**PROJET IoT PERSONNEL**

**Led virtuelle contrôlée par commandes vocales ( Qt C++ / Firebase / Dialogflow / Google Assistant )**

*Ressources code : https://github.com/N-E-O-38/Virtual\_LED\_GG\_Assistant.git*

1. **Création d’un projet Firebase**

* Créer une base de données temps réelle pré-remplie
* Autoriser l’authentification avec mail et mot de passe
* Paramétrer la base de données afin qu’elle ne dialogue qu’avec des utilisateurs authentifiés
* Ajouter une application web afin d’obtenir une clé API

1. **Ecriture du code en C++ sur Qt**

* Permettre l’enregistrement d’utilisateurs et la connexion avec identifiants à une base de données temps réelle, pour effectuer des requêtes avec token, afin récupérer l’état de la led ainsi que sa couleur
* Mettre en place un timer afin d’effectuer à intervalles réguliers des requêtes vers la base de données tout en conservant le token en mémoire
* Parser et gérer les réponses de retour / erreurs, pour pouvoir transmettre les paramètres à une fenêtre Qt, qui affichera l’aspect de la led
* Pourquoi choisir Qt et C++ ? -> Réponses dans la dernière étape “9. Perspective d’évolution du projet”

1. **Test du bon fonctionnement de la modification de l’état et de la couleur de la led virtuelle par modifications des données dans la base de données**
2. **A l’aide de la boîte à outils firebase, initialisation et déploiement d’un projet NodeJs pour écrire une fonction permettant d’effectuer une requête vers la base, afin de pouvoir mettre à jour les données avec les paramètres disponibles dans la requête**

* Attention de bien veiller à ce que la version node spécifiée sous “engine” dans le package.json soit supportée par Dialogflow
* Attention également au fichier package.json qui contient par défaut une erreur (changer “lint”:”eslint .” par “lint”:”eslint” sinon le déploiement notifiera une erreur “unexpected token”

1. **Configuration d’un projet Google Cloud sur Dialogflow (avec le même compte Google utilisé pour Firebase) afin de pouvoir dialoguer avec l’assistant Google**

* Paramétrer des “entities”, qui attribuent des mots-clés aux clés spécifiées, utiles au learning de l’assistant vocal
* Définir un “intent” permettant d’enregistrer des phrases contenant les mots-clés précédemment spécifiés et quelques réponses par défaut

1. **Connecter le projet Dialogflow avec la base de données Firebase**

* Permettre à Dialogflow d’appeler un Webhook en réponse à une commande vocale dans “l’intent”
* Créer un Webhook dans la partie Fulfillment, avec comme URL l’adresse de la fonction créée sur Firebase, Dialogflow transmettra ainsi les paramètres modifiés par commandes vocales à la requête https de la fonction

1. **Finalisation des services composant l’architecture IoT**

* Se rendre dans l’onglet “Integrations” de Dialogflow afin d’accéder aux Actions Google, nommer son test et choisir une action de type conversation en référence à “l’intent” défini dans Dialogflow

1. **Test final**

* Veiller à ce que l’application Qt soit démarrée, et que la led change bien d’aspect lorsque les données en base sont modifiées à l’aide des commandes vocales indiquées dans la page d’Actions Google ou directement à travers le smartphone, en faisant au préalable appel au test, et en étant bien connecté avec le compte Google correspondant à celui des projets Firebase et Dialogflow

1. **Perspective d’évolution du projet**

* Intégrer l’application sur un système embarqué connecté et faire en sorte qu’elle se démarre au boot du système, sur un écran.