

Cahier de recette

*Prototype d'agent
conversationnel pour les
séniors*

Cahier de recette

Prototype d'agent conversationnel pour les séniors

Les informations d'identification du document

Référence du document :	D6
Version du document :	1.01
Date du document :	13 janvier 2025
Auteur(s) :	Prénom Nom

Les éléments de vérification du document

Validé par :	Nom de l'encadrant
Validé le :	13/1/25
Soumis le :	13/1/25
Type de diffusion :	Document électronique (.odt)
Confidentialité :	

Mots clés : modèle de cahier de recette, exemple à adapter, ...

1.Sommaire

Sommaire	3
1. Soumission	6
1.1. Remise de l’application.....	6
1.2. Remise des documents liés au projet	6
2.Vérification.....	6
2.1. Environnement de test.....	6
2.2. Ensemble des tests à réaliser	6
3. Validation.....	7
3.1. Décisionnaire.....	7

2.Soumission

2.1. Remise de l'application

L'application sera remise aux utilisateurs, sous la forme d'un fichier compressé téléchargeable.

2.2. Remise des documents liés à l'application

Les documents en lien avec l'application seront disponibles sur le dépôt git dans le dossier « CR » pour les comptes-rendus de réunions déposés en markdown et dans le dossier « gestion_de_projet » pour le cahier des charges, de recettes et le plan de développement déposés en pdf.

3.Vérification

3.1. Environnement de test

Afin de permettre un développement propre de l'application, il sera essentiel de mettre en place des tests de non régression, permettant, en cas par exemple de revue de code, que les fonctionnalités ne soient pas impactées. Ces tests seront divisés en deux grands types de. Soit, les tests unitaires, qui auront pour rôle de vérifier le bon fonctionnement de portion de code, par le biais d'automatismes. Ainsi que les tests d'intégration, ces derniers auront pour rôle de détecter les erreurs qui n'auront pas été trouvés par les tests unitaires, qui seront pour leur part des tests manuels effectués par les développeurs.

Pour les tests unitaires, ils seront gérés par l'outil Angular pour la partie front end et JUnit pour le back end de l'application. Et les tests devront être implémentés au fur et à mesure du développement de l'application, l'objectif étant d'ajouter les tests en lien avec les nouvelles portions de code.

Pour les tests d'intégration, ils seront stockés dans un fichier Excel, organisé avec une fonctionnalité à tester par page,

FonctionDeRappelPlanning

FonctionPropositionDActivités

Figure 1 : Exemple, page de fonctionnalités

Pour chaque fonctionnalité, il sera demandé au testeur de suivre un scénario et de vérifier que les résultats attendus aux actions effectués sont les bons.

1			
2	Fonction de rappel de planning		
3			
4			
5	Actions	Résultas attendus	
6	Action1		
7	Action2		
8	Action3	Résultat attendus 1	
9			
10			

Figure 2 : Exemple pour une fonctionnalité

3.2. Ensemble des tests à réaliser

Test unitaires :

- Couvrir le plus possible de portions de code, soit quand le programme répond à la demande, mais également les cas d'exceptions

Tests d'intégration :

- Vérification par un humain que l'intégralité des éléments de l'interface sont fonctionnelles et correctement situés
- Vérifier que l'agent répond aux attentes

4. Validation

4.1. Décisionnaire

Couvrir le code par les tests unitaires à prêt de 80% et procéder aux tests à chaque fin de développement de fonctionnalité, avant dépôt sur Git. Et prévoir une liste de tests d'intégration passant par l'intégralité des fonctions du système.