**Projet Algo C – DEWAREZ**

FAVAREL – PUTZ – BOISSIE

**Consignes : Développement d'un Chat-bot de Base en Langage C**

Vous devez créer un jeu textuel interactif, inspiré par la mécanique d'un chat-bot, où le joueur interagit via des commandes textuelles. Vous avez la liberté de choisir le thème du jeu (aventure, quiz, jeu de gestion, etc.), tout en respectant les contraintes techniques ci-dessous.

**Exigences du Projet :**

Gestion des Informations Utilisateurs :

Le système doit être capable de stocker des informations sur les utilisateurs, incluant mais ne se limitant pas à leur identité et un niveau d'importance ou de priorité attribué.

Réponses Contextuelles :

Le chat-bot devra générer des réponses en fonction du contenu des messages reçus et du contexte de la conversation.

Historique des Utilisateurs :

Lorsqu'un utilisateur connu interagit avec le chat-bot, son historique de conversation doit être chargé afin de personnaliser la session.

Mode Administrateur :

L'application doit inclure un mode administrateur, permettant de modifier le comportement du chat-bot (comme la logique de réponse ou les paramètres d'interaction) sans avoir besoin de recompiler le programme.

Fonctionnalités Extensibles :

Le chat-bot doit être conçu de manière à permettre l'ajout facile de nouvelles fonctionnalités ou d'améliorations dans le futur.

Sauvegarde des Données :

Le système doit inclure au moins deux bases de données distinctes. La première sera dédiée au stockage des informations des utilisateurs, tandis que la deuxième conservera l'historique des conversations. Il est conseillé de mettre en place une troisième base de données pour archiver les informations relatives aux réponses fournies par le chat-bot. Cette dernière base de données peut être adaptée ou étendue en fonction des besoins spécifiques du sujet traité.

Nota :

ici les bases de données sont représentées par des tableaux de structures chargés en mémoire et sauvegardées dans des fichiers (texte ou binaire au choix).

**Livrables :**

- Un document décrivant le thème et la façon dont vous avez abordé le projet

- La note personnelle concernant son travail et son rôle dans l'équipe

- La documentation d'installation et d'utilisation

- Architecture Logicielle (modules et principales fonctions)

- La liste des exigences  
- Le cahier de recette avec les exigences réalisées (OK) et celles non validées (FAIL)

- L'ensemble du code les fichiers C (.c .h)

**Cahier de recette :**

|  |  |  |
| --- | --- | --- |
| **N°** | **Exigence** | **Statut** |
| 1 | Environnement de développement : l’application sera développée en C. | OK |
| 2 | Sauvegarde des Conversations : L'application sauvegardera dans un fichier toutes les interactions avec les utilisateurs. |  |
| 3 | Base de Données Utilisateurs : Le fichier des utilisateurs contiendra l'identifiant, le nom, et une valeur d'importance de l'utilisateur. |  |
| 4 | Base de Données Historique : Le fichier d'historique conservera les messages échangés avec chaque utilisateur. |  |
| 5 | Base de Données des Réponses : Un fichier contiendra les déclencheurs (phrases ou mots-clés) associés aux réponses types. |  |
| 6 | Modes de Fonctionnement : L'application disposera d'un mode administration et d'un mode utilisateur. |  |
| 7 | Mode Administration - Gestion des Utilisateurs : Permet la création, modification et suppression des informations d’utilisateurs. |  |
| 8 | Mode Administration - Gestion des Réponses : Permet l'ajout, la modification, et la suppression des déclencheurs et des réponses associées. |  |
| 9 | Mode Utilisation - Interaction avec l'Utilisateur : Le chat-bot interagit avec les utilisateurs en générant des réponses adaptées. |  |
| 10 | Mode Utilisation - Historique de Conversation : L’utilisateur peut consulter l’historique de ses conversations. |  |
| 11 | Gestion du Contexte : Les réponses sont adaptées selon les messages précédents pour une continuité. |  |
| 12 | Personnalisation de l'Interaction : Les réponses sont personnalisées pour les utilisateurs récurrents. |  |
| 13 | Extensibilité du Chat-Bot : Le code est conçu pour permettre l'ajout de nouvelles fonctionnalités. |  |
| 14 | Sauvegarde des Données : Les données sont stockées en mémoire et sauvegardées en fichiers (texte ou binaire). |  |

**Organisation :**

### Membre 1 : Concepteur et Responsable de la Communication

* **Rôle technique : Concepteur**
  + Gère l’expression des besoins et la définition des fonctionnalités du chat-bot.
  + Rédige les exigences du projet, clarifie le thème et assure la cohérence avec les besoins des utilisateurs.
* **Rôle de gestion de projet : Communication**
  + Prépare et anime la présentation du projet.
  + S'assure que tous les membres comprennent bien les objectifs et les fonctionnalités.
  + Communique les progrès et les obstacles rencontrés pour aligner tout le monde.

### Membre 2 : Architecte Technique et Responsable Qualité

* **Rôle technique : Architecte Technique**
  + Conçoit l'architecture du programme et spécifie les fonctions nécessaires.
  + Assemble les parties du code produites par les développeurs et effectue les tests d'intégration.
  + Rédige la documentation technique sur les modules et fonctions.
* **Rôle de gestion de projet : Qualité**
  + Crée le plan de tests et définit les objectifs de validation.
  + S’assure de la mise au point des jeux de tests et vérifie la bonne exécution des tests.
  + Valide les exigences en fonction des critères de réussite (cahier de recette).

### Membre 3 : Développeur et Responsable Suivi du Projet

* **Rôle technique : Développeur**
  + Produit le code en C en suivant les spécifications.
  + Développe les fonctionnalités principales, comme la gestion des utilisateurs et des réponses.
  + Effectue des tests unitaires sur chaque fonction ou module codé.
* **Rôle de gestion de projet : Suivi du projet**
  + Suivi de l’avancement du projet et respect des délais.
  + S’assure que chaque tâche est réalisée selon le calendrier défini.
  + Ajuste le planning si besoin et communique les modifications aux autres membres.

|  |  |  |
| --- | --- | --- |
| **Membre** | **Rôle Technique** | **Rôle Gestion de projet** |
| Membre 1 | Concepteur | Communication |
| Membre 2 | Architecte Technique | Qualité |
| Membre 3 | Développeur | Suivi de projet |

**Architecture logicielle :**

L'architecture de ce projet peut être découpée en plusieurs modules principaux, chacun chargé d'une fonctionnalité spécifique du chat-bot. Les modules sont indépendants et interagissent via des structures de données et des fonctions exposées, ce qui rend le projet extensible et maintenable.

Modules et principales fonctions :

**Module Utilisateur**

Gère la création, la modification et la suppression des utilisateurs.

Fonctions principales : ajouterUtilisateur, modifierUtilisateur, supprimerUtilisateur, chargerUtilisateur.

Fichiers : User.c, User.h

**Module Historique**

Enregistre l'historique de chaque conversation dans un fichier et permet de le charger pour les utilisateurs existants.

Fonctions principales : sauvegarderHistorique, chargerHistorique.

Fichiers : Historique.c, Historique.h

**Module Réponses**

Gère les réponses prédéfinies du chat-bot en fonction de déclencheurs (mots-clés ou phrases).

Fonctions principales : ajouterReponse, modifierReponse, supprimerReponse, chercherReponse.

Fichiers : Reponses.c, Reponses.h

**Module Administrateur**

Permet aux administrateurs de configurer le chat-bot (ajouter/modifier des utilisateurs, configurer des réponses).

Fonctions principales : accesModeAdmin, configurerReponses, configurerUtilisateurs.

Fichiers : Admin.c, Admin.h

**Module Principal**

Gère le flux principal du programme, les modes d’utilisation et d’administration.

Fonctions principales : demarrerChatBot, modeAdmin, modeUtilisateur.

Fichiers : Main.c

**Idées :**

Ajouter logs peut permettre de debugger facilement

Ajouter module configuration (modifier les données sans recompilation)

Diapo de 2 à 3 slides qui place le contexte du projet (teaser, aide-mémoire)

Ajouter énigme ASCII dans le code (facultatif mais sympa)