

TABLE DES MATIERES

1. Introduction	2
2. Intérêt	2
3. Contexte	3
a. Le client	3
b. Les produits existants	4
1. Vector	4
2. Google assistant	5
4. Analyse de la problématique	5
a. Identification des besoins techniques.....	5
b. Le choix des technologies	5
1. Support électronique et logiciel :	5
2. Communication électronique – logiciel :	6
3. Les composants	7
4. esthétique.....	8
5. Site web	9
6. Développement du projet.....	10
A. Méthodologie.....	10
B. Logiciel	10
C. Électronique	17
1. schéma électronique	18
D. Site internet	18
E. Base de données	19
F. VPS	19
7. Analyse de sécurité	19
A. électronique	19
B. SITE weB.....	19
C. VPS	20
D. Services	21
8. Problèmes rencontrés	21
A. Raspberry Pi	21
B. ESP32-CAM	21
C. Sortie audio intégrée au robot.....	21
D. Capteur de mouvement	21
E. Moteurs DC	21
9. Pistes d'amélioration	22

A.	impression 3d.....	22
B.	Meilleurs moteurs dc	22
C.	fonctionnalités supplémentaires	22
D.	Contrôle domotique.....	23
E.	recharge automatique	23
F.	Application de configuration.....	23
10.	Conclusion.....	23
11.	Bibliographie	24

1. INTRODUCTION

Dans notre monde actuel, les nouvelles technologies sont en pleines explosions. Chaque jour, nous faisons de nouvelles découvertes dans l'objectif d'avoir un monde meilleur. Cependant, ces nouvelles technologies sont très souvent destinées aux jeunes personnes. Nos aînés sont souvent mis à l'écart voir même oubliés.

Ayant encore la chance d'avoir mes grands-parents, j'ai pu observer le gouffre technologique entre eux et les jeunes de notre génération.

Dans le cadre de mon Travail de fin d'étude, je me suis alors demandé comment je pourrais aider les personnes âgées en créant un outil qui leur viendrait en aide, mais qui en même temps ne les force pas à devenir des experts nouvelle technologie.

J'ai donc pris pas mal de temps à trouver une solution à cette problématique.

Lors de deux conférences données par Frédéric Carbonnelle et Estelle Auberix lors du DevDay 2019, j'ai petit à petit pensé qu'un robot serait une solution adéquate.

La conférence donnée par F. Carbonnelle concernait l'intelligence artificielle. Durant celle-ci, il a notamment prédit que l'interface utilisateur – écran allait disparaître. C'est à ce moment-là que j'ai imaginé un robot qui pouvait se contrôler avec la voix, ce qui résout la problématique de la difficulté pour utiliser les nouvelles technologies actuelles.

La seconde conférence donnée par E. Auberix nous a montré les différentes solutions qu'il existait pour rendre un robot intelligent à faible coût.

Le sujet de mon TFE était donc tout écrit ; un robot contrôlable par la voix et intelligent qui vient en aide aux personnes âgées et également aux personnes à mobilité réduite.

2. INTÉRÊT

Un robot assistant n'est certainement pas la première chose à laquelle on pense lorsqu'on veut apporter de l'aide à une personne âgée ou à mobilité réduite.

En réalisant ce projet, je cherche à remplacer certaines tâches difficiles pour ces personnes. Par exemple, pour gérer un calendrier si elle n'a pas de smartphone, une personne doit se lever, aller jusqu'au calendrier en

question, le lire et noter des choses dessus. Cette tâche demande un effort et pour certaines personnes un risque lors du déplacement.

De plus, avec la mise en place de commandes vocales pour contrôler le robot évitent l'impact négatif des nouvelles technologies que celles-ci apportent en demandant une connaissance approfondie en informatique.

En ce qui concerne le robot, il peut facilement remplacer un animal de compagnie. Celui-ci apporte des avantages, mais aussi des désavantages qui poussent certaines personnes à ne pas vouloir en prendre un. Mon robot demande de l'interaction et possède une personnalité. Il prend une réelle place dans vie de tous les jours et apporte donc une vie en plus dans la maison.

Ce projet a donc un réel intérêt vis-à-vis de ces personnes.

3. CONTEXTE

A. LE CLIENT

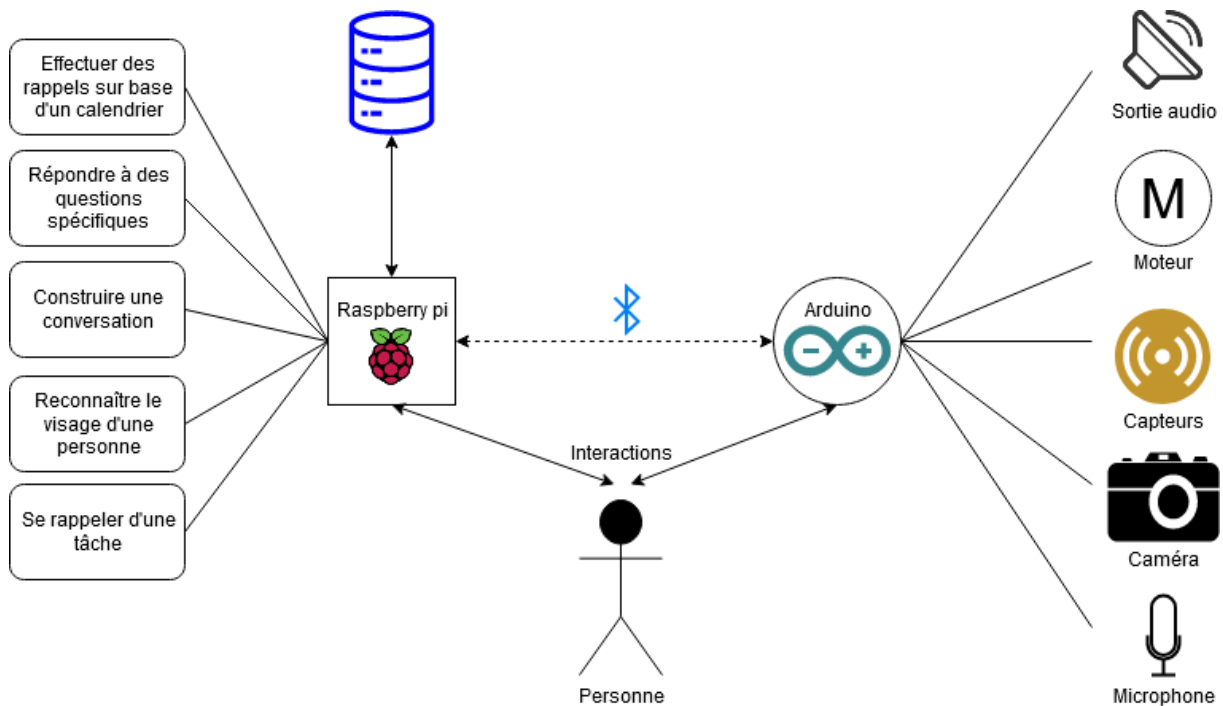
Voulant proposer un nouveau produit et non répondre à une problématique venant d'un client, je n'ai pas eu « la chance » d'en avoir un.

J'ai donc moi-même joué le rôle du client. J'ai commencé par noter sur un document les différentes tâches que mon robot devait accomplir. J'ai également pensé aux différentes contraintes auxquelles celui-ci devait répondre.

En tant que client, je me suis alors posé plusieurs questions :

- Quelles fonctionnalités logicielles pourraient aider une personne âgée ou à mobilité réduite ?
- À quoi doit ressembler le robot ?
- Quel est le prix du robot ?

Après avoir réfléchi à ces questions, j'ai réalisé un schéma regroupant les différentes fonctionnalités du robot



Ce schéma a évolué au cours du développement.

Tout au long de la réalisation de ce projet, j'ai pu constater les avantages et désavantages à ne pas avoir de client.

Au début, j'avais une image assez floue du résultat attendu et je changeais souvent d'avis quant aux fonctionnalités présentes dans le logiciel et dans le robot.

J'ai également dû prendre en charge le financement du robot, donc j'étais limité aux niveaux des composants du robot. J'ai dû renoncer à mettre en place certaines fonctionnalités à cause de mauvais composants.

L'avantage de ne pas avoir de client est que j'avais une grande liberté pour ajouter des fonctionnalités souhaitées. J'ai également pu personnaliser le robot au fur et à mesure du développement de celui-ci.

B. LES PRODUITS EXISTANTS

Bien que mon projet soit innovant, il existe certains produits existants qui ressemble à mon projet.

Cependant, la plupart ne sont pas destinés aux personnes âgées, ce qui fait la différence avec mon projet.

1. VECTOR

Vector¹ est un robot intelligent qui accompagne les personnes durant la journée. Il possède de nombreuses fonctionnalités.

Il peut également jouer avec son utilisateur comme au blackjack.

¹ <https://www.youtube.com/watch?v=4INH3w2rzTg>

D'après la marque, il est entre robot jouet, robot de service, enceinte connectée et contrôleur domotique.

Étant vraiment impressionné par ce robot, j'ai essayé de m'en inspirer au niveau du divertissement que celui-ci peut apporter.

2. GOOGLE ASSISTANT

Il n'est plus vraiment nécessaire de présenter Google Assistant.

Aujourd'hui présent dans tous les téléphones mobiles Android, il assiste son utilisateur en lui proposant des fonctionnalités presque infinies.

J'ai essayé de me démarquer de ce produit, au niveau du logiciel, en ajoutant une personnalité et une humeur à mon robot. L'assistant Google ne fera que répondre aux différentes requêtes. Tandis que mon robot, quant à lui, demandera qu'on interagisse avec lui, sans quoi, son humeur se verrait désagréé.

4. ANALYSE DE LA PROBLÉMATIQUE

A. IDENTIFICATION DES BESOINS TECHNIQUES

Pour réaliser ce projet, il faut créer un robot qui réagit à des interactions vocales et physiques et qui en ressort une action. L'utilisateur doit pouvoir contrôler le robot avec sa voix.

Il doit pouvoir se déplacer de manière autonome.

Contraintes :

- Alimentation par batteries
- Prix raisonnable
- Aucun fil visible
- Séparation du logiciel et de la partie électronique

B. LE CHOIX DES TECHNOLOGIES

1. SUPPORT ÉLECTRONIQUE ET LOGICIEL :

Plusieurs solutions étaient disponibles. Mon choix s'est vite tourné vers l'Arduino pour la partie électronique. L'Arduino a l'avantage d'être programmé facilement. Il se base sur du C++ et possède également de nombreuses bibliothèques créées par sa communauté pour faciliter l'utilisation de composants.

Il possède aussi beaucoup de documentations et de tutoriels sur Internet.

L'Arduino Mega comporte 54 pins programmables dont 16 entrées analogiques, 4 UARTs (hardware serial ports), un quartz à 16 MHz, une connexion USB, une prise jack pour l'alimentation, un en-tête ICSP et un bouton reset.

Cependant, créer un assistant vocal sur un Arduino serait bien trop compliqué étant donné la faible puissance de celui-ci. Il a donc fallu trouver une solution où le logiciel peut tourner et envoyer des commandes pour la partie électronique. Un Raspberry Pi est une solution idéale. Il est petit, puissant et fonctionne comme un ordinateur. Seulement, sa consommation est très élevée. Il n'est donc pas possible de directement l'intégrer au robot.

Le système embarqué du robot sera donc géré par un Arduino. J'ai déjà pu réaliser de nombreux projets avec celui-ci et connais bien ses points forts et ses faiblesses.

Le choix d'avoir pris un Raspberry Pi et non un serveur est que le prix de celui-ci est fixe contrairement à un serveur où il faudrait payer un abonnement et que le prix du Raspberry Pi serait vite atteint.

2. COMMUNICATION ÉLECTRONIQUE – LOGICIEL :

La communication entre ces deux technologies se fera à l'aide du Bluetooth.

Pour cette partie, j'avais le choix entre l'utilisation du Wifi ou bien l'utilisation du Bluetooth.

Comparaison² non exhaustive entre ces 2 technologies :

	Bluetooth	Wifi
Prix	Bas	Élevé
Bande Passante	+/- 800 Kbps	+/- 11 Mbps
Configuration matérielle requise	Adaptateur Bluetooth sur tous les supports qui communiquent	Adaptateur Wifi sur tous les supports qui communiquent, un routeur Wifi et un point d'accès Wifi
Portée	5 à 30 mètres	32 à 95 mètres
Consommation énergétique	Basse	Élevée

Le Bluetooth étant déjà intégré dans le Raspberry Pi, il sera facile de l'utiliser au niveau de la partie logicielle. Une bibliothèque nommée « Bluetooth » est disponible en python et permet de l'utiliser facilement dans le programme.

Du côté de l'électronique, il suffit d'utiliser un adaptateur Bluetooth pour pouvoir utiliser cette méthode de communication. Il existe plusieurs modèles. Pour mon prototype, il faut un module qui possède une communication bidirectionnelle.

Je vais donc utiliser le HC-05 communiquant en RX-TX avec l'Arduino.

Il peut être utilisé en mode esclave ou bien en mode maître.

² https://www.diffen.com/difference/Bluetooth_vs_Wifi

L'Arduino utilise la bibliothèque « SoftwareSerial.h » qui va permettre d'utiliser le Bluetooth dans celui-ci.

L'échange d'information se fera à l'aide de mots clés prédéfinis. Cela va permettre de faciliter la rédaction du code dans les deux parties.

3. LES COMPOSANTS

A. DHT11

³Capteur de température et humidité pour mesurer la température actuelle dans une pièce.

Il est capable prendre des mesures de 0 à +50°C et possède une précision de 2°C. Quant à l'humidité, il peut mesurer de 20 à 80% et possède une précision de 5%.

Il existe également le DHT12. Celui-ci est un peu plus précis dans les mesures. Il est cependant plus cher et est moins documenté sur internet. Il n'est non plus pas nécessaire d'avoir une mesure si précise dans le cadre de ce projet.

Le DHT11 est donc plus adéquat pour ce projet. Il convient parfaitement à ce projet. Il est à la fois facile à mettre en place, très peu coûteux et relativement précis.

B. HC-SR04

⁴C'est un capteur à ultrason « low cost ». Ce capteur fonctionne avec une tension d'alimentation de 5 volts, dispose d'un angle de mesure de 15° environ et permet de faire des mesures de distance entre 2 centimètres et 4 mètres avec une précision de 3mm.

Pour son prix, il est relativement efficace. Le possédant déjà chez moi, je l'ai utilisé dans mon projet sans hésitation.

C. HC-SR501

C'est un capteur de mouvements se basant sur les rayonnements infrarouges. D'après la datasheet, il peut détecter un mouvement dans un angle inférieur à 120° et jusqu'à 7 mètres (durant le développement je remarquerai que c'est n'est pas du tout le cas).

Ce capteur n'étant pas cher et avec des données correctes pour son prix, je l'ai pris pour mon projet.

D. MOTEURS DC KKMOM

³ <https://www.carnetdumaker.net/articles/utiliser-un-capteur-de-temperature-et-dhumidite-dht11-dht22-avec-une-carte-arduino-genuino/>

⁴ <https://www.carnetdumaker.net/articles/mesurer-une-distance-avec-un-capteur-ultrason-hc-sr04-et-une-carte-arduino-genuino/>

J'ai pas mal hésité lors de l'achat des moteurs pour faire déplacer mon robot. Ayant déjà travaillé avec des moteurs DC à bas prix venant d'Amazon, je n'étais pas du tout satisfait du résultat. Mais ceux-ci ne coûtaient vraiment pas cher comparé à d'autres moteurs.

C'est à contre cœur que je les ai repris pour ce projet, mais pour une question d'argent, je n'ai pas eu d'autres choix.

Cependant, ceux-ci ont l'avantage d'avoir des roues fournies avec.

E. L298N

Pour contrôler ces moteurs DC, il faut un driver à double pont H. En effet, les moteurs fonctionnant en courant continu, il n'est pas possible de changer le sens du courant directement depuis l'Arduino. I

Ce composant ne coûte pas cher et fonctionne correctement.

C'est également un des seuls vendu sur la plateforme Amazon.

F. SG90

Ces Servo Moteurs conviennent parfaitement au projet. Ils coûtent moins de 3€ pièces et ont une vitesse de fonctionnement de 0.1s / 60°.

Ils sont également simples à mettre en place.

4. ESTHÉTIQUE

Contrairement à ce qu'on pourrait le penser, le design d'un robot est très important. De nos jours, encore de nombreuses personnes ont peur des robots, surtout quand ceux-ci ressemblent à des humains ; les humanoïdes.

Lorsqu'on regarde le robot Sophia, il est évident que celle-ci nous fasse peur.



Cette peur se nourrit de l'idée que les robots pourraient devenir plus intelligent qu'un humain et prendre le contrôle du monde. Bien évidemment, ce n'est que de la science-fiction.

Pour qu'un robot puisse gagner la confiance d'un humain, le design est donc très important.

Une étude⁶ menée par des chercheurs de l'Université de Washington à Seattle concernant cette problématique a démontré que le design préféré par les humains était épuré avec des émotions plus proches des animaux de compagnies que des hommes.

Lors de la conception du projet, j'ai imaginé un design qui était à la fois simpliste, mais également ressemblant à un robot déjà existant, WALL.E.



Ce design n'est en fait qu'un cube avec une tête, deux bras et des roues pour se déplacer. Il est facile à réaliser et convient parfaitement à mon projet.

5. SITE WEB

⁵ [https://fr.wikipedia.org/wiki/Sophia_\(robot\)](https://fr.wikipedia.org/wiki/Sophia_(robot))

⁶ <https://www.ladn.eu/nouveaux-usages/etude-marketing/un-robot-a-t-il-besoin-de-sourcils-pour-se-faire-aimer-dun-etre-humain/>

⁷

https://vignette.wikia.nocookie.net/pixar/images/d/de/Wall%E2%80%A2e_clipped_rev_1.png/revision/latest?cb=20170807223723

La plateforme qui accueille les données de télémetries est réalisé en Node.js.

J'ai utilisé cet outil car j'ai récemment développé une plateforme web pour un projet utilisant la même technologie. Je n'ai donc pas perdu de temps à apprendre une nouvelle technologie.

Cet outil permet de créer des API rapidement, ce qui convient parfaitement à mon utilisation.

6. DÉVELOPPEMENT DU PROJET

Pour commencer le développement, j'ai dû mettre en place toutes les ressources nécessaires.

J'ai commencé par configurer le Raspberry Pi pour que celui-ci puisse exécuter du Python.

Pour pouvoir développer le logiciel plus facilement, je me suis connecté en SSH au Raspberry Pi. Je pouvais donc entièrement le contrôler à distance.

Une fois toutes les ressources installées, j'ai pu commencer à développer le logiciel.

A. MÉTHODOLOGIE

Durant le développement du projet, les demandes initiales ont pas mal changé à cause de divers facteurs, notamment le Covid-19 et certains composants qui ne fonctionnaient pas comme espérés.

La méthodologie utilisée est donc agile.

Étant donné que je n'avais pas de client, je n'ai pas pu mettre en place une méthodologie précise.

J'ai cependant suivi un fil rouge tout au long du développement. Comme prévu initialement, j'ai d'abord conçu le squelette du logiciel, comprenant la reconnaissance de voix, la synthèse vocale et la connexion au robot en Bluetooth. Ce squelette permet de rendre le projet fonctionnel.

Une fois le squelette prêt, j'ai développé les fonctionnalités les unes après les autres pour toujours avoir un produit qui fonctionne.

B. LOGICIEL

Avant d'expliquer le développement du logiciel, je vais résumer le fonctionnement du logiciel.

Le robot possède une humeur. Celle-ci change en fonction du nombre d'interactions avec le robot. Au plus l'utilisateur utilisera le robot, au plus son humeur sera meilleure. Celle-ci diminue avec le temps.

Lorsqu'un utilisateur fera une requête au robot, une intelligence artificielle analysera la requête pour déterminer la catégorie de celle-ci. Le robot pourra répondre adéquatement à la requête. En fonction de l'humeur du robot, la réponse sera différente.

Liste exhaustive des fonctionnalités du logiciel :

- Envoyer un email

- Signaler un problème au développeur
- Lire les évènements à venir du calendrier
- Ajouter un évènement
- Mettre un minuteur
- Mettre une alarme
- Lire les actualités de la région
- Donner la météo
- Raconter une blague

Fonctionnalités dépendantes du robot :

- Donner la température de la pièce
- Réagir à un mouvement
- Se déplacer dans une pièce

Comme dit dans la partie sur la méthodologie, j'ai commencé le développement par le squelette du projet.

Pour la reconnaissance de voix, j'utilise une API⁸ fournie par Google. En une dizaine de ligne, on peut détecter la voix et la transformer en texte.

Pour la synthèse vocale, j'étais également parti avec une API⁹ de Google.

Cependant, la voix proposée par Google ne correspondait pas avec la personnalité du robot. C'était une voix de femme, mais qui n'était pas du tout naturelle.

J'ai donc changé cette API par un site¹⁰ en ligne qui propose une multitude de voix de voix et dans de nombreuses langues. Cet outil offre également la possibilité de modifier le pitch de la voix, ce qui correspond parfaitement à la personnalité de mon robot.

Pour la connexion Bluetooth avec le robot, il suffit simplement d'utiliser la librairie Bluetooth de Python.

Une fois ce squelette mis en place, j'ai développé les différentes fonctionnalités.

La plus grosse difficulté était de décortiquer la requête et en ressortir les différents éléments nécessaires.

Il fallait également prendre en compte toutes les variations possibles pour une seule et même requête. Lorsqu'une requête n'entre pas dans les différents cas possibles, une erreur est signalée à l'utilisateur.

⁸ <https://cloud.google.com/speech-to-text/docs/libraries>

⁹ <https://cloud.google.com/text-to-speech/docs/libraries>

¹⁰ <https://ttsdemo.com/>

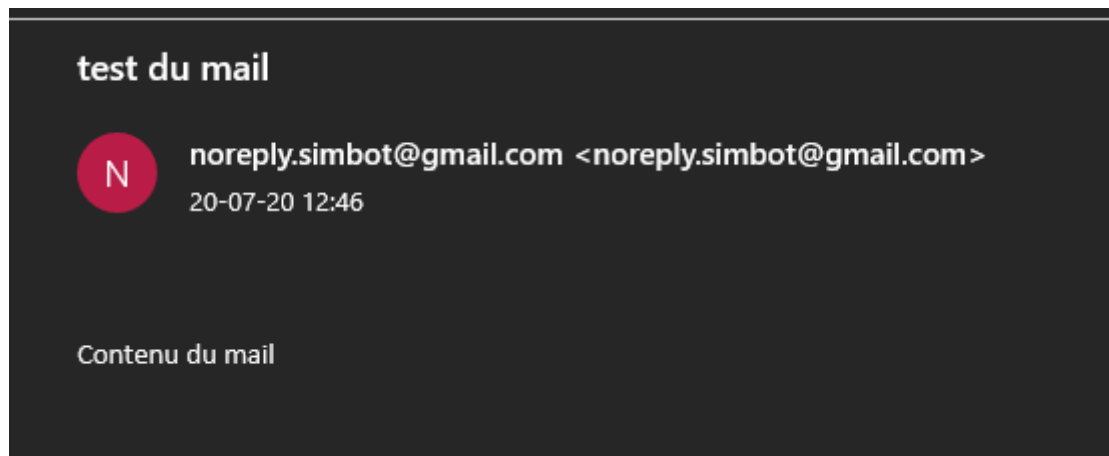
1. Envoyer un mail

- L'utilisateur doit pouvoir envoyer un mail en précisant le destinataire et le contenu du mail.

Cette fonctionnalité utilise la librairie « smtplib » incluse dans Python. Elle se connecte à un serveur SMTP. J'ai utilisé le serveur de Gmail et je l'ai donc relié à un compte Gmail créé pour l'occasion.

Lorsqu'un utilisateur veut envoyer un email, le robot lui demande à qui il veut envoyer le mail. L'utilisateur répond par un nom. Le programme va alors chercher dans un fichier de configuration l'adresse email reliée au nom. Ensuite le robot demande à l'utilisateur le corps du mail.

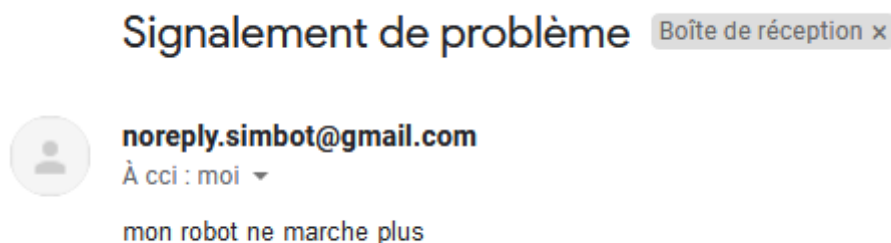
Une fois ces étapes remplies, le mail est envoyé au destinataire. L'adresse de l'expéditeur est noreply.simbot@gmail.com et l'objet est « Mail de [nom de l'utilisateur] ». Il n'est en revanche pas possible de recevoir un mail pour l'instant.



2. Signaler un problème au développeur

- L'utilisateur doit pouvoir signaler aux développeurs un problème rencontré.

Cette fonctionnalité utilise l'envoi d'un mail. Aucun destinataire ne sera demandé et lors de l'envoi du mail, celui-ci sera envoyé à l'adresse noreply.simbot@gmail.com avec le problème rencontré.



3. Lire les événements à venir du calendrier

- L'utilisateur doit recevoir les événements dans les quelques jours à venir de son calendrier.

Pour cette fonctionnalité, j'utilise l'API de Google par facilité de développement mais également par facilité pour l'utilisateur. S'il utilise déjà un calendrier personnel sur Google, celui-ci et celui du robot seront synchronisés et il pourra donc en profiter sur d'autres plateformes.

9	10	11	12	13	14	15
						L'Assomption • 12:30pm sortir les poubelles
16	17	18	19	20	21	22
• 6:30pm Aller courir						

```
2020-08-15T12:30:00+02:00 sortir les poubelles
sortir les poubelles
mois : 08
Jour : 15
sortir les poubelles le 15 août
```

```
2020-08-16T18:30:00+02:00 Aller courir
Aller courir
mois : 08
Jour : 16
Aller courir le 16 août
```

4. Ajouter un évènement

- L'utilisateur doit pouvoir ajouter un évènement dans son calendrier en précisant le nom de l'évènement, une description, une localisation, une date et heure de début et une heure de fin.

Cette fonctionnalité se base également sur le calendrier Google. Lorsqu'un utilisateur veut ajouter un évènement, une série de question lui sera demandée afin d'avoir tous les éléments nécessaires pour envoyer l'évènement vers l'API.

```
Evènements : nager-Perwez-/2020-8-24T15:45:00+02:00-2020-8-24T17:30:00+02:00
Event created: https://www.google.com/calendar/event?eid=bmkzYjRlZWJjZjYxcTQ3MmFyc3I2Y2QwNzQgZ3VpbGxhdW1ldmRoMzEwNS5ndkBT
['calendar/addEvent.py', 'nager', 'Perwez', '/', '2020-8-24T15:45:00+02:00', '2020-8-24T17:30:00+02:00']
```



5. Mettre une alarme ou un minuteur

- *L'utilisateur doit pouvoir lancer un minuteur en précisant le temps voulu.*

- *L'utilisateur doit pouvoir mettre une alarme à une heure précise.*

Pour mettre en place cette fonctionnalité, j'ai dû utiliser la programmation parallèle avec le threading. L'alarme ou le minuteur doivent tourner en parallèle du programme principale pour éviter de le rendre inutilisable.

En Python, il est relativement simple d'utiliser les Threads. Il suffit de définir une fonction qui sera exécutée au lancement d'un Thread et puis de le démarrer.

Pour recevoir le temps du minuteur lors de la requête, la phrase est décomposée et des mots clés sont recherchés dedans, par exemple « minute ». En fonction des mots clé présents, la longueur du minuteur est modifiée.

En ce qui concerne l'alarme, l'heure précisée dans la requête est transformée en format Timestamp. Le Thread vérifiera alors le Timestamp actuel et lorsque les 2 seront identiques.

Dans ces 2 fonctionnalités, une alarme sera émise à la fin.

6. Lire les actualités de la région

- *L'utilisateur doit pouvoir recevoir les actualités de sa région.*

Cette fonctionnalité utilise une API¹¹ qui fournit les gros titres provenant de différentes sources.

¹¹ <https://newsapi.org/>

Dans la requête, on peut choisir le pays où trouver les articles. On reçoit alors une collection contenant les articles et les sources.

Afin d'éviter les répétitions lorsque le robot dit les articles, je ne que les articles venant d'une seule et même source.

Cette source par défaut est « RTBF », mais peut être changée dans le fichier de configuration.

```
Voici les actualités
actualité
Direct / Coronavirus en Belgique : suivez à 11h la conférence de presse du Centre de crise - RTBF
La cancel culture : quand le harceleur devient victime - RTBF
Remco Evenepoel : le plus précoce depuis Giuseppe Saronni - RTBF
```

7. Donner la météo

- *L'utilisateur doit pouvoir recevoir la météo actuelle proche de chez lui*

Cette fonctionnalité utilise une API¹² qui fournit des données météorologiques sur une localisation.

La location utilisée ici est Bruxelles.

Cette API fournit une très grande quantité de données, mais je ne retiens que la température et un alias pour décrire la météo actuelle.

```
météo
Actuellement à Bruxelles , il fait 24 degré et un temps Eclaircies
```

8. Raconter une blague

- *L'utilisateur doit pouvoir recevoir une blague de la part du robot*

Afin d'apporter un peu un peu plus de personnalité au robot, celui-ci peut raconter une blague.

Cette fonctionnalité se base également sur une API¹³ fournissant des blagues.

Le robot racontera une blague aléatoire parmi les réponses fournies.

```
Voici une blague
blague
Quel est l'arbre préféré des chômeurs ?
Le bouleau
```

¹² <https://prevision-meteo.ch/services>

¹³ <https://bridge.buddyweb.fr/api/blagues/blagues/>

9. Donner la température de la pièce

- L'utilisateur doit pouvoir recevoir la température qu'il fait actuellement dans la pièce

Cette fonctionnalité interagit directement avec la partie électronique du robot. Elle envoie via le Bluetooth une requête au robot pour demander la température et attend. Le robot lit la valeur du capteur et envoie cette donnée via le Bluetooth.

Lorsque le logiciel reçoit la donnée, il l'intègre dans sa réponse pour l'utilisateur.

10. réagir à un mouvement

- Le robot doit pouvoir réagir à un mouvement dans la pièce.

Lorsque le robot n'est plus utilisé pendant un certain moment, il passe en mode « veille ». À partir de ce moment-là, l'Arduino va lire le capteur de mouvement. Si celui-ci détecte un mouvement, le robot réagira et enverra au logiciel une alerte de mouvement.

11. Se déplacer dans une pièce

- Le robot doit pouvoir se déplacer de manière autonome dans une pièce.

À la suite d'une requête ou bien lorsque le robot est de bonne humeur, il peut se déplacer dans une pièce.

L'Arduino envoie l'information vers le L298N pour mettre en marche les moteurs DC. Lorsqu'il avance, l'Arduino lit le capteur de proximité pour que le robot puisse éviter un obstacle.

Si un obstacle est détecté, le capteur va regarder à droite puis à gauche pour voir quel chemin est libre.

Le robot doit s'arrêter de bouger lorsque l'utilisateur l'ordonne ou lorsque l'algorithme lié à l'humeur le lui ordonne.

12. Intelligence artificielle

Au début du développement du logiciel, pour faire une requête, l'utilisateur devait dire la phrase mot pour mot pour être compris par le logiciel. Après réflexion, ce n'était pas du tout envisageable pour les personnes âgées.

J'ai alors cherché une alternative. En parcourant divers tutoriels Python sur YouTube, je suis tombé sur une vidéo¹⁴ de quelqu'un qui développe un chatbot basé sur une intelligence artificielle. Ce chatbot avait pour avantage d'être basé sur un dataset (fichier de données sur lequel se construit l'IA) personnel et en plus détecter la catégorie de la requête. Par exemple, si un utilisateur veut avoir la météo, il pourra dire « donne-moi la météo » ou bien « Quelle est la météo actuellement ? ». L'intelligence artificielle détectera que ces phrases parlent toutes les 2 de la météo.

¹⁴ <https://www.youtube.com/watch?v=wypVcNIH6D4>

Le fonctionnement de cette IA est à la fois très simple mais en même temps très complexe. Je ne rentrerai pas dans les détails pour son explication.

Lorsqu'un utilisateur entre une commande. Par exemple « donne-moi la météo », elle va décomposer cette phrase et mettre une position à chaque mot. Dans cette phrase, le mot « météo » est à la 3^{ème} position. L'IA cherchera alors dans son réseau neuronal quelle sujet de phrase possède le mot « météo » à la 3^{ème} position.

Cette explication est bien évidemment très vulgarisée par rapport à son fonctionnement réel.

Grâce au dataset personnel, j'ai pu créer les requêtes que l'IA pourrait comprendre et mettre des réponses adéquates.

Ajouter cette intelligence artificielle dans mon logiciel un énorme avantage. Elle m'a permis de gagner beaucoup de temps dans le développement de l'interaction entre l'utilisateur et le robot.

13. L'humeur du robot

Pour ajouter une personnalité à mon robot, je lui ai implémenté une humeur. Elle demandera à l'utilisateur d'interagir avec le robot, sans quoi, elle se dégraderait.

J'ai donc mis en place un Thread qui diminue l'humeur au cours du temps. Celle-ci est initiée à 1000. Lorsqu'un utilisateur effectue une requête, elle est augmentée. Au plus l'utilisateur fera de requêtes, au plus l'humeur du robot sera bonne.

Cette humeur reste également dans la mémoire du robot. Lorsque le robot est redémarré, elle reprendra au même stade que lorsqu'il s'est éteint.

Afin de la mettre en évidence, les réponses aux requêtes seront impactées. L'utilisateur pourra alors percevoir l'humeur du robot.

Au niveau de la partie électronique, la couleur des yeux du robot changera. Son « énergie » corporelle sera également impactée.

C. ÉLECTRONIQUE

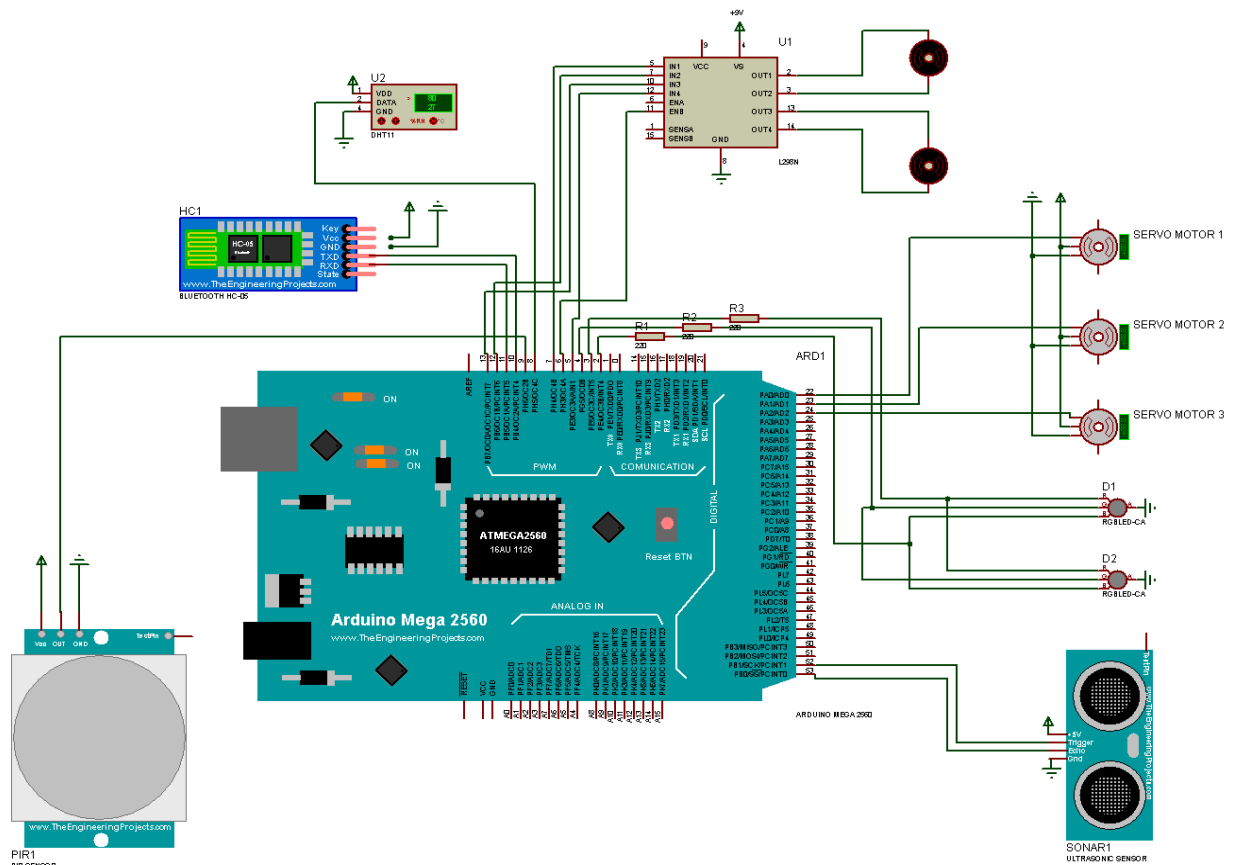
La partie électronique est relativement simple en dehors de la conception du robot. Tous les composants du robot à l'exception des 2 moteurs DC sont reliés à l'Arduino. Ceux-ci sont donc directement gérés par le code source.

Afin de faciliter la gestion des tâches lors de l'exécution du programme, le code n'est en fait qu'une série de fonctions qui sont appelées par le Bluetooth.

C'est le logiciel présent sur le Raspberry Pi qui gère les actions du robot.

Pour certaines fonctionnalités, des données provenant de capteurs sont envoyées vers le logiciel.

1. SCHÉMA ÉLECTRONIQUE



D. SITE INTERNET

Afin d'avoir un aperçu de l'utilisation du robot, la télémétrie est affichée sur une plateforme web.

Elle affiche le nombre de requêtes réalisés chaque jour.

Elle permet également d'afficher en JSON les requêtes et le nombre de robots fonctionnels.

Un graphique présente la quantité de chaque type de requêtes au cours d'une semaine.

Ces données me serviront à savoir quel type de requêtes sont le plus et le moins réalisées dans le but de savoir lesquelles il faut améliorer.

Pour accéder aux données, il faut se connecter à la plateforme.

Cette plateforme est basée sur un template¹⁵.

E. BASE DE DONNÉES

Pour stocker les données venant du logiciel, j'utilise une base de données. Elle stocke chaque requête avec son type et son horodatage.

Je stock également chaque robot et leur ID afin de compter le nombre de robots en état de fonctionnement.

Aucune donnée personnelle de l'utilisateur n'est stockée.

J'ai également une base de données qui s'occupe de stocker les utilisateurs de la plateforme de télémétrie. Il n'est pas possible de s'enregistrer sur la plateforme car seul un administrateur peut accéder aux données.

F. VPS

Pour héberger le site web et la base de données, j'utilise un VPS loué chez OVH¹⁶.

Le service utilisé pour la base de données est MySQL.

7. ANALYSE DE SÉCURITÉ

A. ÉLECTRONIQUE

L'Arduino a l'avantage de posséder ses propres protections.

Il possède notamment un régulateur de tension DC-DC (D24V6F5) qui régule la tension à 5V et se coupe en cas de surchauffe ou de court-circuit.

Il est également équipé d'un fusible ré-armable automatique qui protège le port USB des courts-circuits¹⁷.

Les LED RGB des yeux sont protégées par des résistances de 220 Ω .

Lors de la conception, un des composants (ESP32-CAM) chauffait anormalement très fort. Étant donné que la structure du robot est en carton, il a été décidé de retirer ce composant pour éviter tous risques.

B. SITE WEB

- Connexion à la plateforme

Pour accéder aux données télémétriques du robot, il faut s'authentifier à la plateforme. Ces données ne doivent être que visibles par un administrateur. Il n'est donc pas possible de s'inscrire sur la plateforme.

¹⁵ <https://codepen.io/JFarrow/pen/fFrpg>

¹⁶ <https://www.ovh.com/>

¹⁷ <https://www.locoduino.org/spip.php?article69>

- HTTPS

La communication entre le serveur et le site web est chiffrée à l'aide de l'HTTPS. Étant donné qu'il faut se connecter à la plateforme et qu'il y a donc un échange de mot de passe, celui-ci doit-être envoyé au serveur de manière confidentielle.

- Certification du nom de domaine

Un certificat a été mis en place avec Let's Encrypt à l'aide de Certbot.

C. VPS

- Firewall

Un firewall est présent sur le VPS afin de gérer les ports ouverts vers l'extérieur.

```
Starting Nmap 7.70 ( https://nmap.org ) at 2020-08-08 09:13 UTC
Nmap scan report for vps-abd9d948.vps.ovh.net (51.77.201.156)
Host is up (0.083s latency).

PORT      STATE      SERVICE
21/tcp    filtered  ftp
22/tcp    open      ssh
23/tcp    filtered  telnet
80/tcp    open      http
110/tcp   filtered  pop3
143/tcp   filtered  imap
443/tcp   open      https
3389/tcp  closed    ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds
```

18

- SSH

La communication avec le serveur est chiffrée à l'aide du protocole SSH.

- Fail2ban

Nous avons mis en place Fail2Ban pour empêcher les attaques à répétitions sur le serveur. Les règles mises en place concernent les ports ouverts, soit le port 22 (SSH) ici.

- Authentification

¹⁸ <https://hackertarget.com/nmap-online-port-scanner/>

L'authentification au serveur ne peut se faire qu'avec un fichier de clé et une passe-phrase. Nous avons également désactivé la connexion par mot de passe et la connexion avec l'utilisateur root.

D. SERVICES

- Base de données

La base de données se trouvant sur le serveur, celle-ci est protégée avec la sécurité du serveur. Les mots de passes sont cryptés dans la base de données.

8. PROBLÈMES RENCONTRÉS

A. RASPBERRY PI

Au début du développement de ce projet, je n'étais pas certain de quel OS utiliser pour supporter la partie logicielle. J'en ai testé différents et j'étais parti vers Ubuntu. Au cours du développement, j'ai eu des nombreux problèmes avec celui-ci. Certaines libraires Python ne fonctionnaient pas et j'ai dû faire des manipulations. Certaines d'entre elles ont fait complètement crasher le Raspberry Pi. J'ai donc du tout réinstallé et je suis finalement parti sur Raspbian, qui est l'OS installé par défaut.

B. ESP32-CAM

Comme précisé dans la partie sécurité, j'ai rencontré pas mal de problèmes avec ce composant. Tout d'abords, la qualité de celui-ci est tellement mauvaise qu'il n'a pas pu être utilisé dans le projet.

Malgré un essai de mise au point de l'objectif, il n'était pas possible de travailler avec.

De plus, ce composant chauffait anormalement très fort.

J'ai donc décidé de ne pas utiliser ce composant.

C. SORTIE AUDIO INTÉGRÉE AU ROBOT

À l'origine du projet, il était voulu que la sortie audio du logiciel soit intégrée au robot. J'ai pensé à mettre un baffle Bluetooth étant donné que le Raspberry Pi pouvait gérer plusieurs périphériques Bluetooth. Cependant, lorsqu'un son était joué, celui-ci était saccadé et donc inaudible. Je pense que cela était dû aux faibles performances du Raspberry Pi.

La solution alternative était donc d'utiliser un baffle relié par câble au Raspberry Pi.

D. CAPTEUR DE MOUVEMENT

D'après la datasheet de ce capteur, il pouvait percevoir un mouvement jusqu'à 7m. Après de nombreux essais et manipulations des potentiomètres pour changer la sensibilité de détection, je n'ai pas réussi à capter un mouvement au-delà de 30cm du capteur.

E. MOTEURS DC

Étant déjà sceptique lors de l'achat de ces moteurs, je ne m'attendais pas à avoir autant de problèmes avec ceux-ci.

Pour commencer, un des moteurs provoquait des courts-circuits lorsqu'il était mis sous tension. Il m'a cependant fallu un moment avant de m'apercevoir que le problème venait de ce moteur. J'ai donc dû prendre un moteur venant d'un ancien projet personnel.

Les deux moteurs n'étant pas de la même fabrication, ceux-ci ne tournent pas à la même vitesse.

Heureusement, grâce au composant L298N, il est possible de gérer la vitesse des moteurs.

9. PISTES D'AMÉLIORATION

Je pense que je trouverai toujours des améliorations à faire pour ce projet. Durant le développement, j'ai dû me fixer des limites sans quoi je n'aurais pas eu assez de temps pour tout faire.

A. IMPRESSION 3D

Réaliser un robot imprimé en 3D serait réellement intéressant pour ce projet. La structure actuelle, en carton, est relativement fragile. Si une personne âgée veut déplacer le robot, elle risque de l'abîmer.

B. MEILLEURS MOTEURS DC

Avec de meilleurs moteurs DC, il aurait été possible d'apporter des fonctionnalités intéressantes. Notamment une cartographie 2D de la pièce. Le robot se balade et lorsqu'il rencontre un obstacle, il met un point sur la coordonnée correspondante de la matrice.

Au fur et à mesure qu'il se déplace, il va améliorer cette carte 2D de la pièce. Il pourra donc à la fin se balader librement sans devoir éviter les obstacles.

C. FONCTIONNALITÉS SUPPLÉMENTAIRES

- GPS

Le robot pourrait se déplacer de pièce en pièce après une configuration. L'utilisateur pourrait par exemple lui demander de le suivre jusqu'à la cuisine pour l'utiliser dans cette pièce. Cette fonctionnalité augmenterait le côté autonome du robot. Mais il permettrait également d'augmenter l'utilisation du robot.

- Reconnaissance faciale et vocale

Initialement, une reconnaissance faciale devait être mise en place dans ce projet. Mais à cause de l'incompatibilité de la caméra, ce n'était plus possible dans le temps imparti.

Cette fonctionnalité aurait pu augmenter la personnalité du robot. Lorsqu'une personne interagit avec, celui-ci pourrait donner des réponses personnalisées. Certaines fonctionnalités pourraient également changer en fonction de qui utilise le robot.

D. CONTRÔLE DOMOTIQUE

Pour apporter encore plus d'aide aux utilisateurs. Il est envisageable d'ajouter un contrôle des objets connectés de la maison, notamment les prises et interrupteurs connectés.

L'utilisateur pourra alors demander au robot d'allumer ou éteindre les lumières/appareils connectés.

E. RECHARGE AUTOMATIQUE

Actuellement, il faut recharger la pile présente dans le robot.

Ce n'est pas toujours évident certaines personnes.

À l'aide d'un module GPS, le robot pourrait alors aller jusqu'à une plateforme de recharge lorsque sa batterie est faible.

Il pourrait être rechargé sans fil à l'aide du composant EF03093. Celui apporte 12 Vcc/600 mA. Il faudrait également mettre une batterie 12V dans le robot.

Cette fonctionnalité rendrait le robot 100% autonome en matière d'énergie.

F. APPLICATION DE CONFIGURATION

Lorsqu'un utilisateur reçoit le robot, il aurait été intéressant d'avoir une application mobile pour configurer le robot. Cette configuration serait trop compliquée par commande vocale car cette fonctionnalité demande Internet pour fonctionner.

Cette application permettrait notamment de configurer la connexion Internet du Raspberry Pi et les différents comptes utilisés pour certaines fonctionnalités.

10. CONCLUSION

Ce TFE était un projet qui me tenait vraiment à cœur. J'ai pu réaliser des produits dans des domaines que j'apprécie fortement tout en ayant un objectif derrière qui me motivait tout au long du développement.

J'ai pu apprendre de nouvelles technologies comme le langage Python qui m'a d'ailleurs motivé à réaliser d'autres projets, la communication sans-fil vers un système électronique et bien d'autres encore.

Malgré le fait que je n'ai pas pu vivre une expérience avec un client, j'ai quand même su me mettre à sa place et définir clairement les besoins et les contraintes au début du projet. Cependant, au cours du développement, j'ai pu me permettre de modifier ou d'ajouter des fonctionnalités supplémentaires pour améliorer l'expérience utilisateur.

Tout au long du développement, j'ai dû faire face à des problèmes, tant au niveau du développement qu'au niveau électronique. J'ai pas mal de fois du tout recommencer à 0 au niveau du logiciel car certaines manipulations au niveau du Raspberry Pi pour faire fonctionner des librairies l'ont fait complètement crasher. J'ai aussi dû démonter le robot à plusieurs reprises car certains composants avaient cessé de fonctionner ou bien à cause d'un câble ou une soudure qui ont lâché.

Finalement, je suis réellement satisfait du résultat de ce projet. Il est entièrement fonctionnel. Même si le robot n'est qu'un prototype, il peut déjà être utilisé. Certains points sont encore à améliorer, mais ce n'est qu'une plus-value à l'état actuel du projet.

J'espère que mon projet pourra enfin montrer que les personnes âgées doivent également être prises en comptes lors de la réalisation de nouvelles technologies.

11. BIBLIOGRAPHIE