

Plan de développement

Application de compagnon virtuel sur smartphone



Étudiants : Mathis Ruffieux et Néréïs Dugaleix

Encadrante : Mme. Lydie du Bousquet

Tuteur : Mr. Damien Pellier

Sommaire :

I) Introduction

II) Organisation des tâches

III) Cycle de vie

IV) Planification

V) Méthodes et outils

VI) Critères de qualité

VII) Annexes

I) Introduction

Ce document a pour but d'aider les développeurs de l'application à écrire correctement l'application. Nous énumérerons le travail à réaliser ainsi que les contraintes du projet afin de mettre en place un plan de développement efficace. Dans ce document nous expliquerons les principes de base du langage que nous allons utiliser, pour que chaque développeur puisse travailler correctement.

II) Organisation des tâches

Comme nous ne sommes que deux développeurs dans le projet, nous essayons de travailler sur chaque fonctionnalité ensemble. De ce fait, une personne pourra écrire le code et l'autre aider à l'améliorer en réglant les potentiels problèmes d'un point de vue extérieur.

Pour définir chaque tâche nous avons décidé de travailler de façon agile. Pour cette raison, nous avons commencé par définir les User Stories de notre projet. Une User Story est une phrase permettant d'énoncer les besoins d'une personne sur l'application. Cette phrase une fois détaillée permet de définir les points clé pour développer l'application afin de répondre à ce besoin.

Les User Stories :

L'ordre des User Stories ne représente pas un ordre de priorité, ni un ordre de développement de ces dernières.

De plus, cette liste est susceptible de changer ou d'être améliorée au cours du projet.

1) Apprentissage Flutter/Dart

En tant que développeur, je veux apprendre les langages Dart et Flutter afin de pouvoir programmer l'application.

2) Récupérer les données

En tant que scientifique, je veux récupérer les données d'un utilisateur afin de pouvoir les analyser.

3) Télécharger l'app

En tant qu'utilisateur, je veux télécharger l'application afin de pouvoir l'utiliser.

4) Changer les paramètres

En tant qu'utilisateur, je veux pouvoir accéder aux paramètres afin de les modifier.

5) Question du jour

En tant qu'utilisateur, je veux répondre à la question du jour afin de discuter avec le chatbot.

6) Journal

En tant qu'utilisateur, je veux ouvrir le journal afin d'y écrire des notes personnelles.

7) Récupère CSV

En tant qu'utilisateur, je veux récupérer un fichier CSV afin de mettre à jour mon application.

8) Envoie CSV

En tant que scientifique, je veux envoyer un fichier CSV afin que les utilisateurs puissent le télécharger de manière manuelle ou automatique.

9) Rédaction Documents

En tant que développeur, je veux rédiger les documents afin de les envoyer au client (ou aux professeurs).

10) Prise en main du projet

En tant que développeur, je veux prendre en main le sujet afin de partir dans la bonne direction.

11) Personnage robot

En tant qu'utilisateur, je veux interagir avec le chatbot afin d'avoir un partenaire de discussion.

III) Cycle de vie

Méthode agile

Comme nous l'avons énoncé précédemment, nous travaillons en suivant des méthodes agiles. Pour cela, notre travail est découpé en différents sprints. Chaque sprint fonctionne sur une durée de travail précise. Ici, nous l'avons fixé à six jours/homme pour les deux premiers sprints et quatorze jours/homme pour les deux

derniers sprints. L'énumération des sprints peut être retrouvée en dessous de cette section et en annexe sur un diagramme de gantt.

Chaque sprint se déroule toujours de la même manière :

On commence par énoncer le travail déjà réalisé sur le précédent sprint, puis on énonce le travail à réaliser pendant ce sprint. Chaque développeur choisit son travail parmi la liste des User Stories. Il pourra ainsi travailler sur une nouvelle fonctionnalité. Si un développeur rencontre des problèmes liés à l'implémentation de l'un d'entre eux, il pourra le remettre dans la catégorie "à faire" en précisant le travail qu'il a déjà réalisé.

Chaque jour du sprint, les développeurs se réunissent pour énoncer le travail terminé et le travail qu'ils comptent réaliser pendant la journée.

Pour chaque sprint, une ou plusieurs réunions est effectuée avec le client pour connaître l'avancement du projet et convenir des fonctionnalités à effectuer.

IV) Planification

Pour la planification nous avons fait un diagramme de Gantt prévisionnel sur l'année 2021-2022, en annexe 1.

V) Méthodes et outils

Pour ce projet nous utilisons différentes applications et outils afin de réaliser son développement de manière efficace :

Outils de planification/organisation :

[Excel](#) : Réalisation de diagramme de gantt

[Git](#) : Logiciel pour versionner notre application et sauvegarder les données

[Notion](#) : Permet de réaliser les méthodes agiles sous forme de tableau

[Discord](#) : Pour la communication

Outils de développement :

[Visual Studio code](#) et [Android Studio](#) : Ide pour développer

[Emulateur android](#) : Afin de tester notre application

Les technologies utilisées :

[Flutter/Dart](#) : Permet de coder l'application

Librairie SQLite : Permet de créer des Base de données interne à l'application

Flutter flow : Framework permettant d'écrire du contenu flutter proprement

SQL : permet d'interagir avec la base de données

VI) Critères de qualité

Les critères de qualité de l'application mobile ont été définis avec le client. Nous souhaitons que l'application soit fluide, sans faute d'orthographe et qu'elle doive être adaptée à tous les utilisateurs en ayant une interface et des dialogues personnalisables.

Par ailleurs, nous nous sommes appuyés sur **les critères de Bastien et Scapin** vus en cours en guise de critères de qualités.

Guidage :

L'utilisateur est guidé tout au long de l'application à l'aide de feedbacks immédiats et le fonctionnement du système apparaît clairement.

Charge de travail :

La charge de travail est optimisée, il y aura un nombre d'interactions de la part de l'utilisateur limitées pour réaliser toute action (tel que les réglages ou l'ouverture du journal).

Par ailleurs, il n'y aura pas plus d'un certain nombre de phrases par bulle de dialogue du robot. Si un dialogue est long, il sera découpé en plusieurs bulles.

Contrôle explicite :

L'utilisateur doit avoir le contrôle du système. C'est-à-dire qu'il peut abrégé une conversation à tout moment sans être obligé d'aller au bout. Il peut aussi changer de sujet grâce aux options de réponses fermées qui lui sont proposées.

Pouvoir interrompre les traitements et revenir en arrière sont des actions importantes.

De plus, une action est effectuée à chaque interaction de la part de l'utilisateur, pour qu'il comprenne que l'action a bien été réalisée.

Adaptabilité :

L'interface doit être adaptable selon le contexte, les besoins et les préférences des utilisateurs. Par exemple, en changeant la taille de la police ou en changeant la langue de dialogue. Par ailleurs, en fonction des réponses fermées de l'utilisateur, on pourra voir si il préfère tel ou tel dialogue. Nous avons de plus envisager d'implémenter un système de likes pour avoir des feedback sur certaines phrases du robot.

Gestion des messages d'erreurs :

L'application ne devra pas diriger l'utilisateur vers une page où il sera bloqué. De plus, les boutons sur lesquels on ne souhaite pas qu'il clique seront grisés pour éviter les erreurs. Par exemple, lors du premier dialogue d'ouverture de l'application, on ne souhaite pas que l'utilisateur ouvre le journal, le bouton sera donc soit grisé soit non affiché. Chaque bouton affiché sans être grisé sur l'écran devra être fonctionnel afin de ne pas perdre l'utilisateur.

Homogénéité et cohérence :

L'affichage des informations est identique d'une page à l'autre. Les boutons seront situés au même endroit d'un scénario à l'autre. De plus, une charte graphique sera définie pour avoir une cohérence entre chaque page de l'application.

Signification des Codes et Dénominations :

Les icônes auront un sens, par exemple le bouton "like" sera sous la forme d'un cœur.

Compatibilité :

Comme c'est une application mobile, on s'attend à retrouver une interface de chat similaire à ce qu'on a l'habitude de voir, avec des bulles de l'utilisateur à droite et celles de notre interlocuteur à gauche.

VII) Annexes

Annexe 1 : Diagramme de Gantt prévisionnel

Tâches à effectuer	Octobre	Novembre	Décembre	Janvier	Février
Prise en main du Sujet					
Apprentissage de Flutter & Dart					
Définition des tâches					
Création de la maquette					
Tournage des interviews					
Rédaction des documents de la première soutenance					
Implémentation d'une interface de dialogue					
Implémentation du scénario de bienvenue					
Implémentation de la question du jour					
Implémentation des boutons					
Implémentation du journal					
Traduction de l'application en Anglais					
Traduction de l'application en Japonais					
Mises à jour des scénarios de manière automatique					
Possibilité de récupérer les données en local					
Phase de tests					
Rédaction de la documentation interne					
Rédaction des comptes rendu de RDVs					
Rédaction des autres documents de la soutenance finale					

Tâches à effectuer	Mars	Avril	Mai	Juin
Prise en main du Sujet				
Apprentissage de Flutter & Dart				
Définition des tâches				
Création de la maquette				
Tournage des interviews				
Rédaction des documents de la première soutenance				
Implémentation d'une interface de dialogue				
Implémentation du scénario de bienvenue				
Implémentation de la question du jour				
Implémentation des boutons				
Implémentation du journal				
Traduction de l'application en Anglais				
Traduction de l'application en Japonais				
Mises à jour des scénarios de manière automatique				
Possibilité de récupérer les données en local				
Phase de tests				
Rédaction de la documentation interne				
Rédaction des comptes rendu de RDVs				
Rédaction des autres documents de la soutenance finale				

Annexe 2 : Vidéo de présentation de la maquette

La vidéo se trouve dans le même répertoire que ce document, en format mp4.