**Dédicaces**

*Nous dédions ce modeste travail à nous même,*

*Et à Monsieur Ferid Helali qui nous beaucoup encouragé et soutenue.*

*A tous ceux qui nous connaissent et contribuent du près ou de loin à entamer ce travail et auxquels nous devons être reconnaissant.*

*Ahlem Mhamdi & Rihem Zairi*

**Remerciements**

*D'emblée nous tenons à remercier DIEU le tout puissant de nous avoir aidé à entamer et à terminer ce travail.*

*Nous devons être reconnaissante en remerciant infiniment nos encadrant universitaire* ***Mr Mohamed Bjaoui*** *ainsi que nos encadrant de stage* ***Mr Ferid Helali*** *qui nous ont beaucoup aidé et soutenu pour que ce travail soit bien organisé en nous mettant sur la bonne voie sur tous les plans et que sans eux ce travail ne pourrait voir le jour.*

*Tous nous remerciements les plus sincères à toute personne qui nous aidé de près ou de loin pour réaliser et présenter ce projet dans de bonnes conditions.*

## INTRODUCTION GENERALE

Au cours des dernières décennies, la révolution informatique a envahi tous les pays du monde, et cela inclut notre pays, la Tunisie, qui a connu un développement indéniable et puissant dans ce domaine.

L’évolution et les avancées technologiques de ces dernières années ont facilité l’apparition

d’applications pour les Smartphones ou « téléphones intelligents ». **Ces outils supposent une**

**avancée incroyable dans l’immédiateté et l’accessibilité de contenus** auparavant

uniquement disponibles offline dans des publications spécialisées. Les applications sont de

tous types et sont adaptées à toutes les tranches d’âge : jeux, divertissement, information.

Parmi elles les applications de psychologie pour les téléphones que tous les passionnés

de cette discipline devraient connaître.

Dans ce contexte, se déroule notre projet de fin d’études à l’Institut Supérieur des études Technologiques de Mahdia (ISET Mahdia). Il consiste à concevoir et à mettre en place

**« Une Application Mobile de psychologie»** dans le but d’enrichir mes connaissances informatiques théoriques acquises tout au long de mon parcours éducatif ainsi que découvrir le domaine professionnel.

Les méthodes de développement ont toujours essayé d’apporter un contrôle continu sur un projet tout au long de son processus de vie pour aboutir à la fin à un logiciel fiable qui satisfait les besoins du client.

Les démarches **agiles** étant à la mode et prouvent de jour en jour leurs efficacité par rapport aux méthodes classiques, nous avons opté pour **SCRUM** comme cadre méthodologique pour notre projet.

Le présent rapport est organisé en cinq chapitres. Le premier chapitre intitulé **« Cadre général du projet »**, sera consacré à la description de l’organisme d’accueil et de notre projet et ce en étudiant les éventuels problématiques tout en proposant notre solution.

Dans le deuxième chapitre, intitulé **« Planification et architecture »**, nous définissons le Backlog de notre produit, et nous présentons une vue architecturale et conceptuelle globale de notre application. Aussi, nous exposerons les outils et technologies utilisés pour le développement de notre solution.

Les chapitres suivants, se concentrent sur l’étude et la réalisation des différents sprints de notre projet, dégagés dans le chapitre précédent. Dans chaque sprint, nous commençons, par le Backlog du Sprint courant qui décrit les tâches à faire et ensuite nous présentons le diagramme de classes et les diagrammes de séquences. Dans ce dernier, nous allons exposer notre travail par des captures écrans sans oublier la revue de sprint.

En guise de conclusion, une évaluation des résultats atteints et les perspectives d’avenir du présent projet seront exposées.

# Chapitre 1

**Cadre général du Projet**

##### INTRODUCTION

Dans ce premier chapitre, nous nous intéresserons tout d’abord au cadre général de notre projet et ceci en présentant l’organisme d’accueil. Nous aborderons l’analyse de l’existant après l’exposition de la problématique, et nous exposerons ensuite la solution proposée. Enfin, nous terminerons ce chapitre en mettant l’accent sur la méthodologie de travail adoptée et nous énoncerons les raisons que nous ont poussés à ce choix.

##### PRESENTATION DE LA SOCIETE

Dans le cadre de notre stage de perfectionnement, nous avons réalisé notre stage au sein de

la société Alfa Computer & Consulting dans le but d’enrichir nos connaissances théoriques par

l’aspect pratique.

Alfa Computers est une agence des services numériques spécialisé en solutions interactives, en

création des sites internet, e-commerce, multimédia, design, développement et hébergement situé

à l’avenue BechirSfar 5100 Mahdia, Tunisie. Alfa computers a débuté son activité l’année 2009.

##### CADRE DE PROJET

**II.1 Contexte de projet**

Aujourd'hui, la santé mentale est un problème mondial de premier plan, le nombre de dépressions et d’anxiété augmentant de façon spectaculaire. Et pour ça on a décidé de créer notre application pour aider des gens déprimés, des gens stressé et toute personne avoir besoin une aide psychologique.

Cette application a pour objectif unique d’aide les gens qui ont une perturbation mentale.

###### II.1.1 Notion Emily

Emily est, en intelligence artificielle, un programme informatique qui simule un psychothérapeute en fournir des conseils à des personnes. Elle fonctionne par reconnaissance de formes et substitution des mots-clés dans les phrases produites.

**II.1.2 Les avantage**

Emily est à nombreux avantages qui s’expriment au niveau de l’individu :

* Meilleure gestion du temps
* Des économies d’argent
* Offre une réponse 24h/24 et 7j/7
* Répondre aux questions des utilisateurs ou les aider à résoudre ses problèmes
* N’est jamais stressé

**II.1.3 Les inconvénients**

Emily a des avantages mais aussi des inconvénients :

* La collecte de données fait de l’utilisateur une cible
* N’est pas toujours bons à improviser
* Ne peut pas détecter l'ironie dans une phrase
* Ne peut pas faire preuve d'empathie
* Ne peut pas saisir les nuances d’une voix

##### II Analyse de l’existant

**ii.2.1 Description de L’existant**

il existe une chatbot en ligne qui permettent aux curieux de poser des questions pour comprend ses états émotionnels mais il échoue très rapidement si vous lui posez quelques questions complexes.

###### L’Agent conversationnel (chatbot) « Eliza» :

À l'aide d'une méthodologie de « correspondance de modèles » et de substitution, eliza donne des réponses prédéfinies qui ont donné aux premiers utilisateurs l'impression de parler à quelqu'un qui a compris leur contribution.

La figure suivant présente le bot :



***Figure 1: eliza***

**II.2.2 Critique de l’existant**

Il existe une solution mais il n'utilise pas les techniques modernes de développement surtout au niveau Front-End ce qui met en cause l Interface Homme Machine d’un point de vue confort :

- l'utilisateur a des problèmes lors de l'interaction avec le bot car c pas les gens ont pc portables ou compris comment utilise : absence d’une version mobile

- Au niveau de performance : un temps de réponse important, surtout en cas de nombreux utilisateurs utilisent l'application en même temps

- au niveau sémantique : les réponses ne sont pas clair.

**Solution Proposée**

Pour remédier aux lacunes déjà mentionnées dans le paragraphe précédent, nous proposons comme solution, la conception et le développement d’une application mobile sous forme d’un chatbot qui facilite l’interaction avec les gens curieux, et qui permet de connais son état psychologique et de prend les recommandations nécessaire pour être en bonne santé.

##### LA DEMARCHE ADOPTEE

La complexité croissante des systèmes informatiques a conduit les concepteurs à s’intéresser aux méthodes de développement. Ces dernières ont toujours essayé d’apporter un contrôle continu sur un projet tout au long de son processus de vie pour aboutir à la fin à un logiciel fiable et de qualité et pour garantir de rester fidèle aux besoins du client.

##### II.1 Choix du cadre méthodologique

Etant à la mode, les démarches agiles prouvent de jour en jour leur efficacité par rapport aux méthodes classiques caractérisées par leur rigidité en définissant un cadre strict en termes du coût, du cadre et du périmètre. En se focalisant sur la satisfaction des clients, en autorisant le changement, en offrant une meilleure visibilité aux développeurs et en se basant sur des cycles courts, ces approches sont de plus en plus adoptées par de ne nombreuses entreprises, notamment de développement informatique.

Agile représente un ensemble de “méthodes et pratiques basées sur les valeurs et les principes du Manifeste Agile”, qui repose entre autres sur la collaboration, l’autonomie et des équipes pluridisciplinaires. [1] L’agilité ne représente pas un processus mais une culture, un état d’esprit, une philosophie et un ensemble de valeurs.

Sous le parapluie de l’agilité, on trouve plusieurs noms dont les plus connus sont XP, SCRUM, KANBAN. Pour notre projet, nous nous sommes orientés vers SCRUM.

##### 

##### III.5 Présentation du SCRUM

Ken Schwaber et Jeff Sutherland les créateurs de SCRUM le définissent dans son guide comme suit : « SCRUM est un cadre de travail itératif permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible. » SCRUM se base sur la théorie du contrôle empirique de processus, ou l’empirisme. L’empirisme soutient que les connaissances proviennent de l’expérience et d’une prise de décision basée sur des faits connus. [2]

La figure ci-dessous décrit le « processus » SCRUM en mettant l’accent sur ses principaux artefacts, rôles et évènements :



***Figure 2: Processus SCRUM***

Dans un contexte SCRUM, je vais devoir utiliser quelques termes propres à cette méthodologie. En voici les plus pertinents :

|  |  |
| --- | --- |
| **Terme** | **Définition** |
| **Backlog du produit** | La définition des besoins fonctionnels sous forme de (user story) |
| **Backlog du Sprint** | la liste des tâches à implémenter dans un sprint, classées par importances et état |
| **Produit partiel** | la liste des tâches à implémenter dans un sprint, classées par importances et état |
| **Mêlé quotidienne** | le SCRUM meeting et réunion organisée tous les jours debout qui dure au plus 15 minutes |

*Tableau 1 : Glossaire SCRUM*

|  |  |
| --- | --- |
| **Rôle** | **Mission** |
| **SCRUM Master** | Supervision de l’avancement du projet et des activités de l’équipe |
| **Propriétaire du produit (Product Owner)** | Présentation des caractéristiques et des fonctionnalités du produit à développer et approbation du produit à livrer |
| **L’équipe de développeurs** | Réalisation des user stories et élaboration des sprints |

*Tableau 2 : L'équipe d'un projet SCRUM*

##### III.6 UML

D’après la société SPARX, UML peut être défini comme suit :

« UML est un standard de l’industrie modelant la langue avec une notation

graphique riche et un ensemble complet de diagrammes et d’éléments.

En fait, je vais devoir utiliser quelques diagrammes UML afin de modéliser

des idées tout au long de la réalisation du produit. Je peux en citer le

digramme de cas d’utilisation, le diagramme de classes, le digramme de

séquences ainsi que le diagramme de déploiement.

**CONCLUSION**

Dans ce chapitre nous avons justifié le choix du framework SCRUM comme cadre méthodologique du travail pour la mise en œuvre de ce projet, et il sera utilisé tout au long des prochains chapitres dans ce rapport.

Le chapitre suivant sera consacré à l’étude des besoins fonctionnels et non fonctionnels, la spécification du Backlog de produit final et la préparation du planning de notre travail.

# Chapitre 2

**Planification et architecture**

##### INTRODUCTION

Ce chapitre sera consacré à présenter la planification et l’architecture de notre projet.

Nous commencerons par l’identification des acteurs de notre application. Puis, nous spécifierons les besoins fonctionnels et non fonctionnels tout en appliquant notre cadre méthodologique présenté dans le chapitre précédent. Enfin, nous donnerons un bref aperçu sur le matériel utilisé, les technologies et les langages de programmation utilisés pour la mise en place de notre plateforme.

##### CAPTURE DES BESOINS

##### Identification des acteurs

***Un acteur*** *est la personne ou le matériel qui interagit avec notre système afin de réaliser une valeur ajoutée.* Notre application fait intervenir deux acteurs principaux :

* **Emily (Psychothérapeute) :** cet acteur donne un diagnostic au curieux selon leur discussion.
* **Curieux :** cet acteur est toujours à la recherche de comprendre soi même et d’améliorer son vie.

##### Identification des besoins

###### Capture des besoins fonctionnels

La capture des besoins fonctionnels va nous permettre de préciser l’étude du contexte fonctionnel en décrivant les différentes façons qu’auront les acteurs d’utiliser la future application.

Notre application offrira à ses gens curieux une interface mobile et doit permettre à :

###### Emily de :

* + Faire un diagnostic selon discussion avec curieux.
  + Donner les conseils.

###### Un Curieux de :

* + Créer son propre compte pour se connecter à notre application pour bénéficier de nos services.
  + Répondre aux questions

###### Besoins non fonctionnels

Une fois les besoins fonctionnels sont bien définis, les besoins non fonctionnels doivent être pris en compte tout au long du processus de développement de notre application.

Les besoins non fonctionnels sont des besoins qui ont un aspect visible pour le curieux, mais qui ne sont pas reliés directement au comportement du système.

* **Besoins d’utilisation :** L’application devrait être cohérente au point de vue de l’ergonomie. Tous les standards d’ergonomies doivent être présents : interface curieux bien claire, conviviale et simple dans l’utilisation.
* **Besoins de sécurité :** L’accès aux services de l’application sera strictement réservé aux curieux inscrits, un accès avec login et mot de passe sera exigé.
* **Besoins de performance :** il s’agit d’optimiser le temps de chargements des activités de l’application et ceci par l’utilisation des bonnes et nouvelles pratiques du développement mobile.
* **Besoin d’extensibilité :** L’application doit permettre à tout moment l'ajout d'autres modules.

##### PILOTAGE DU PROJET AVEC SCRUM

##### Equipe et rôle

Nous présentons dans ce paragraphe les différents acteurs participant au déroulement des différentes phases de notre projet et leurs rôles associés à savoir le *Scrum team*, le *Product Owner* et le *Scrum Master*.

|  |  |
| --- | --- |
| **Rôle Scrum** | **Personne Affectées** |
| **Product Owner** |  |
| **Scrum Master** | Mr. Ferid Helali |
| **Devlopment Team** | Ahlem Mhamdi  Rihem Zairi  Mr. Ferid Helali |

***Tableau 1: Equipe et rôle***

##### Les fonctionnalités du Product Backlog

Le Product Backlog est une liste ordonnée de tout ce qui pourrait être requis dans le produit et est l’unique source des besoins pour tous les changements à effectuer sur le produit. Le Product Owner est responsable du Product Backlog dans son contenu, sa disponibilité et son ordonnancement. [3]

Le Backlog du produit est l’artefact le plus important de SCRUM. En effet, c’est l’ensemble des caractéristiques fonctionnelles et techniques qui composent le produit. ***Les caractéristiques fonctionnelles*** sont appelées des ***histoires utilisateurs (user story)*** tandis que ***les caractéristiques techniques*** sont nommées ***des histoires techniques (technical story)***.

On s’intéressera dans le présent projet aux histoires utilisateurs seulement, les histoires techniques quant à eux seront écartées.

Le tableau suivant représente le Product Backlog de notre plateforme. Chaque histoire utilisateur constitue **un item** du backlog et elle est caractérisée par un rang déduit à partir de sa priorité. Nous détaillons ici la signification des différents termes utilisés.

* **Id** qui représente l’identifiant d’histoires utilisateur ;
* **Thème** pour mieux ordonnez les histoires utilisateur comme exemple gestion de profil ;
* **User Story** comporte la description des histoires utilisateurs suivant la forme « *En tant que…Je veux…Afin que » ;*

###### Conditions de satisfaction

* **La priorité** des histoires utilisateur selon la valeur métier et l’ordre de réalisation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Thème** | **En tant que**  **…** | **Je veux...** | **Afin que** | **Condition de**  **satisfaction** | **Priorit é** |
| **1** | Inscription | Visiteur | M’inscrire sur l’application en créant mon compte. | Je devienne membre | * Le champ   E-mail sera du type adresse e- mail.   * Le nom d'utilisateur est unique pour chacun. * le mot de passe sera caché avec une confirmation de mot de passe. * Demander la confirmation d’inscription. * Après appui sur le bouton   « let’s be a friend », je me retrouve sur l’activité de  chatbot. | 1 |
| **2** | Authentification | Curieux | Me connecter | J’accède à mon espace | -Après authentification je me retrouve sur mon canal de chatbot. | 2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **7** |  |  |  |  |  | 4 |
| **8** |  |  |  |  |  | 4 |
| **9** |  |  |  |  |  | 5 |
| **10** |  |  |  |  |  | 5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

***Tableau 2: Backlog du produit***

##### DIAGRAMME DES CAS D’UTILISATION GLOBALE

Le diagramme de cas d’utilisation définit les besoins fonctionnels du système pour chaque acteur identifié.

La figure suivante présente le diagramme de cas d’utilisation globale.



***Figure 3: Diagramme de cas d’utilisation globale***

##### PLANIFICATIONS DES SPRINTS

En se basant sur le Backlog du produit et en tenant compte des priorités attribuées à ses items, l’équipe de développement de notre plateforme dont je fais partie s’est mis d’accord que la décomposition sera effectuée en se basant sur les objectifs (Goals). De ce fait, on aura :

**Sprint 1**

* **Goals**
  + Inscription
  + Authentification

**Sprint 2**

* **Goal**
  + Création du modèle
  + Entraînement du modèle

**Sprint 3**

* **Goal**
  + Création du chat bot

***Figure : Planification des Sprints***

1. **Planning prévisionnel des Sprints**

Avant de présenter les différents sprints qui composent notre Backlog du produit, nous évoquons le planning de notre travail qui peut prendre une période de trois mois. Tout au long des chapitres suivant, nous allons présenter les parties d’analyse, conception et présentation des interfaces avec une revue et une rétrospective de notre travail. Le travail sera distribué comme suit

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Phase** | **Mars** | | | | **Avril** | | | | **Mai** | | | | **Juin** | | | | |
| **S1** | **S2** | **S3** | **S4** | **S1** | **S2** | **S3** | **S4** | **S1** | **S2** | **S3** | **S4** | **S1** | **S2** | **S3** | **S4** | |
| **Etude du projet** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| **Familiarisation avec la technologie et le cadre méthodologique (SCRUM)** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| **Réalisation du Backlog** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| **Réalisation des**  ***Sprints*** |  |  |  |  |  |  |  |  | Sprint1 | | Sprint2 | | | Sprint3 | | |  |

***Tableau 3: Planning prévisionnel des Sprints***

##### ENVIRONNEMENT DE TRAVAIL

##### Environnement matériel

L’architecture matérielle utilisé pour la réalisation du cette plateforme est la suivante :

|  |  |
| --- | --- |
| **PC Portable** | **Acer , Asus** |
| **RAM** | 8GO, 8GO |
| **Processeur** | Intel® Core™i5-8250U 1.6GHz  Intel® Core™i5-6200U 2.8GHz |
| **Système d’exploitation** | Windows 10 |

***Tableau 4: Architecture matérielle***

##### Environnement de développement

**C:\Users\ACER ASPIRE 3\Pictures\Saved Pictures\Kotlin-Blog.png Kotlin :**

est un langage de programmation orientée objet (POO) de type statique, interopérable avec la machine virtuelle Java, les bibliothèques Java et Android. Kotlin représente un gain de temps pour les développeurs : plus concis, ce langage permet d'écrire moins de code et d'éviter les redondances. Il est compilable en JavaScript et en encodeur LLVM (Low Level Virtual Machine). [4]

**C:\Users\ACER ASPIRE 3\Pictures\Saved Pictures\flask.jpg Flask :**

Flaskest un micro framework web écrit en Python. Il est classé comme un micro framework car il ne nécessite pas d'outils ou de bibliothèques particulières. Il n'a pas de couche d'abstraction de base de données, de validation de formulaire ou tout autre composant où des bibliothèques tierces préexistantes fournissent des fonctions communes. Cependant, Flask prend en charge les extensions qui peuvent ajouter des fonctionnalités d'application comme si elles étaient implémentées dans Flask lui-même. [5]

**C:\Users\ACER ASPIRE 3\Pictures\Saved Pictures\Python-Logo-PNG-Image.png Python :**

Python est un langage de programmation open source créé par le programmeur Guido van Rossum en 1991. Il tire son nom de l’émission Monty Python’s Flying Circus. Il s’agit d’un langage de programmation interprété, qui ne nécessite donc pas d’être compilé pour fonctionner. Un programme ” interpréteur ” permet d’exécuter le code Python sur n’importe quel ordinateur. Ceci permet de voir rapidement les résultats d’un changement dans le code. [6]

**TensorFlow**:

TensorFlow est un outil open source d'apprentissage automatique développé par Google. Le code source a été ouvert le 9 novembre 2015 par Google et publié sous licence Apache.Il est fondé sur l'infrastructure DistBelief, initiée par Google en 2011, et est doté d'une interface pour Python, Julia et R2.TensorFlow est l'un des outils les plus utilisés en IA dans le domaine de l'apprentissage machine. [7]

**VI.3 Environnement logiciel**

**Android Studio :**

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux. [8]

**C:\Users\ACER ASPIRE 3\Pictures\Saved Pictures\pycharm.jpgPyCharm :**

PyCharm est un environnement de développement intégré utilisé pour programmer en Python.Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django. Développé par l'entreprise tchèque JetBrains, c'est un logiciel multi-plateforme qui fonctionne sous Windows, Mac OS X et Linux. [9]

Description de l'image EA-logo-type-col.gif.**Enterprise Architect :**

Entreprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne Sparx Systems. Couvrant, par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application, il est l'un des logiciels de conception et de modélisation les plus reconnus. [10]

**C:\Users\ACER ASPIRE 3\Pictures\Saved Pictures\jupyter.pngJupyter :**

Jupyter est une application web utilisée pour programmer dans plus de 40 langages de programmation, dont Python, Julia, Ruby, R, ou encore Scala2. C'est un projet communautaire dont l'objectif est de développer des logiciels libres, des formats ouverts et des services pour l'informatique interactive. Jupyter est une évolution du projet IPython. Jupyter permet de réaliser des calepins ou notebooks. [11]

**C:\Users\ACER ASPIRE 3\Pictures\Saved Pictures\xampp.pngXAMPP :**

Xampp est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique.Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. [12]

**Anaconda :**

Anaconda est une distribution libre et open source2 des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), qui vise à simplifier la gestion des paquets et de déploiement. [13]

**VI.4 Les outils agiles :**

**C:\Users\ACER ASPIRE 3\Pictures\Saved Pictures\AnyDesk.pngAnyDesk :**

AnyDesk est une application de bureau à distance distribuée par AnyDesk Software GmbH. Le programme logiciel propriétaire fournit un accès à distance indépendant de la plate-forme aux ordinateurs personnels et autres dispositifs exécutant l'application hôte. Il offre des fonctionnalités de contrôle à distance, de transfert de fichiers et de VPN. [14]

**C:\Users\ACER ASPIRE 3\Pictures\Saved Pictures\Git.pngGit :**

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. Le principal contributeur actuel de git et depuis plus de 16 ans est Junio C Hamano. En 2016, il s’agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes. [15]

**VII. ARCHITECTURE GENERALE DE L’APPLICATION**

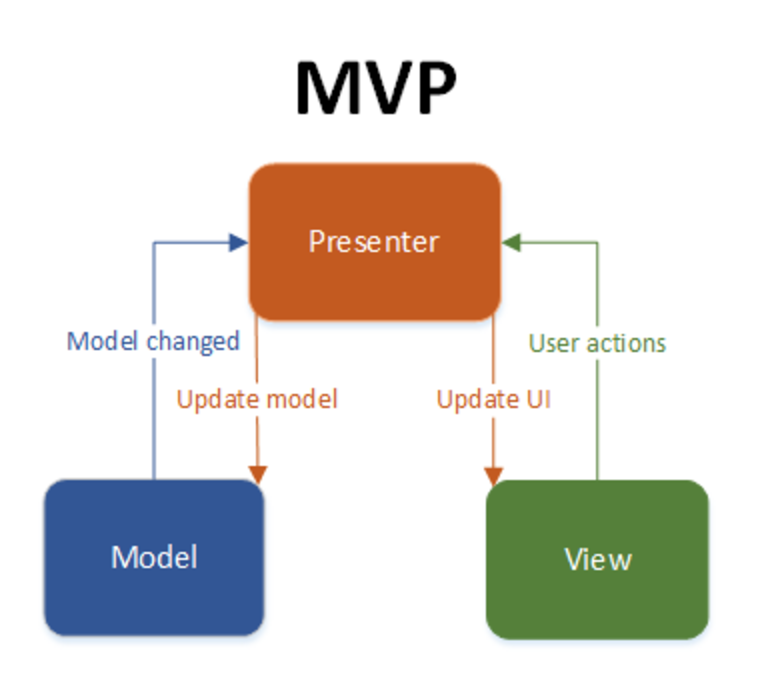
**VII.1 Architecture physique**

L’architecture physique est décrite par l’ensemble des composants matériels supportant l’application. [16]

La figure suivante représente l’architecture physique de notre application :

**VII.2 Spécification logicielle du système**

L’architecture logique de notre application se basera sur le MVP. MVP est l’acronyme de Modèle-vue-présentateur, est une dérivation du modèle architectural modèle-vue-contrôleur (MVC) et est principalement utilisé pour créer des interfaces utilisateur. Il sépare le développement de l'interface graphique avec logique métier, logique back-end et logique de présentation dans le but de rendre le code plus clair et maintenable.



**Figure 11: Architecture MVP**

MVP est un modèle architectural d’interface utilisateur conçu pour faciliter les tests unitaires automatisés et améliorer la séparation des préoccupations dans la logique de présentation :

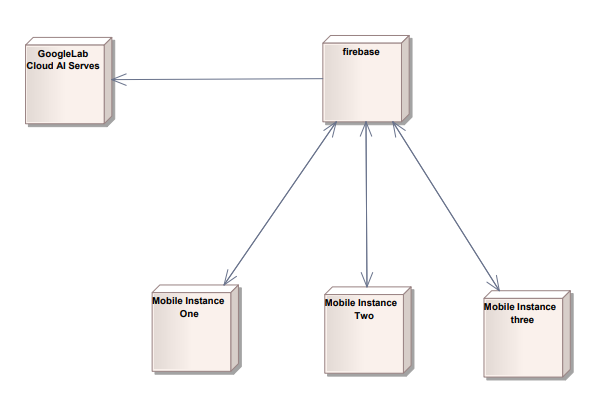
**Le modèle :** est une interface définissant les données à afficher ou à traiter d'une autre manière dans l'interface utilisateur.

**La vue :** est une interface passive qui affiche les données (le modèle) et achemine les commandes utilisateur (événements) vers le présentateur pour agir sur ces données.

**Le présentateur :** agit sur le modèle et la vue. Il récupère les données des référentiels (le modèle) et les formate pour l'affichage dans la vue. [17]

**VIII. DIAGRAMME DE DEPLOIEMENT**

Dans ce paragraphe, nous nous intéressons à l’architecture physique de notre application traduite par le diagramme de déploiement fournie par le langage UML. Ce dernier montre la disposition physique des matériels qui composent notre système. Les ressources matérielles sont représentées sous forme de nœuds.



**Figure 13: Diagramme de déploiement**

**CONCLUSION**

Dans Ce chapitre nous avons décrit les besoins fonctionnels et non fonctionnels de l’application et les rôles des différents acteurs. Ensuite, nous avons défini le groupe de travail, créé un backlog et listé les sprints, c'est la première étape de la démarche que nous avons adoptée. Plus tard, nous avons présenté les environnements matériels et logiciels utilisés tout au long du développement de l'application. Puis, nous avons décrit l'architecture de l'application. Dans le prochain chapitre, nous commencerons à développer notre premier sprint défini.

**Chapitre 3** **Étude préalable et état de l’art**

## Introduction :

Ce chapitre est consacré à une étude détaillée de ce qu’il existe sur le marché, étude comparative des solutions d’architecture et il se termine par une présentation de quelques concepts.

### Agent conversationnel(Chatbot)

### C’est quoi un Agent conversationnel (Chatbot)?

Un chatbot est un robot qui joue un rôle d’un assistant personnel disponible 24/24 et 7/7, il sait reconnaître des questions posées par vos contacts après l’entraîner à donner les bonnes réponses, il peut proposer des différentes choix pour préciser une question , il peut alimenter une conversation pour arriver à la bonne information et il s’améliore chaque jour, ses réponses sont de plus en plus précises et vous êtes de nouveau disponible pour les réponses personnalisées.

Un chatbot est un programme informatique capable de simuler une conversation avec un utilisateur par échange vocal ou textuel et grâce à un enclenché à l’encontre d’un signal ou d’une donnée.il est capable de répondre aux vessions en utilisant l’intelligence artificielle(IA) et le traitement du langage naturel [(NLP). [20]](#_bookmark161)

### 

### Les type de Chatbots :

Actuellement, trois types de chatbots cohabitent.

**Les chatbots basiques :** élaborés à base de simples arbres de décisions, permettent d’avancer dans sa demande à travers une série de questions proposées par le bot. [21]

❑ **Exemple d’un Chatbot basique:**



**FIGURE 3.1: *Chatbot*** [***basique***](#_bookmark163)

**Les chatbots intelligents :** comprennent le langage naturel et sont dotés d’intelligence artificielle.

* **Exemple d’un Chatbot intelligent :**



**FIGURE 3.2: *Chatbot*** [***intelligent***](#_bookmark163)

**Les chatbots hybrides :** mêlant scripts et IA avec parfois un renvoi à l’humain sur des cas plus complexes.

* **Exemple d’un Chatbot hybride :**



**FIGURE 3.3: *Chatbot*** [***hybride***](#_bookmark163)

**Aucune catégorie n’est meilleure qu’une autre.**

### Les familles de chatbots :

Suivant l’expérience d’interactions, On distingue principalement quatre « familles » de

Chatbots :

* **Chatbot serviciel :** apporte un service précis à l’utilisateur.
* **Chatbot expérientiel : fait vivre une expérience.**
* **Chatbot commercial :** réalise une transaction commerciale.
* **Chatbot Entertainment :** fonctionne comme divertissement.

### Les approches techniques pour créer une IA conversationnelle

Il y a plusieurs approches mais dont notre cas nous mettons en œuvre seulement deux approches les plus [reconnus. [22]](#_bookmark165)

* + - 1. **Chatbots basés sur la récupération :**

Un chatbot basé sur la récupération utilise des modèles d’entrée et des réponses prédéfinis. Il utilise ensuite un type d’approche heuristique pour sélectionner la réponse appropriée. Il est largement utilisé dans l’industrie pour créer des chatbots orientés objectifs où nous pouvons personnaliser le ton et le flux du chatbot pour conduire nos clients avec la meilleure expérience.



**FIGURE 2.4: *Chatbots basés sur la*** [***récupération***](#_bookmark166)

* + - 1. **Chatbots génératifs :**

Les modèles génératifs ne sont pas basés sur certaines réponses prédéfinies. Ils sont basés sur des réseaux de neurones seq 2 seq.

C’est la même idée que la traduction automatique. En traduction automatique, nous traduison

Le code source d’une langue vers une autre langue mais ici, nous allons transformer une entrée en une sortie. Il a besoin d’une grande quantité de données et est basé sur des réseaux Deep Neural.

**FIGURE 3.5: *Chatbots*** [***génératifs***](#_bookmark167)

* **étude comparative entre les chatbots basés sur la récupération et les chatbots génératifs :**

|  |  |  |
| --- | --- | --- |
|  | **chatbots basés sur la récupération** | **chatbots génératifs** |
| **Facile** | Oui | Non |
| **Réponse intelligente** | Non | Oui |
| **Réponse prédéfinie** | Oui | Non |
| **Erreur grammaticale** | Non | Oui |
| **Cas invisible** | Non | Oui |
| **Plus intelligent** | Non | Oui |
| **Difficile à entraîner** | Non | Oui |
| **Dataset massive** | Non | Oui |

**TABLE 2.1:** **les chatbots basés sur la récupération vs les chatbots génératifs**

### Synthèse :

Selon l’analyse effectuée sur le Chatbot Emily dans le premier chapitre, nous avons pris la décision d’utiliser cet outil dans notre projet car il offre les fonctionnalités ci-dessous :

* **Type de chatbot (chatbot intelligent)** : à l’aide d’un algorithme intelligent intégré, le chatbot est capable d’interagir avec les utilisateurs de notre application en langage humain, chaque requête envoyée au chatbot est en texte brut, puis elle est analysée en temps opportun et renvoyer la réponse à l’utilisateur.
* **Famille de chatbot :** Notre chatbot appartient à la **famille psychologie**, car il propose un service basé sur un accès interactif aux informations à l’aide d’une information dynamique partagée entre l’utilisateur final et l’outil.
* **Approche de chatbot(basés sur la récupération) :** L’outil Chatbot joue un rôle important en aidant les utilisateurs à identifier les cours qui correspondent le mieux à leurs objectifs pédagogiques en analysant leurs questions lors des conversions on récupérant la réponse convenable.
  1. **L’intelligence artificielle démystifiée (pré requis pour comprendre l’agent virtuel) :**

L’intelligence artificielle, les robots conversationnels, les chatbots. Ce sont des termes au monde qui retentissent pratiquement dans tout dialogue traitant de CRM et de centres de service client. Mais savons-nous réellement leurs significations ? Et quelles valeurs mesurables apportent-ils dans la gestion de la relation client ?

Nous appelons intelligence artificielle (IA) toute forme d’interaction qu’une machine exerce avec son entourage et ce d’une manière qui imite l’humain. L’algorithme dicte et conditionne un déroulement d’opérations enclenché à l’encontre d’un signal, d’une donnée. Les signaux que sait lire un système à intelligence artificielle peuvent représenter des images, des signes graphiques, des sons.

En théorie, tout fonctionnement cérébral et comportement humain pourrait être traduit en algorithmes, ce ne serait qu’une question de capacité d’élaboration d’algorithmes suffisamment complexes.

Avec le temps et la progression de la recherche, des algorithmes de plus en plus recherchés sont élaborés et par conséquent, l’intelligence artificielle s’approche progressivement du niveau de l’intelligence [humaine. [23]](#_bookmark168)

* 1. **Traitement du langage naturel :**

Le traitement du langage naturel ou NLP (Natural Language Processing) est défini comme la capacité d’une machine à analyser, comprendre et générer un discours humain.

Le but de la NLP est de faire des interactions entre les machines et les humains qui peuvent ressembler exactement à des interactions entre les humains.

La NLP est un domaine de l’informatique qui existe depuis un certain temps,et elle devient de plus en plus utilisé au cours des dernières années grâce aux progrès de la technologie qui ont facilité le développement des machines dotés de capacités NLP.

Les applications de la NLP compris les filtres anti-spam,réponses aux questions, les dialogues et d’autres applications.

Pour comprendre comment elle fonctionne, nous devons examiner ses deux principales composantes :

* **NLU (Natural Language Understanding) :** est la partie la plus difficile de la NLP,de comprendre ou de donner un sens au langage naturel que l’ordinateur a reçu. [24]
* **NLG (Natural Language Generation) :** est beaucoup plus simple à accomplir. C’est la traduction de langage artificiel d’une machine en texte, ou en un discours audible avec une synthèse vocale.

## L’apprentissage automatique (Machine learning)

L’apprentissage automatique (ML) : est une application de l’intelligence artificielle (IA) qui offre aux systèmes la possibilité d’apprendre et de s’améliorer automatiquement à partir de

L’expérience sans être explicitement programmés.

L’apprentissage automatique se concentre sur le développement de programmes informatiques qui peuvent accéder aux données et les utiliser pour apprendre par eux-mêmes.

Les algorithmes d’apprentissage automatique sont souvent classés comme supervisés ou non [supervisés [25]](#_bookmark170)

* faire des prédictions algorithmiquement : **Apprentissage supervisé**
* déchiffrer algorithmiquement les modèles dans les données : **Apprentissage non supervisé**

### Apprentissage supervisé :

* Le concept de base est d’utiliser des données étiquetées pour former des modèles prédictifs.
* Les données étiquetées désignent des observations où la vérité terrain est déjà connue.
* Les modèles de formation signifie caractériser automatiquement les données étiquetées de manière à prédire les étiquettes pour points de données inconnus.
* Par exemple : un modèle de détection de fraude par carte de crédit peut être formé à l’aide d’un enregistrement historique d’achats frauduleux. Le modèle résultant estime la probabilité qu’un nouvel achat soit frauduleux.
* Les méthodes courantes d’apprentissage des modèles vont des régressions de base aux réseaux neuronaux complexes.
* Tous suivent le même paradigme connu sous le nom d’apprentissage supervisé.

Exemple d’algorithmes de ML supervisé : **Classification, Régression**

1. **Apprentissage non supervisé :**

* Un autre paradigme de modélisation connu sous le nom d’apprentissage non supervisé tente de faire surface sous-jacent modèles et associations dans les données lorsqu’aucune vérité terrain existante n’est connue (c.-à-d sont étiquetés).
* Dans cette large catégorie de méthodes, les plus couramment utilisées sont les techniques de clustering, qui détectent de manière algorithmique quels sont les regroupements naturels qui existent dans un ensemble de données.
* Par exemple, le clustering peut être utilisé pour apprendre par programme les segments de clientèle naturels dans la base d’utilisateurs d’une entreprise.
* D’autres méthodes non supervisées pour l’extraction des caractéristiques sous-jacentes comprennent : le principal analyse de composants, modèles de Markov cachés, modèles de sujets, etc. Exemple d’algorithmes ML non supervisés : **Clustering**

1. **Étude comparative :**

|  |  |  |
| --- | --- | --- |
| **Paramètres** | **Apprentissage supervisé** | **Apprentissage non supervisé** |
| **Processus** | Dans un modèle d’apprentissage supervisé, les variables d’entrée et  de sortie seront données. | Dans le modèle d’apprentissage non supervisé, seules les données  d’entrée seront fournies. |
| **Des données d’entrée** | Les algorithmes sont formés à l’aide de données étiquetées. | Les algorithmes sont utilisés contre des données qui ne sont pas  étiquetées. |
| **Algorithmes utilisés** | Supporte la machine vectorielle, le réseau neuronal, la régression linéaire et logistique, la forêt aléatoire et les arbres de  classification. | Les algorithmes non supervisés peuvent être divisés en différentes catégories : comme les algorithmes de cluster, les K-means, le  clustering hiérarchique, etc. |

|  |  |  |
| --- | --- | --- |
| **Complexité**  **informatique** | L’apprentissage supervisé est une  méthode plus simple. | L’apprentissage non supervisé est  complexe en termes de calcul. |
| **Utilisation des données** | Le modèle d’apprentissage supervisé utilise des données de formation pour apprendre un  lien entre l’entrée et les sorties. | L’apprentissage non supervisé n’utilise pas les données de sortie. |
| **Exactitude des**  **Résultats** | Méthode très précise et fiable. | Méthode moins précise et fiable. |
| **Apprentissage**  **en temps réel** | La méthode d’apprentissage se  déroule hors ligne. | La méthode d’apprentissage se  déroule en temps réel. |
| **Nombre de**  **Classes** | Le nombre de classes est connu. | Le nombre de classes n’est pas  connu. |
| **Principal inconvénient** | La classification du big data peut être un véritable défi dans l’apprentissage supervisé. | Vous ne pouvez pas obtenir d’informations précises sur le tri des données et la sortie en tant que données utilisées dans l’apprentissage non supervisé est  étiquetée et n’est pas connue. |

**TABLE 2.2:** **Comparaison entre Apprentissage supervisé et Apprentissage non supervisé** [**[26]**](#_bookmark171)

1. **Synthèse :**

Dans la section précédente, nous avons présenté les fonctionnalités de l’outil chatbot implémenté dans le projet, qui est un outil intelligent capable de prendre des décisions basées sur l’analyse

de données étiquetées en temps réel en utilisant un concept de «Machine Learning» qui fait référence à l’apprentissage supervisé pour garantir un service optimale à nos utilisateurs.

* 1. **Deep learning :**

L’apprentissage en profondeur est une fonction d’intelligence artificielle (IA) qui imite le fonctionnement du cerveau humain dans le traitement des données et la création de modèles à utiliser dans la prise de décision.

L’apprentissage en profondeur est un sous-ensemble de l’apprentissage automatique dans l’intelligence artificielle qui dispose de réseaux capables d’apprendre sans surveillance à partir de données non structurées ou non étiquetées. Également appelé apprentissage neuronal profond ou réseau neuronal profond. L’apprentissage en profondeur, un sous-ensemble de l’apprentissage automatique, utilise un niveau hiérarchique de réseaux de neurones artificiels pour mener à bien le processus d’apprentissage automatique.

Les réseaux de neurones artificiels sont construits comme le cerveau humain, avec des nœuds de neurones connectés entre eux comme une toile.

Alors que les programmes traditionnels construisent l’analyse avec des données de manière linéaire, la fonction hiérarchique des systèmes d’apprentissage en profondeur permet aux machines de traiter les données avec une approche non [linéaire.](#_bookmark172) [27]

1. **Les couches de deep learning :**
2. **Artificial Neural Networks (ANN) :**

Les réseaux de neurones artificiels ANN sont des réseaux de neurones multicouches entièrement connectés qui ressemblent à la figure ci-dessous. Ils se composent d’une couche d’entrée, de plusieurs couches masquées et d’une couche de sortie. Chaque nœud d’une couche est connecté à tous les autres nœuds de la couche suivante. Nous approfondissons le réseau en augmentant le nombre de couches cachées.[[28]](#_bookmark173)



**FIGURE 2.6: *Artificial Neural Networks*** [***(ANN)***](#_bookmark176)

1. **Convolutional Neural Networks (CNN) :**

Les réseaux de neurones convolutifs (CNN) sont l’une des variantes des réseaux de neurones largement utilisés dans le domaine de la vision par ordinateur. Il tire son nom du type de couches cachées dont il se compose. Les couches cachées d’un CNN se composent généralement de couches convolutifs, de couches de regroupement, de couches entièrement connectées et de couches de normalisation. Ici, cela signifie simplement qu’au lieu d’utiliser les fonctions d’activation normales définies ci-dessus, les fonctions de convolution et de regroupement sont utilisées comme fonctions d’activation.[[29]](#_bookmark174)



**FIGURE 2.7: *Convolutional Neural Networks*** [***(CNN)***](#_bookmark177)

1. **Recurrent Neural Networks (RNN) :**

Les réseaux de neurones récurrents ou RNN comme on les appelle en bref, sont une variante très importante des réseaux de neurones largement utilisés dans le traitement du langage naturel. Dans un réseau neuronal général, une entrée est traitée à travers un certain nombre de couches et une sortie est produite, avec l’hypothèse que deux entrées successives sont indépendantes l’une de l’autre. Cette hypothèse n’est cependant pas vraie dans un certain nombre de scénarios réels. Par exemple, si l’on veut prédire le prix d’une action à un moment donné ou si l’on veut prédire le mot suivant dans une séquence, il est impératif de prendre en compte la dépendance aux observations [précédentes. [30]](#_bookmark175)



**FIGURE 2.8: *Recurrent Neural Networks*** [***(RNN)***](#_bookmark178)

1. **Étude comparative :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **ANN** | **CNN** | **RNN** | |
| **Type de données** | Données tabulaires,  Données textuelles. | Données de séquence | Données d’image | |
| **Partage de**  **paramètres** | Non | Oui | Oui | |
| **Entrée de**  **longueur fixe** | Oui | Oui | Non | |
| **Connexions**  **Récurrentes** | Non | Non | Oui | |
| **Dégradé de disparition et**  **d’explosion** | Oui | Oui | Oui | |
| **Relation spatiale** | Non | Oui | Oui | |
| **Performance** | ANN est considéré comme moins puissante que CNN, RNN. | CNN est considéré comme plus puissant qu’ANN, RNN. | RNN inclut moins de compatibilité des fonctionnalités par  rapport à CNN. | |
| **Application** | Reconnaissance faciale et vision par ordinateur. | Reconnaissance faciale, numérisation de texte et traitement du langage  naturel. | Conversions synthèse vocale. | de |

**TABLE 2.3:** **Comparaison entre ANN, CNN et** [**RNN**](#_bookmark179)

1. **Synthèse :**

Nous avons présenté dans la section précédente une étude comparative liée à tous les aspects d’apprentissage profond **"Deep Learning"** qui nous aidera à identifier les point fortes et aussi

Les faiblesses de chaque couche de cette technologie afin de nous permettre de choisir la meilleure architecture qui peut être adoptée pour notre outil chatbot :

* Les architectures d’apprentissage profond telles que les réseaux de neurones profonds, les réseaux de croyances profondes, les réseaux de neurones récurrents et les réseaux de neurones convolutifs sont des technologie qui peuvent être appliquées à des variétés de domaines, notamment la reconnaissance vocale, le traitement du langage naturel, la reconnaissance audio et traduction automatique, etc.
* L’apprentissage profond comporte de nombreuses couches (ANN, CNN et RNN) mais nous n’avons sélectionné que celle qui correspondait à nos besoins et également compatible avec l’outil chatbot.
* Chaque couche parmi les couches ci-dessus a ses points fortes et ses faiblesses, par exemple les couches CNN et RNN partagent la capacité de partage de paramètres, tandis que la couche ANN ne le permet pas, il y a aussi des fonctionnalités comme "Dégradé de disparition et d’explosion" qui est partagée entre toutes les couches.
* Les concepts de Deep Learning aident le chatbot à interagir de manière intelligent avec nos utilisateurs et à prendre également des décisions comme le fait de promouvoir des cours qui conviennent le mieux aux utilisateurs en fonction des conversations analysées.
* La couche ANN aide à l’ingestion et au traitement des données dans un format de texte brut, tandis que la couche RNN se chargera des données séquentielles et la couche CNN intégrera un fichier multimédia spécifique comme des images.
* Les couches CNN et RNN étant nécessaires pour traiter une quantité massive de données ce qui n’est pas le cas de notre projet où le nombre de cours que nous proposons est limité, nous avons décidé d’utiliser la couche ANN

1. **Cloud Computing :**

Le cloud computing est le livraison d’hébergement les services fournis à un client sur le L’Internet. (Activez des services à grande échelle sans investissement [initial)[31]](#_bookmark181)

**FIGURE 2.9: *Piliers du Cloud***

1. **Les caractéristiques du cloud :**

Cinq caractéristiques à retenir....

* + - 1. **Elasticité (rapid elasticity) :** Adaptation rapide des ressources à une variation du besoin.
      2. **Ressources en libre-service (self-service) :** Offertes à la demande sans intervention humaine.
      3. **Accès réseau ubiquitaire (location indépendance) :** Signifie qu’un utilisateur ou un

Client doit être en mesure d’accéder au service cloud de manière omniprésente et réactive, quel que soit son emplacement.

* + - 1. **Mise en commun des ressources (ressources pooling) :** Décrire une situation dans laquelle les fournisseurs desservent plusieurs clients «locataires» avec des services provisoires et évolutifs.
      2. **Service mesurable et facturable (Pay-as-you-go) :** Facture en fonction de l’utilisation.

1. **Modèles de déploiement :**

Il existe quatre modèles de déploiement :

**1. Cloud Public :**Le cloud est externe à l’organisation, accessible via internet, géré par un prestataire externe propriétaire des infrastructures, avec des ressources partagées entre plusieurs sociétés.

**2. Cloud Privé :**Dédié et accessible via des réseaux sécurisés, hébergé chez un tiers, mutualisé entre les différentes entités d’une seule même entreprise.

**3. Cloud Communautaire :**Il s’agit d’une variation du modèle de cloud public incluant uniquement les membres de plusieurs entreprises.

**4. Cloud Hybride :** Est la combinaison de cloud privé et de cloud public.

1. **Les Services cloud :**

Il existe trois services :

**1. Software as a Service(SaaS) :** Le logiciel est hébergé chez le fournisseur, dans son propre datacenter.

**2. Platform as a Service (Paas) :** Fournit des environnements de développement d’applications prêts à l’emploi, fonctionnels et performants.

**3. Infrastructure as a Service(Iaas) :** Désigne la mise à disposition de ressources matérielles (calcul, stocka et réseau) vitalisées.

1. **Synthèse :**

Le cloud computing est la disponibilité à la demande des ressources du système informatique sans gestion active directe par l’utilisateur qui permet de déployer des applications sur le Web en utilisant l’un des quatre principaux modèles déjà présentés dans la section précédente.

Nous avons pris la décision de déployer notre application et l’outil chatbot sous la forme d’une URL locale on utilisant une solution "cloud public" afin qu’elle puisse être accessible et disponible pour tous nos utilisateurs partout via internet par le biais d’un service IaaS qui met à notre disposition des API de haut niveau utilisées pour déréférencer divers détails de bas niveau de l’infrastructure réseau sous-jacente comme des ressource physique, emplacement, partitionnement des données, mise à l’échelle, sécurité, sauvegarde, etc.

**Conclusion :**

Durant ce chapitre, nous avons étudié les chatbots ainsi que les approches et les méthodes utilisées pour l’implémenter. Après nous avons discuté la visualisation des données, cloud computing et le système répartie.

Alors que dans le chapitre suivant, nous allons introduire notre solution avec une conception détaillée.

**Chapitre 4 Implémentation de chatbot :**

**Introduction**

Ce chapitre sera consacré sur le travail réalisé dans la partie IA comme d'habitude en passant par des étapes bien déterminées.

Ces étapes visent la réalisation de la construction de chatbot.

1. **Étude expérimentale :**

Dans ce chatbot, nous avons utilisé le « modèle seq2seq », également appelé modèle encodeur-décodeur, qui utilise la mémoire à long court terme-LSTM pour la génération de texte à partir du corpus d'apprentissage. Ce modèle prédit un mot donné dans l'entrée utilisateur, puis chacun des mots suivants est prédit en utilisant la probabilité de probabilité que ce mot se produise. Nous avons utilisé une méthode appelée 'forcing enseignant' pour entraîner le décodeur qui lui permet de prédire les mots suivants dans une séquence cible donnée dans les mots précédents.

J’ai simplifié la construction de ce chatbot en 5 étapes :

1. **Étape 1 : Prétraitement du jeu de données**

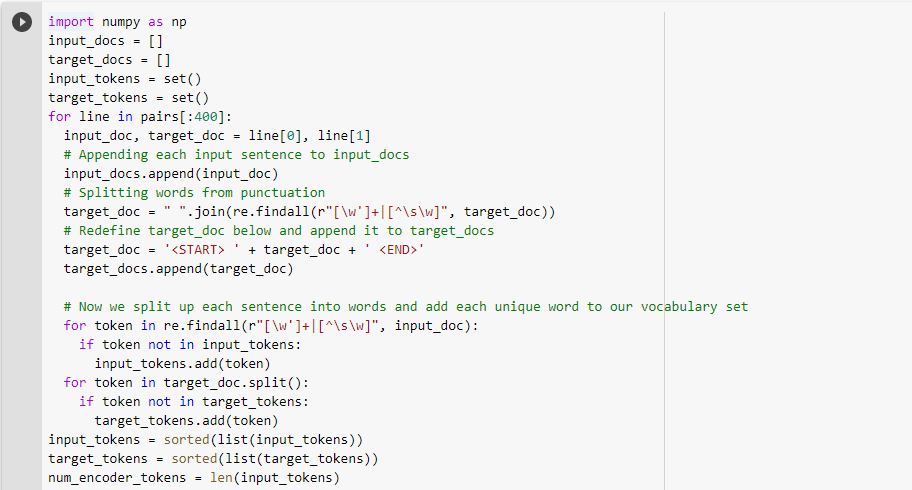
Nous utilisons des mots réguliers pour supprimer des groupes. Ensuite, nous créons des réponses du robot des réponses humaines aux partenaires pour former leurs structures SEQ2SEG.



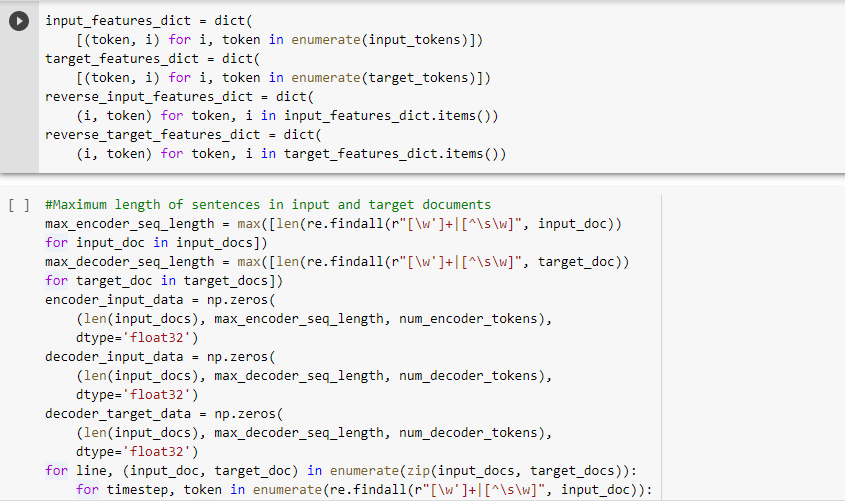
**FIGURE : Prétraitement du jeu de données**

1. **Étape 2 : Configuration de la formation**

Nous avons implémenté trois matrices de vecteurs one-hot, les données d'entrée du codeur, les données d'entrée du décodeur et les données de sortie du décodeur. La raison de l'utilisation de deux matrices réside dans le forçage de l'enseignant qui est utilisé par le modèle seq2seq tout en formant le jeton d'entrée du pas de temps précédent pour aider le modèle à prédire le jeton cible actuel.



**FIGURE : 1 ère partie de configuration de la formation**

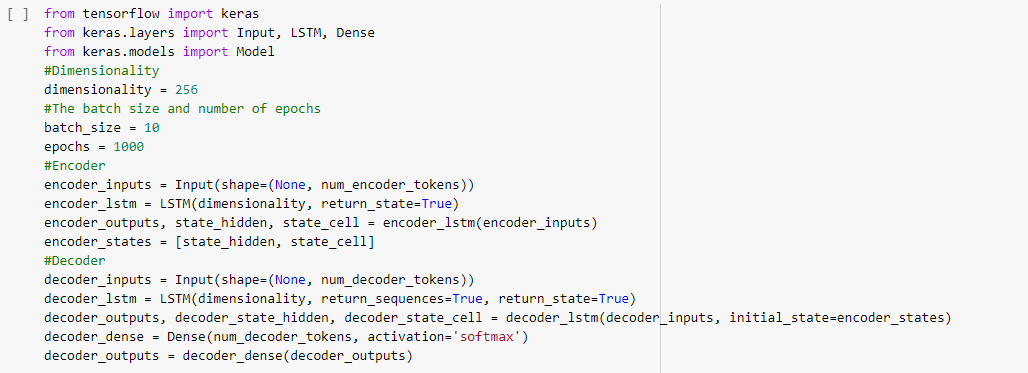


**FIGURE : 2 éme partie de configuration de la formation**

1. **Partie3 : Configuration de la formation encodeur-décodeur**

Notre modèle d'encodeur nécessitait une couche d'entrée qui définit une matrice pour contenir les vecteurs one-hot et une couche LSTM avec un certain nombre d'états cachés. La structure du modèle de décodeur est presque la même que celle de l'encodeur, mais ici nous passons les données d'état avec les entrées du décodeur.

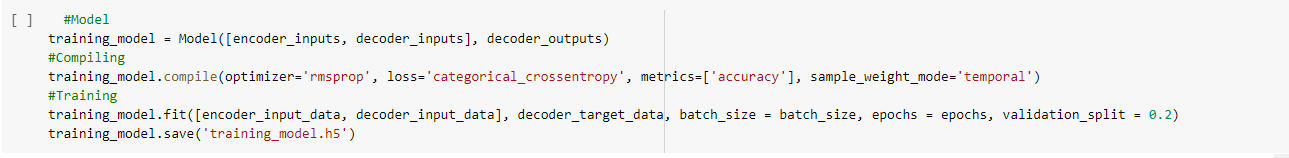
Nous avons importé Keras, LSTM, Dense de tensorflow pour nous aider à construire le modèle.



**Figure : 1 ère partie de Configuration de la formation encodeur-décodeur**

1. **Partie 4 : Création et formation du modèle seq2seq**

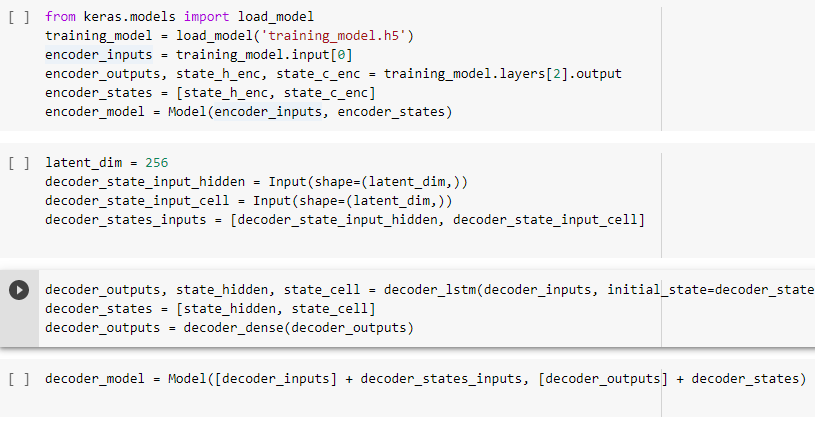
Nous avons utilisé rmsprop comme optimiseur et categorical\_crossentropy comme fonction de perte. Nous appelons la méthode .fit() en donnant les données d'entrée du codeur et du décodeur (X/input) et les données cibles du décodeur (Y/label).



**Figure: Création et formation du modèle seq2seq**

1. **Partie5 : Configuration de test et modèle final**

Nous avons établi la couche d'entrée du décodeur, les états finaux du codeur, les sorties du décodeur de la couche dense du décodeur, et les états de sortie du décodeur qui est la mémoire pendant le réseau d'un mot à l'autre.



**Figure: Configuration de test et modèle final**

**Conclusion**

Dans ce chapitre, nous avons présenté la partie de construction du chatbot en passant par Prétraitement du jeu de données, configuration de la formation, Configuration de la formation encodeur-décodeur, Création et formation du modèle seq2seq ,Configuration de test et modèle final. Dans le chapitre suivant, nous présenterons l’étude et la réalisation du premier sprint.

**Chapitre 5 Etude et réalisation du Sprint 1**

**INTRODUCTION**

Dans Ce chapitre, nous allons déterminer le travail qui a été faite pendant la première itération (sprint). En fait, chaque sprint représente une vision de la production de production.En effet, chaque itération représente une vision distribuée de la période de la production livrable. Le développement de chaque sprint passe par les étapes d’analyse, de conception et de réalisation. Le premier sprint vise la réalisation des items « Authentification, Inscription»

1. **LE BACKLOG DU SPRINT 1**

L’équipe Scrum doit obligatoirement se réunir pour définir le but du sprint avant d’y se lancer. Ce but doit être défini en terme métier pour qu’il soit compréhensible par les membres en dehors de l’équipe. Il s’agit de répondre à une question fondamentale « pourquoi faisons nous ce sprint ?». Le but du présent sprint est de réaliser les deux Items :

* S’inscrire
* S’authentifier

Il est le temps maintenant de décider quelles histoires de notre Backlog seront incluses dans le Backlog du sprint. Les estimations des histoires sont définies en jour. Le but de ce sprint est de réaliser des fonctionnalités basiques de l’application.

|  |  |
| --- | --- |
| **ID U. S** | **User Stories** |
| **1.1** | En tant qu’un curieux, je veux m’inscrire sur l’application en créant mon  compte. |
| **2.1** | En tant que curieux, je peux me connecter et accéder à mon  profil. |

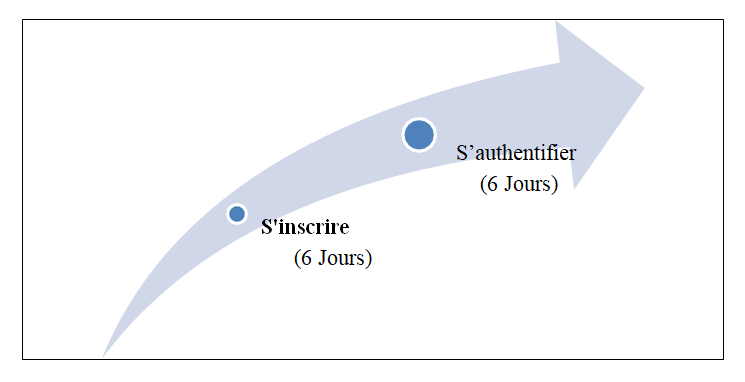
***Tableau 5: Les user stories de sprint 1***

Le tableau suivant représente le Backlog du premier Sprint :

|  |  |  |  |
| --- | --- | --- | --- |
| Items | User Stories | Taches | Estimation de temps |
| **S’inscrire** | M’inscrire sur l’application | Créer les IHM | 3 jours |
| Contrôle sur les champs | 3 jours |
| **S’authentifier** | Me connecter à mon compte | Créer les IHM | 3 jours |
| Contrôle sur les champs | 3 jours |

***Tableau 6: Backlog de "Sprint 1"***

Notre premier sprint s’étalera donc sur 12 jours répartis comme suit :

******

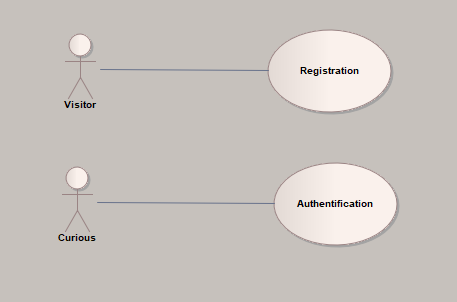
***Figure 7: Décomposition sprint 1 en Items***

##### SPECIFICATION FONCTIONNELLE DU SPRINT 1

Selon le Backlog du premier sprint, nous présentons le diagramme des cas d’utilisation global du premier sprint avec une description textuelle.

##### Diagramme des cas d’utilisation globale de « Sprint 1 »

La figure suivante présente le diagramme des cas d’utilisation globale du premier sprint.



***Figure 8: Diagramme des Cas d'utilisation " Sprint 1 "***

##### Description textuelle des cas d’utilisation de « Sprint 1 »

###### 1. Item « S’inscrire »

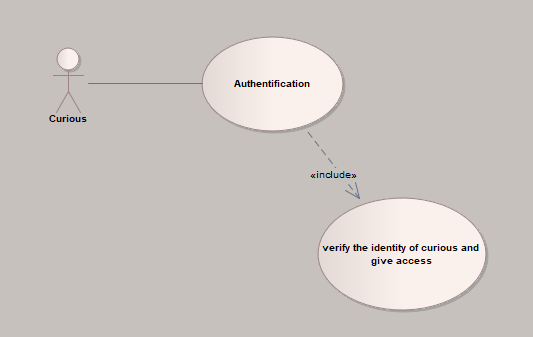
|  |  |
| --- | --- |
| **Description textuelle de cas d’utilisation « S’inscrire »** | |
| **Titre** | S’inscrire |
| **Acteur** | Visiteur |
| **Prés condition** | Visiteur non Inscrit |
| **Post condition** | Visiteur inscrit est possède un compte |
| **Description du scénario nominal** | 1. Le visiteur remplit le formulaire d’inscription 2. Le visiteur Confirme la saisie de ses données 3. Le système enregistre la demande et affiche un message de confirmation 4. Le système effectue une redirection vers l’activité chatbot. |
| **Scénarios alternatif** | **2.a les informations sont incorrectes :** Le système affiche un  message d’erreur et l’enchaînement se démarre à l’action 2. |

***Tableau 7: Description textuelle du cas d'utilisation "S'inscrire"***

###### 2. Item « S’authentifier »

**A. Raffinement du cas d’utilisation « S’authentifier »**

La figure suivante présente le raffinement de l’item « S’authentifier » par l’utilisateur

******

***Figure 9: Diagramme des cas d'utilisation de l’item" S'authentifier "***

###### B. Descriptions textuelles du cas d’utilisation « S’authentifier »

|  |  |
| --- | --- |
| **Description textuelle de cas d’utilisation « S’authentifier »** | |
| **Titre** | S’authentifier |
| **Acteur** | Curieux |
| **Prés condition** | Curieux non authentifié |
| **Post condition** | Curieux authentifié |
| **Description du scénario nominal** | 1. Le Curieux saisie son nom et mot de passe. 2. Le Curieux confirme la saisie des données. 3. Le système vérifie la saisie des données. 4. Le système affiche l’interface chatbot. |
| **Scénario d’exception** | **3.a les informations sont incorrectes :** l’enchaînement démarre à l’action 3, le système affiche un message d’erreur  pour que le curieux vérifie ses données. |

***Tableau 8: Description textuelle de cas d'utilisation "S'authentifier"***

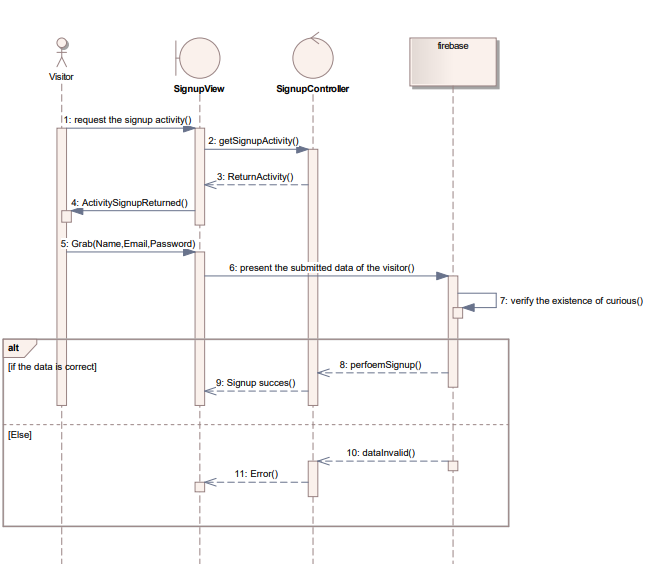
##### CONCEPTION DU SPRINT 1

La conception est l’activité suivante pour ce sprint, elle lui donne une architecture et une forme claire et bien définit. Elle se traduit par les diagrammes de séquence et le diagramme des classes participantes.

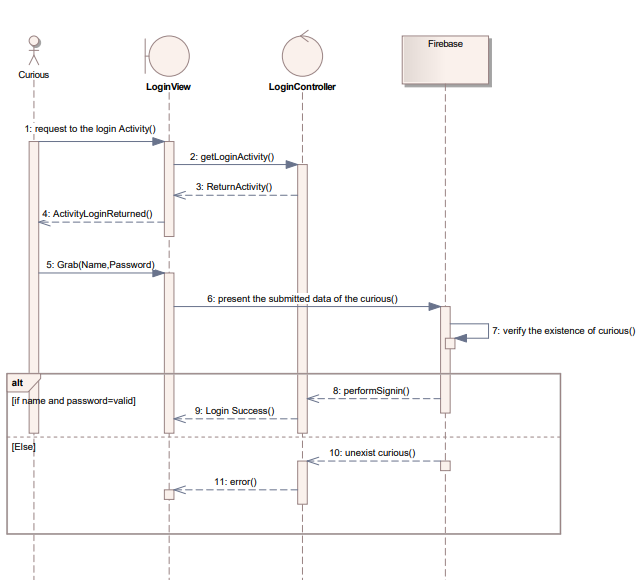
##### Diagrammes de séquence

Pour schématiser la vue comportementale de notre système informatique, nous faisons recours au diagramme de séquences d’UML. Ce diagramme permet de présenter les interactions entre l’acteur et le système avec des messages présentés dans un ordre chronologique.

En nous référant aux descriptions textuelles dans la section précédente, nous présentons les diagrammes de séquences adéquats. Sur la base de ces descriptions, nous pouvons constater que certains cas d’utilisations sont similaires. C’est pour cette raison que nous avons choisi de sélectionner quelques exemples pour les traiter.



***Figure 10: Diagramme de séquence du cas d’utilisation" S'inscrire "***



***Figure 11: Diagramme de séquence du cas d’utilisation " S'authentifier "***

##### REALISATION DU SPRINT 1

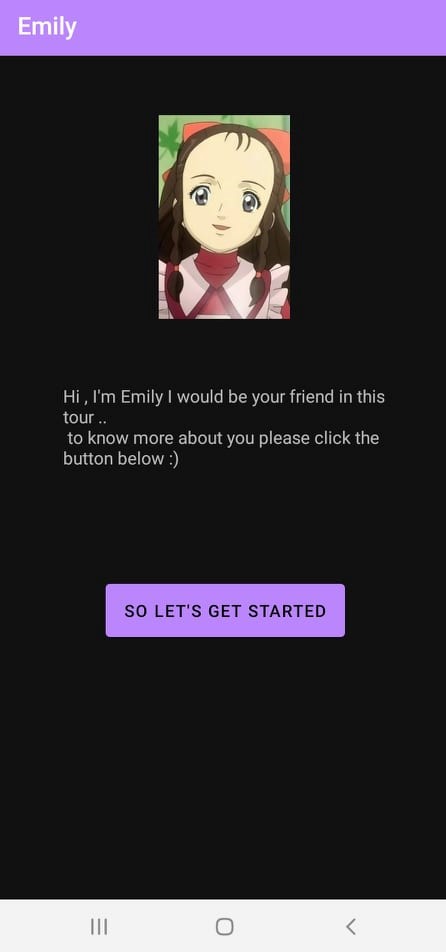
Dans cette partie, nous allons présenter des captures d’écrans des différentes interfaces développées pendant ce sprint.

L’interface principale est accessible sans authentification. Il s'agit de la première interface que vous trouverez lors de l'ouverture de l’application contenant le bienvenu.

Un Curieux doit s’authentifier, pour sa il doit enter leur nom et mot de passe :

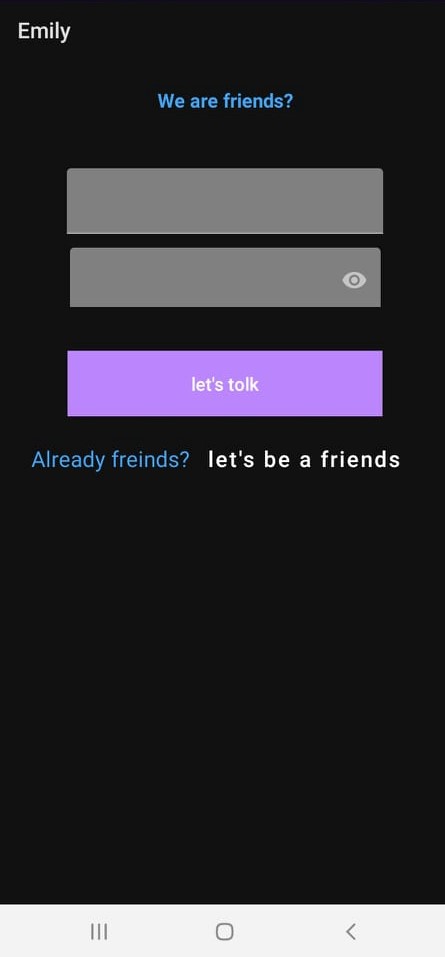
Avec un nom et un mot de passe valide, il est redirigé directement ver le chat bot :

Si non, va créer son propre compte par la clique sur la bouton «  let’s be a friend  », pour s’inscrire il doit remplir le formulaire suivant :



***Figure 12: Activité d'accueil***

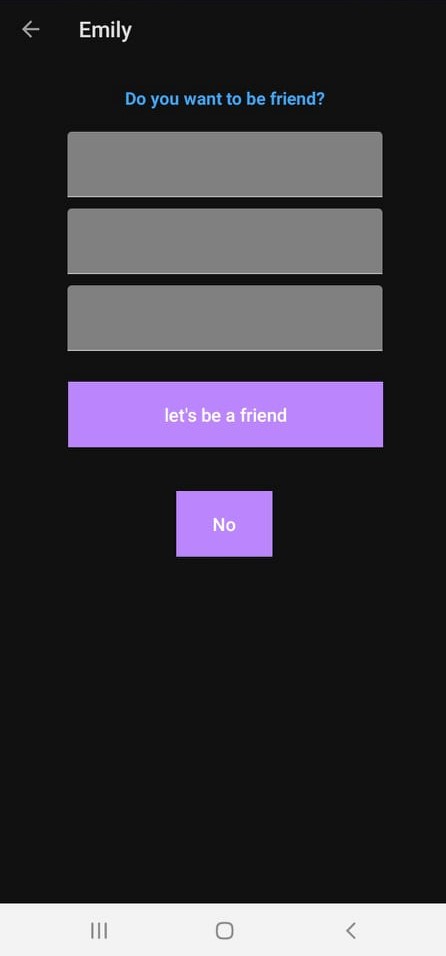
Un Curieux doit s’authentifier, pour sa il doit enter leur nom et mot de passe :



***Figure 13: Interface de connexion***

Avec un nom et un mot de passe valide, il est redirigé directement ver le chat bot :

Si non, va redirigé pour créer son propre compte par la clique sur la bouton «  let’s be a friend  » ,pour s’inscrire il doit remplir le formulaire suivant :



***Figure 14: Interface d'inscription***

##### REVUE DE SPRINT 1

*Une revue de Sprint (Sprint Review) est tenue à la fin du Sprint pour inspecter l’incrément réalisé et adapter le Backlog Produit si nécessaire. Pendant la revue de Sprint, l'équipe Scrum et les parties prenantes échangent sur ce qui a été fait durant le Sprint*. [18]

Notre premier sprint étant terminé, une réunion de revue de sprint est effectuée pour valider l’incrément développé avec le Product owner. Nous avons testé les fonctionnalités de ce dernier. Cette étape garantit le bon fonctionnement du système à travers une comparaison entre les comportements attendus et les résultats obtenus.

##### RETROSPECTIVE DE SPRINT 1

Dans le langage agile, le sprint rétrospectif s'inscrit dans le principe d'amélioration continue. L'objectif est que le prochain sprint soit plus efficace que le précédent et ainsi de suite. C'est une méthode empirique, c'est-à-dire basée sur l'expérience et sur l'auto-apprentissage. [19]

* Ce qui a fonctionné : Le travail sur ce sprint s'est bien déroulé.il n'y a rien de mal à ce sprint.
* Ce que nous pouvons améliorer :
  + - Les performances de l'interface peuvent être améliorées.
    - Le temps nécessaire pour accomplir une tâche une fois que vous commencez à maîtriser les compétences que vous utilisez.
* Améliorations : lors de l’inscription on peut ajouter la notification par mail.

Le sprint 2 peut maintenant démarrer bien évidemment après une réunion de planification.

##### CONCLUSION

Tout au long de ce chapitre, nous avons présenté le premier sprint en passant par l’analyse, la conception, la réalisation sans oublier la revue de sprint et la rétrospective. Dans le chapitre suivant, nous dévoilons les fonctionnalités du deuxième sprint.