

Programme D-CLIC

Module : UML/Merise

Projet Fil Rouge - JARVIS partie 1

Durée de l'exercice : 2 semaines

Du 09/05/2022 au 21/05/2022

Heure limite du dépôt : 21/05/2022 à 20H

I - Introduction

Vous attaquez le vaste et complexe univers du backend et des bases de données. Pour commencer, vous allez découvrir UML/Merise. À ne pas confondre, UML est un langage de modélisation et Merise est une méthodologie. Tous deux permettent de traduire des attentes émises dans un cahier des charges, ou plus généralement directement d'un client, en base de données fonctionnelle.

Le projet JARVIS est un projet "fil rouge" que vous allez développer tout au long de votre parcours en Backend. Il est décomposé en plusieurs parties : UML/Merise, SQL/MySQL, Node JS. Le but de ce projet sera de créer un assistant pour la gestion domotique des appareils connectés d'une maison.

Il existe des étapes à respecter pour réaliser correctement la conception de ce type de projet.

II - Objectif

Le projet JARVIS partie 1 a pour objectif de vous préparer au mieux à interpréter les informations issues d'un cahier des charges. Par la suite, vous allez devoir traduire ces informations à partir de méthodologies, pour préparer la construction de votre base de données. Voici les éléments que vous devrez interpréter, analyser et concevoir :

- Diagramme de cas d'utilisation (à interpréter)
- Modèle Conceptuel de Données MCD (à faire)
- Diagramme de classe (à faire)

Cette étape initiale permet aux développeurs de prendre le recul nécessaire pour construire correctement la base de données et s'assurer que tous les acteurs ainsi que leurs fonctionnalités respectives sont bien représentés.

À travers ce document et ce projet, nous allons vous guider pour réaliser ces différentes étapes.

Remarque : C'est une étape essentielle pour la suite. Elle est souvent réalisée par des profils expérimentés, car une erreur ou un oubli peut rallonger énormément la durée du projet. Pire encore, cela peut bloquer l'amélioration et le développement du projet, imposant parfois de repartir de zéro. Nous allons éviter ça !

III - Énoncé

III.1 - Avant-propos :

Internet a évolué ces dernières décennies d'un réseau de calculateurs à un réseau d'ordinateurs personnels, puis vers un réseau qui intègre tout dispositif communiquant : les réseaux de capteurs, les actionneurs, les réseaux véhiculaires, etc. Cette évolution a donné naissance à l'Internet of Things (IoT).

Aujourd'hui, cette technologie est omniprésente dans notre environnement, que ce soit au travail, dans les espaces publics ou à la maison. En effet, les réseaux locaux, les interphones, les systèmes d'alarme et Internet sont devenus une solution incontournable pour améliorer la qualité et le confort de notre quotidien.

Nous pouvons dire que le grand nombre d'appareils électroniques sur le marché, avec leur prix décroissant et l'avancement dans la technologie de télécommunication, a rendu accessible l'utilisation de cette technologie. Nous la retrouvons partout et notamment dans nos maisons, d'où la naissance des systèmes domotique.

La domotique définit l'ensemble des techniques qui permettent de centraliser le contrôle de différents appareils de la maison ou d'un bâtiment. Elle vient donc s'installer sur nos ordinateurs, téléphones et/ou tablettes, pour piloter tous nos objets connectés, avec des solutions appropriées et peu coûteuses.

III.2 - Mission JARVIS et Cahier des charges

Vous êtes un développeur / une développeuse freelance et vous êtes sollicités pour développer ce projet ambitieux nommé JARVIS.

Le client souhaite créer une application mobile qui automatise la gestion des objets connectés ainsi que les autres actions relatives aux domaines de la domotique (sécurité, confort et gestion d'énergie). De plus, une assistance vocale spécialisée, conçue pour faciliter et améliorer le cadre de vie d'un foyer (couple/famille), sera présente dans l'application. Pour finir, toute la famille pourra connecter ses agendas et autres informations personnelles pour cohabiter plus facilement et organiser les activités de la famille. Elle pourra également échanger via une messagerie chiffrée sur les événements familiaux.

Votre mission consistera principalement à créer l'API, qui sera en lien avec l'application mobile, pour gérer les données d'utilisation des objets connectés de la maison.



L'application mobile sera quant à elle développée par une équipe de développeurs, en parallèle de votre travail. Une fois l'API terminée, vous devrez réaliser la connexion API-Application.

Ce projet JARVIS va se réaliser en trois phases bien distinctes :

1. UML/MERISE - Interprétation des diagrammes
 - a. Diagramme de cas d'utilisation
 - b. Modèle Conceptuel de Données (MCD)
 - c. Diagramme de classe
2. SQL/MySQL
 - a. Coming soon
3. Node JS
 - a. Coming soon

Ce projet tend à s'étendre de manière globale afin de faciliter la vie quotidienne à chacun tout en jouant avec les nouveaux appareils qui apparaissent chaque jour, comme les Philips, Hue, Google Home ou bien Alexa. La solution doit prendre en compte uniquement les objets connectés au Wi-Fi (lumière connectée, enceinte, etc. de la marque Philips et Konyks).

L'utilisateur devra être équipé des appareils suivants :

- Ordinateur, Tablette ou Smartphone : permettant la création et gestion de son compte et de ses appareils sur le site web.
- Smartphone ou Tablette : permettant la gestion du comportement des appareils, partage de données avec les utilisateurs du même compte sur l'application mobile.
- Matériels domotiques : Google Home, objets connectés de la marque Philips, Samsung ou Konyks.

La totalité de vos fichiers source, sauf tous les fichiers inutiles (binary, temp files, obj fichiers,...), doivent être inclus dans votre livraison, dépôt Github, en respectant impérativement la nomenclature suivante :

SAYNA-UMLMERISE-JARVIS1-052022

⚠ Si la nomenclature n'est pas respectée, le projet ne sera pas pris en compte lors de la correction et l'évaluation ⚠

⚠ Pensez à mettre votre dépôt en "Public". Le projet ne sera pas corrigé si le dépôt se trouve en "Privé" ⚠



Consignes pour le rendu du projet :

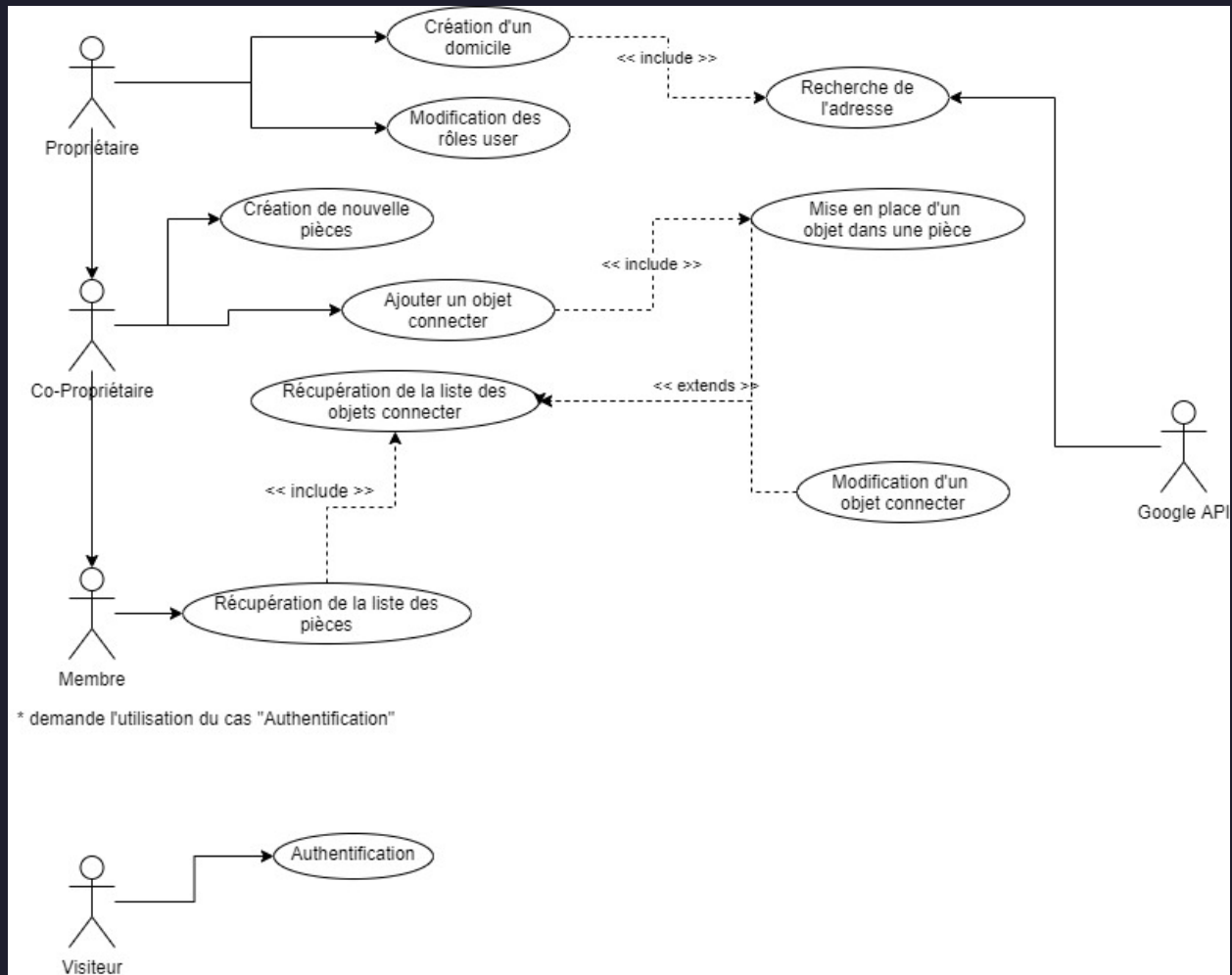
- Nommez votre projet en respectant la nomenclature.
- Déposez votre projet sur Github dès le début et actualisez-le au fur et à mesure de votre avancée.
- Vous scinderez bien les différentes parties du projet.

Notes pour les apprenants :

- Lisez l'intégralité de l'énoncé dès le début pour bien commencer.
- Prenez le temps de coder en commentant votre code dès qu'il est nécessaire.
Il faut que le correcteur puisse comprendre ce que vous avez fait !

IV - Ressources utiles

- Diagramme de cas d'utilisation :



- MCD
 - Coming Soon
- Diagramme de classe
 - Coming soon

V - Tâches à réaliser

Pour vous guider dans la réalisation de ce projet, nous allons vous détailler les différentes tâches à réaliser. Suivi correctement les étapes.

V.1 - Diagramme de cas d'utilisation :

Les cas d'utilisation permettent de décrire l'interaction entre l'acteur et le système. L'idée forte est de dire que l'utilisateur d'un système logiciel a un objectif quand il utilise le système ! Le cas d'utilisation est une description des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système. Les use case (cas d'utilisation) sont représentés par une ellipse sous-titrée par le nom du cas d'utilisation (éventuellement le nom est placé dans l'ellipse). Un acteur et un cas d'utilisation sont mis en relation par une association représentée par une ligne.

Le plus souvent, le diagramme de cas d'utilisation est établi par la maîtrise d'ouvrage (MOA) d'un projet lors de la rédaction du cahier des charges afin de transmettre les besoins des utilisateurs et les fonctionnalités attendues associées à la maîtrise d'œuvre (MOE).

Vous devez faire preuve de bon sens, car peu de précisions vous sont apportées dans cette partie. C'est justement l'objectif : se poser des questions et arriver à comprendre la logique des diagrammes.

DESCRIPTION

1. À partir des diagrammes, identifier les différents acteurs de l'application
2. À partir des diagrammes, détailler la liste des fonctionnalités pour chaque acteur
3. Parmi l'ensemble des fonctionnalités que vous avez listées, identifiez celles qui sont obligatoires (include) et celles qui sont facultatives (exclude).

V.2 - MCD :

4. Coming soon

V.3 - Diagramme de classe :

5. Coming soon