

# Cahier des Charges

PIERROT Grégoire

SAA Anis

## 1. Contexte et Objectifs du Projet

### 1.1. Contexte

Le projet s'inscrit dans un contexte d'apparition de la robotique dans notre quotidien et où les étudiants et le personnel du CERI sont en recherche de multiples informations au cours des années d'études. Actuellement, les moyens de renseignements se limite à l'utilisation de cellulaires et d'ordinateurs personnels. L'opportunité est d'utiliser un robot Pepper, pour créer une plus grande interactivité entre l'informatique et le réel, grâce à un dialogue oral et tactile via une tablette.

### 1.2. Objectifs

- Développer une solution logicielle intégrée dans le robot Pepper qui servira d'assistant orientant vers les salles de cours (et autres dispositifs).
- Consultation de l'emploi du temps (Étudiants/Enseignants).
- Orientation des étudiants vers de multiples aides sociales et financières.
- Effectuer une recherche d'information plus interactive et rapide.
- Sécurité des données via utilisation d'un serveur flask avec protocole HTTPS.

## 2. Périmètre du Projet

### 2.1. Fonctionnalités attendues

- **Orientation** : Indication de l'emplacement de salle de cours et divers autres dispositifs (Hall d'entrée, BDE, Scolarité, WC, Machine à café ...)
- **Emploi du temps** : Récupération en temps réel d'un emploi du temps en fonction d'informations transmises.
- **Salle libre** : Récupération en temps réel de salles disponibles.
- **Aides sociales/financières** : Orientation de l'étudiant vers un service d'aide adapté en fonction d'une problématique précise.
- **Recherche** : Génération d'un code QR permettant le téléchargement d'un PDF contenant l'information recherchée.

## 2.2. Technologies utilisées

- **Langages de programmation** : Python (pour la logique d'IA), Choregraphe (outil de programmation pour Pepper).
- **Plateforme** : NAOqi OS (système d'exploitation de Pepper).
- **Base de données** : MongoDB pour stocker les données des salles de cours et autres dispositifs de l'établissement et stocker les données de différents aides aux étudiants.
- **Reconnaissance vocale** : Intégration de services comme Google Speech-to-Text ou Microsoft Azure Cognitive Services via la librairie SpeechRecognition sur python.
- **API** : L'API de l'EDT du CERI pour la consultation de l'emploi du temps.
- **LLM/Agent**: Utilisation du framework LangChain avec API Fireworks pour la création d'un modèle ainsi que d'un agent.
- **Logiciel de gestion de version décentralisée** : Git

## 3. Public Cible

### 3.1. Profil des utilisateurs

- **Âge** : 17 ans et plus.
- **Profession** : Étudiants et personnel du CERI.
- **Compétences techniques** : S'exprimer en français ou en anglais.
- **Besoins** : Renseignements (Salles/Emploi du temps/Documentation/Aide sociale).

### 3.2. Nombre d'utilisateurs pour les tests

- **Phase 1** : 4 utilisateurs pour des tests exploratoires afin de recueillir des premières impressions.
- **Phase 2** : 10 utilisateurs pour des tests en conditions réelles.

## 4. Scénarios de Tests Utilisateurs

#### **4.1. Objectifs des tests utilisateurs**

- Valider l'intuitivité de la communication vocale avec Pepper.
- Tester la pertinence des recherches et liens retournés.
- Évaluer la vitesse de réponses en utilisant l'api de CERI.
- Tester la validité de l'aide sociale proposée vis-à-vis de la problématique évoquée.
- Vérifier la bonne compréhension des utilisateurs concernant l'indication d'une salle.

#### **4.2. Méthodologie de tests**

- **Tests initiaux** : Réalisés pendant la phase de développement.
- **Tests modérés** : En présence d'un facilitateur pour observer les interactions et guider les utilisateurs.
- **Tests non modérés** : Réalisation de tests dans des espaces calmes, sans supervision directe, pour observer l'usage en conditions réelles.

#### **4.3. Scénarios de tests**

- Interagir avec Pepper pour un renseignement sur un sujet donné.
- Demander l'emplacement d'une salle libre.
- Énoncer un problème social spécifique.
- Consulter son emploi du temps.

### **5. Critères de Réussite et Évaluations**

#### **5.1. Indicateurs de performance**

- Taux de réussite des interactions vocales (>90%).
- Temps moyen pour effectuer une recherche (<2 minutes).
- Satisfaction des utilisateurs sur la pertinence des indications de salle (à 100%).
- Pertinence de la solution donnée (aide sociale) au problème évoqué (>80%).

#### **5.2. Feedback des utilisateurs**

- Collecte de feedback via des questionnaires après chaque session avec Pepper.
- Interviews qualitatives pour analyser les impressions et suggestions d'améliorations.

## 6. Contraintes et Risques

### 6.1. Contraintes techniques

- Compatibilité avec les infrastructures existantes (Wi-Fi, espace pour la mobilité de Pepper).
- Fiabilité de la reconnaissance vocale en environnements bruités.
- Autonomie insuffisante.

### 6.2. Risques potentiels

- Retard dans l'implémentation de la fonction de recherche.
- Possibilités d'erreurs dans les bases de données (entraînant une mauvaise indication de salle/aide sociale).
- Indisponibilité des robots lors de nos tests.
- Utilisation inappropriée de la fonction de recherche (sites indésirables).

## 7. Planning Prévisionnel

### 7.1. Phases du projet

- **Phase de conception** : 1 mois et demi (design des interactions et recherche de la faisabilité).
- **1<sup>ère</sup> phase de développement** : 2 mois et demi (développement de nos fonctionnalités, intégration avec Pepper).
- **Phase d'interrogation du public cible** : 1 semaine (sur de potentielles fonctionnalités supplémentaires).
- **2<sup>ème</sup> phase de développement** : 3 semaines (implémentations des idées conservées)
- **Phase de tests utilisateurs** : 2 semaines.

- **Ajustements post-tests** : 1 mois (optimisation des fonctionnalités et corrections de bugs).

## 7.2. Livrables

- **Conception des scénarios d'interaction et fonctionnalités** (fin de phase de conception).
- **Prototype fonctionnel de l'application sur Pepper** (fin de phase de développement).
- **Questionnaires des tests utilisateurs** (fin de phase de tests).

## 8. Équipe du Projet : PIERROT Grégoire M1 ILSEN / SAA Anis M1 IA

- **Chef de projet (Grégoire/Anis)** : Prise de décision sur les outils utilisés et gestion du git.
- **Organisation / Gestion du temps (Grégoire)** : Rédaction de la documentation (Cahier des charges, fiche de procédure), planification des jalons.
- **Développeur IA (Anis)** : Implémentation de fireworks, Reconnaissance vocale.
- **Développeur Robotique (Anis)** : Responsable de l'intégration des fonctionnalités sur Pepper (via Choregraphe).
- **Développeur Backend hors IA (Grégoire)** : Requêtes API, Gestion de base de données, serveur flask (protocole HTTPS)...
- **Testeur (Grégoire/Anis)** : En charge des tests utilisateurs et des retours sur l'expérience utilisateur.

## 9. Annexes

Modèle et Agent :

Fireworks : <https://docs.fireworks.ai/getting-started/introduction>

Librairies utilisées :

Langchain : <https://python.langchain.com/docs/introduction/>

Pyaudio : <https://people.csail.mit.edu/hubert/pyaudio/docs/>

SpeechRecognition : <https://pypi.org/project/SpeechRecognition/>

Serveur framework: Flask

Lien du Git : [https://github.com/anis-saa77/AMS\\_Project](https://github.com/anis-saa77/AMS_Project)