'ai bien reçu ton dernier e-mail me confirmant la bonne initialisation des modules concernés discuté précédemment. □
- **Plan d'intégration de Fréddie** : Freddie aura une semaine de montée en compétence en QA puis participera activement à la définition et à l'exécution des tests en collaboration avec l'équipe. Il sera encadré principalement par notre QA David.

Pour l'instant David et moi avons identifié certains tests critiques que je vais t'envoyer en Pi-èce jointe, et j'ai mis à jours le Diagramme de Gantt pour prévoir une approche TDD Cordialement,

Alice

1.1.1. Micro du prévisionnel réel du premier sprint

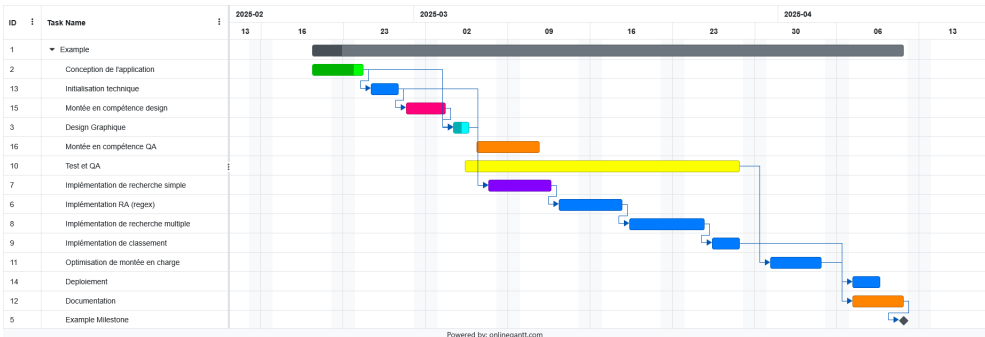


FIGURE 1.1 – Diagramme de Gantt Micro du premier sprint

1.2. Batterie de tests

ID	Objectif	Instances de test	Assertions
T1	Vérifier que la recherche par mot-clé exact retourne les résultats attendus.	Requête avec le mot-clé "salmon".	<code>assert_true("salmon" in returned_urls)</code>
T2	Vérifier la prise en charge des expressions régulières dans la recherche avancée.	Requête avec l'expression "s.*n".	<code>assert_true(any(re.match("s.*n", tag) for tag in returned_tags))</code>
T3	Vérifier que les résultats sont classés en fonction d'un critère de pertinence.	Recherche avec le mot "fish" et analyse de l'ordre des résultats.	<code>assert_true(returned_urls[0] is most_relevant)</code>
T4	Vérifier que la recherche multiple retourne des résultats distincts par mot-clé.	Requête avec "shrimp%1664tuna".	<code>assert_true(len(returned_urls) == 2 and "shrimp" in returned_urls[0] and "tuna" in returned_urls[1])</code>
T5	Vérifier la réponse du moteur de recherche à une requête vide.	Requête avec une chaîne vide "".	<code>assert_true(returned_urls == [])</code>
T6	Vérifier que les caractères spéciaux sont correctement interprétés dans la recherche.	Requête avec "sea-bream".	<code>assert_true("sea_bream" in returned_urls)</code>
T7	Vérifier la réaction du moteur à un mot-clé absent de la base.	Requête avec "dragonfish".	<code>assert_true(returned_urls == [])</code>
T8	Tester la capacité du moteur à répondre rapidement sous une charge importante.	Exécuter 100 000 requêtes en une minute.	<code>assert_true(avg_response_time < threshold)</code>
T9	Vérifier que la liste des résultats ne contient pas de doublons (TDD).	Requête avec "oyster".	<code>assert_true(len(set(returned_urls)) == len(returned_urls))</code>
T10	Tester la gestion des erreurs lorsqu'une requête est mal formée.	Requête avec "salmon OR tuna".	<code>assert_true(error_raised)</code>

TABLE 1.1 – Batterie de tests unitaires pour le moteur de recherche d'images

BIBLIOGRAPHIE