

Cahier de recette

Nom du projet

PRÉNOM NOM

Cahier de recette

Nom du projet

Les informations d'identification du document

Référence du document :	D6
Version du document :	1.01
Date du document :	3 juin 2025
Auteur(s) :	Marc Dieudonne

Les éléments de vérification du document

Validé par :	Nom de l'encadrant
Validé le :	03/06/2025
Soumis le :	03/06/2025
Type de diffusion :	Document électronique (.odt)
Confidentialité :	

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. Description de la fourniture.....	5
5. Moyen d'essai et outils.....	5
6. Conformité aux spécifications générales.....	6
7. Conformité aux spécifications fonctionnelles.....	6
7.1. Pour chaque scenario :.....	6
7.1.1. Identification.....	6
7.1.2. Description.....	6
7.1.3. Contraintes.....	6
7.1.4. Dépendances.....	6
7.1.5. Procédure de test.....	6
8. Conformité aux spécifications d'interfaces.....	6
9. Conformité de la documentation.....	7
10. Annexes.....	7
11. Glossaire.....	7

<u>12. Références.....</u>	<u>7</u>
<u>13. Index.....</u>	<u>7</u>

1.Introduction (ou préambule)

Ce cahier de recette décrit l'ensemble des tests fonctionnels permettant de valider le bon fonctionnement du prototype d'agent conversationnel pour les seniors, tel que défini dans le cahier des charges. Il couvre les fonctionnalités principales, les critères de réussite, les résultats attendus ainsi que le suivi de validation.

1.1. Objectifs et méthodes

Ce document vise à :

- Vérifier la conformité du prototype avec les besoins fonctionnels exprimés.
- Tester la robustesse de l'interface utilisateur, la pertinence du chatbot, l'accessibilité et les interactions.

La méthodologie utilisée repose sur :

- Des scénarios de tests détaillés.
- Des tests manuels réalisés en environnement contrôlé.
- Des ateliers de validation utilisateur (seniors).

1.2. Documents de référence

Cahier des charges – Agent conversationnel seniors

- Interfaces HTML (formulaire, index, paramètres...)
- Documentation API WeatherAPI
- Dossier Figma (maquettes interactives)

2.Guide de lecture

2.1. Maîtrise d'œuvre

Ce document est destiné à différents profils de lecteur. Il est structuré pour permettre à chacun de retrouver rapidement l'information nécessaire à son rôle.

2.2. Maîtrise d'ouvrage

Public : Commanditaire du projet, équipe produit

- **Utilisation** : Vérifier l'adéquation des livrables aux besoins exprimés

3. Concepts de base

- **Chatbot** : agent logiciel simule une conversation humaine, ici spécialisé en prévention des risques domestiques.
- **Accessibilité** : adaptation visuelle de l'interface (taille police, contraste...).
- **Stockage local** : données sauvegardées dans le navigateur de l'utilisateur (localStorage).
- **Profil utilisateur** : ensemble des données personnelles et médicales enregistrées via formulaire.
- **Bulle** : style d'affichage du texte dans l'assistant (dialogue, pensée, comics).

4. Description de la fourniture

Le livrable final comprend :

- Une application web responsive (HTML/CSS/JS)
- Sept interfaces :
 - Formulaire d'évaluation des risques
 - Interface de dialogue
 - Interface de personnalisation (paramètres)
 - Interface de profil utilisateur
 - Interface d'aide
 - Interface de numéro d'urgence
 - Interface de gestion de l'historique
- Stockage local des données utilisateur
- API météo intégrée
- Utilisation du LLM Llama 2
- Documentation utilisateur et technique

5.Moyen d'essai et outils

- Navigateurs : Chrome, Firefox, Edge (versions récentes)
- Outils de test : console navigateur, localStorage debugger, simulateur réseau
- Tests fonctionnels manuels selon scénarios
- Participants tests utilisateurs (personnes âgées)
- API WeatherAPI (clé test)

6.Conformité aux spécifications générales

Fonctionnalité	Test de recette	Critère de validation
Navigation entre vues	Cliquer sur tous les boutons/menu	Redirection correcte vers chaque section
Profil utilisateur	Création/modification du profil via formulaire.html	Données sauvegardées dans localStorage et affichées dans profil.html
Chatbot spécialisé	Poser des questions sur les risques domestiques	Réponses pertinentes générées par Llama 2
Météo géolocalisée	Vérifier l'affichage des données météo dans index.html	Données cohérentes avec WeatherAPI et conseils adaptés
Historique conversationnel	Recherche/modification dans historique.html	Fonction de suppression opérationnelle avec localStorage

Fonctionnalité	Test de recette	Critère de validation
Paramètres d'accessibilité	Changer police/fond dans settings.html	Changements appliqués sur toutes les pages (vérifier via common.js)
Contact d'urgence	Cliquer sur "Appeler" dans contact.html	Ouverture du composeur téléphonique avec le bon numéro

7. Conformité aux spécifications fonctionnelles

Ce chapitre décrit les scénarios de test permettant de valider les fonctionnalités exprimées dans le cahier des charges. Chaque scénario suit une structure détaillée afin de garantir la reproductibilité des tests et leur traçabilité.

7.1. Pour chaque scenario :

7.1.1. Navigation entre vues

Identification : SCN-001

Description : Vérifier que tous les liens de navigation permettent d'accéder aux différentes interfaces (formulaire, profil, paramètres, contact, historique).

- **Contraintes :** Test réalisé sur navigateur supportant JavaScript.
- **Dépendances :** Aucune.
- **Procédure de test :**
 - Cliquer sur chaque lien/bouton du menu.
 - Observer la redirection.
- **Critère de validation :** Chaque clic mène à la page cible sans erreur de chargement.

7.1.2. Profil utilisateur

Identification : SCN-002

Description : Tester la création et modification du profil à partir de `formulaire.html`, et vérifier l'affichage dans `profil.html`.

- **Contraintes** : localStorage activé.
- **Dépendances** : Aucune.
- **Procédure de test** :
 - Saisir des données dans chaque champ.
 - Revenir sur `profil.html`.
- **Critère de validation** : Les données sont sauvegardées et correctement restituées.

7.1.3. Contraintes

Identification : SCN-003

Description : Simuler une conversation avec le chatbot autour des risques domestiques (ex. “Comment sécuriser ma salle de bain ?”).

- **Contraintes** : LLM (Llama 2) actif sur le serveur local.
- **Dépendances** : SCN-002 (profil requis pour contexte).
- **Procédure de test** :
 - Poser plusieurs questions sur les risques quotidiens.
 - Lire les réponses affichées.
- **Critère de validation** : Réponses compréhensibles, cohérentes, adaptées à la question posée.

7.1.4. Météo géolocalisée

Identification : SCN-004

Description : Vérifier l’affichage météo dans l’en-tête de `index.html` en fonction de la position utilisateur.

- **Contraintes** : Formulaire Utilisateur remplie avec la localisation et clé API valide.
 - **Dépendances** : SCN-002 (profil requis pour contexte).
 - **Procédure de test** :
 - Autoriser la géolocalisation.
 - Recharger la page d’accueil.
 - **Critère de validation** : Température, icône météo et nom de la ville affichés correctement.
-

7.1.5. Historique conversationnel

Identification : SCN-005

Description : Tester les fonctionnalités de consultation, recherche et modification des messages dans `historique.html`.

- **Contraintes** : Données de discussion déjà générées.
- **Dépendances** : SCN-003 (chat nécessaire).
- **Procédure de test** :
 - Accéder à l’historique.
 - Rechercher un mot-clé.
 - Supprimer une entrée.
 - **Critère de validation** : Les actions sont enregistrées dans `localStorage` et l’affichage est mis à jour.

7.1.6. Paramètres d’accessibilité

- **Identification** : SCN-006
- **Description** : Vérifier que les réglages visuels dans `settings.html` (police, fond, style de bulle) s’appliquent à toutes les pages.
- **Contraintes** : `localStorage` activé.
- **Dépendances** : Aucune.

- **Procédure de test :**
 - Sélectionner une police, une couleur de fond et un style de bulle.
 - Naviguer sur l'interface d'accueil et le chatbot.
 - **Critère de validation :** Les choix sont appliqués de manière cohérente sur toutes les interfaces via `common.js`.

7.1.7. Contact d'urgence

- **Identification :** SCN-007
- **Description :** Vérifier que le bouton "Appeler" dans `contact.html` ouvre le composeur téléphonique avec le bon numéro.
- **Contraintes :** Test à réaliser sur tablette ou smartphone.
- **Dépendances :** Aucune.
- **Procédure de test :**
 - Cliquer sur le bouton d'urgence.
 - Observer le comportement du navigateur.
 - **Critère de validation :** Le composeur s'ouvre automatiquement avec le bon numéro prérempli.

8. Conformité aux spécifications d'interfaces

Les tests suivants valident l'ergonomie, l'accessibilité et la conformité visuelle des interfaces selon les standards définis en phase de maquettage.

Interface	Élément testé	Méthode de test	Résultat attendu
<code>index.html</code>	Affichage météo Interaction ChatBot	API WeatherAPI, Saiesie	Température, icône et ville affichées Réponse du Chatbot
<code>formulaire.html</code>	Validation des champs	Saisie, erreurs, rechargement	Comportement correct selon le remplissage

Interface	Élément testé	Méthode de test	Résultat attendu
settings.html	Personnalisation (police, fond, bulle)	Changement + navigation	Paramètres appliqués à toutes les pages
contact.html	Bouton d'appel	Clic sur bouton d'urgence	Lancement du composeur avec numéro correct
historique.html	Recherche et modification de texte	Rechercher, éditer, supprimer	Affichage mis à jour dynamiquement avec localStorage

9. Conformité de la documentation

Cahier des charges : Toutes les fonctionnalités testées y sont présentes

Documentation utilisateur : Clair, simple, avec captures

Maquettes Figma : Interfaces conformes à celles validées

10. Annexes

11. Glossaire

- Résultats bruts des tests manuels
- Captures d'écran des pages testées
- Journal de test avec identifiants des anomalies
- Liste des navigateurs et OS utilisés

12. Références

- Cahier des charges – v1.0
- Maquettes Figma
- Documentation WeatherAPI
- [MDN localStorage](#)
- Documentation interne du projet

13. Index

Liste les mots-clés du document et où les trouver dans celui-ci