

Plan de test

Nom du projet

PRÉNOM NOM

Plan de test

Nom du projet

Les informations d'identification du document

Référence du document :	D7
Version du document :	1.01
Date du document :	5 juin 2025
Auteur(s) :	Marc Dieudonné

Les éléments de vérification du document

Validé par :	Nom de l'encadrant
Validé le :	05/06/2025
Soumis le :	05/06/2025
Type de diffusion :	Document électronique (.odt)
Confidentialité :	

Mots clés : modèle de plan de test, exemple à adapter, ...

Sommaire

Sommaire.....	3
1. Introduction (ou préambule).....	5
1.1. Objectifs et méthodes.....	5
1.2. Documents de référence.....	5
2. Guide de lecture.....	5
2.1. Maîtrise d'œuvre.....	5
2.2. Maîtrise d'ouvrage.....	5
3. Concepts de base.....	5
4. Tests fonctionnels.....	5
4.1. Pour chaque scenario :.....	5
4.1.1. Identification.....	5
4.1.2. Description.....	6
4.1.3. Contraintes.....	6
4.1.4. Dépendances.....	6
4.1.5. Procédure de test.....	6
5. Tests d'intégration.....	6
5.1. Pour chaque test d'intégration :.....	6
5.1.1. Identification.....	6
5.1.2. Description.....	6
5.1.3. Contraintes.....	6
5.1.4. Dépendances.....	7
5.1.5. Procédure de test.....	7

6. Tests unitaires.....	7
6.1. Pour chaque test unitaire :.....	7
6.1.1. Identification.....	7
6.1.2. Description.....	7
6.1.3. Contraintes.....	7
6.1.4. Dépendances.....	7
6.1.5. Procédure de test.....	8
7. Vérification de la documentation.....	8
8. Annexes.....	8
9. Glossaire.....	8
10. Références.....	8
11. Index.....	8

1.Introduction (ou préambule)

Précise l'objectif du document et en résume le contenu

1.1. Objectifs et méthodes

Ce document décrit les tests à effectuer sur le prototype du chatbot pour les seniors. Il couvre :

- les tests fonctionnels
- les tests d'intégration
- les tests unitaires
- la vérification de la documentation

Les méthodes utilisées incluent des tests manuels sur navigateur, des ateliers de validation utilisateurs, et des tests automatisés via console développeur, localStorage debugger, etc.

1.2. Documents de référence

Cahier des charges – Agent conversationnel seniors

- Cahier de recette
- Maquettes Figma
- Documentation API WeatherAPI
- MDN Web Docs (localStorage)

2.Guide de lecture

Précise, pour chaque type de lecteur, comment utiliser efficacement le document

2.1. Maîtrise d'œuvre

Destiné aux développeurs, testeurs et designers : suivre les cas de test et vérifier la conformité.

2.2. Maîtrise d'ouvrage

Destiné au commanditaire et aux utilisateurs finaux pour valider que l'application réponde aux besoins.

3. Concepts de base

- **Chatbot** : Assistant numérique spécialisé dans les risques domestiques.
- **Accessibilité** : Réglages personnalisés de l'interface (police, couleurs, bulle).
- **localStorage** : Système de stockage local des données utilisateur.
- **Profil utilisateur** : Données personnelles saisies via formulaire.
- **API WeatherAPI** : Service de météo géolocalisée intégré.

4. Tests fonctionnels

Décrire les scénarii (ainsi que leur enchainement) permettant de vérifier que l'application recouvrira bien le périmètre fonctionnel qui a été défini lors de la phase de spécification. Le périmètre fonctionnel est défini dans le cahier des charges (expression fonctionnelle du besoin)

4.1. Navigation entre vues:

4.1.1. SCN-001

4.1.2. Description

Vérifier l'accès correct aux interfaces via le menu.

4.1.3. Contraintes

JS activé

4.1.4. Dépendances

Aucune

4.1.5. Procédure de test

Cliquer sur chaque lien et vérifier que la redirection fonctionne et que la page soit chargée sans erreur

4.2. Profil utilisateur

4.2.1. SCN-002

4.2.2. Description

Tester formulaire de profil et affichage.

4.2.3. Contraintes

localStorage activé

4.2.4. Dépendances

Aucune

4.2.5. Procédure de test

Remplir le formulaire puis le afficher profil vérifier que les données persistes et sont affichées correctement

4.3. Chatbot conversationnel

4.3.1. SCN-003

4.3.2. Description

Évaluer la pertinence des réponses

4.3.3. Contraintes

LLM actif (Llama 2)

4.3.4. Dépendances

SCN-002

4.3.5. Procédure de test

Poser des questions aux chatbot et analyser la pertinence et la compréhensibilité des réponses

4.4. Météo géolocalisée

4.4.1. SCN-004

4.4.2. Description

Tester affichage météo en fonction de la localisation

4.4.3. Contraintes

Avoir déjà rempli le formulaire

Avoir une clé API Valide

4.4.4. Dépendances

SCN-002

4.4.5. Procédure de test

Charger la page index.html et observé si la météo est renseigné dans la navbar. Il doit il y avoir 3 choses visible : la température, la ville et le bon icone météo.

4.5. Historique conversationnel

4.5.1. SCN-005

4.5.2. Description

Tester la recherche dans l'historique ainsi que la modification et la suppression d'élément

4.5.3. Contraintes

Données existantes

4.5.4. Dépendances

SCN-003

4.5.5. Procédure de test

Aller sur la page de l'historique grâce au bouton voir plus puis effectuer une Recherche par mot-clé ensuite modifier un élément de l'historique et enfin supprimer un élément. Résultat attendu : l'historique est mis correctement à jour en temps réel

4.6. Paramètres d'accessibilité

4.6.1. SCN-006

4.6.2. Description

Vérifier que les réglages s'appliquent globalement

4.6.3. Contraintes

localStorage activé

4.6.4. Dépendances

Aucune

4.6.5. Procédure de test

Dans l'onglet paramètre modifier les différents champs puis naviguer sur les différentes vues pour vérifier que tous fonctionne. L'affichage doit être cohérent sur toute les pages

5.Tests d'intégration

5.1. Intégration API météo

5.1.1. INT-001

5.1.2. Description

Vérifier l'intégration WeatherAPI

5.1.3. Contraintes

Clé API valide

5.1.4. Dépendances

SCN-004

5.1.5. Procédure de test

Comparer les données affichées et les données reçues via API. Il faut vérifier que les informations météo sont rapidement et correctement affichées. Vérifié aussi que le message affiché dans la bulle soit pertinent et dépend de la météo.

5.2. Liaison profil/conversation

5.2.1. INT-002

5.2.2. Description

Vérifier que le profil influence les réponses du chatbot

5.2.3. Contraintes

Données du formulaire remplies

5.2.4. Dépendances

SCN-002, SCN-003

5.2.5. Procédure de test

Modifier le profil de l'utilisateur et lancer une conversation avec le chabot. Il faut vérifier que les réponses soient adapté aux profils de l'utilisateur.

6.Tests unitaires

6.1. Gestion du localStorage

6.1.1. UNIT-001

6.1.2. Description

Vérifier lecture/écriture dans le navigateur

6.1.3. Contraintes

JS activé

6.1.4. Dépendances

Aucune

6.1.5. Procédure de test

Rafraîchir la page et vérifier que les données sont intacte et accessible.

6.2. Gestion du localStorage

6.2.1. UNIT-002

6.2.2. Description

Tester le script common.js pour l'accessibilité

6.2.3. Contraintes

JS activé

Paramètres modifiables

6.2.4. Dépendances

Aucune

6.2.5. Procédure de test

Appliquer les changements puis vérifier que tout est correct au niveau du DOM. Les valeurs doivent être correctement propagées.

7. Vérification de la documentation

- Cahier des charges : Alignement total avec les fonctionnalités développées
- Documentation utilisateur : Simple, claire
- Maquettes Figma : Respect visuel validé
- Commentaires dans le code : Conformés aux standards

8. Annexes

- Journal de tests
- Captures d'écran
- Liste des navigateurs testés
- Retours utilisateurs

9. Glossaire

- **Chatbot** : Agent conversationnel
- **LLM** : Large Language Model (ex. Llama 2)
- **localStorage** : Mécanisme de stockage dans le navigateur
- **Accessibilité** : Adaptation UI pour utilisateurs avec besoins spécifiques

10. Références

- Cahier des charges – v1.0
- Documentation WeatherAPI
- Maquettes Figma
- Documentation interne
- MDN : <https://developer.mozilla.org/fr/docs/Web/API/Window/localStorage>

11. Index

Chatbot

localStorage

Accessibilité

Météo