**République Islamique de Mauritanie**

**Ministère de l’Enseignement Supérieur et de la Recherche Scientifique**



**Institut Supérieur du Numérique**

**RAPPORT DU STAGE**

***Thème***

**DEVELOPEMENT D’UNE APPLICATION MOBILE CROSSPLATFORME**

**POUR LA VENDRE DE GLACE**

***Réalisé par***

**MOHAMED LEMINE EL HAJ (22043)**

**AHMED SALIM SHENAN (22050)**

**FILIERE : Développement des Systèmes Informatiques**

***Encadré par***

D.AYNINA RABI

*Année universitaire 2023-2024*

***Remerciement***

Nous tenons tout d'abord à exprimer notre profonde gratitude et nos sincères remerciements à tous ceux qui ont contribué à la réalisation de ce fascicule de projet. Ce document représente le fruit de nombreuses heures de travail, de dévouement et de passion de la part de chaque membre de notre équipe.

Nous souhaitons tout particulièrement exprimer notre reconnaissance à notre encadreur Dr Sidi Mohamed, dont l'expertise, la patience et les précieux conseils ont été d'une importance cruciale tout au long de notre parcours. Votre soutien indéfectible et vos encouragements nous ont permis de surmonter les obstacles et de nous dépasser constamment.

Nous tenons également à remercier l'ensemble de nos proches, amis et familles, qui ont été une source inépuisable de soutien moral et d'encouragement. Vos encouragements et votre compréhension face à nos longues heures de travail ont été essentiels pour notre réussite.

Nous ne saurions oublier les différentes personnes qui ont contribué de près ou de loin à la réalisation de ce projet. Que ce soit par leurs conseils, leur expertise ou leur participation active, chacune de vos contributions a été inestimable. Votre collaboration nous a permis d'enrichir nos connaissances, de découvrir de nouvelles perspectives et de repousser les limites de notre projet.

Ce fascicule de projet est le résultat d'un travail collectif et de l'engagement de nombreux acteurs. Nous sommes fiers de présenter aujourd'hui le fruit de nos efforts et espérons qu'il contribuera à la compréhension, à l'innovation et à l'avancement dans notre domaine.

En conclusion, nous tenons à remercier chaleureusement chacune des personnes mentionnées et à exprimer notre gratitude à tous ceux qui ont soutenu notre projet. Votre confiance et votre appui ont été les moteurs de notre réussite.

**Table des matières**

[**Introduction** **1**](#_Toc168436851)

[**CHAP 1 : Cadre générale du projet** 2](#_Toc168436852)

[I. Problématique : 2](#_Toc168436853)

[ Coordination des équipes : 2](#_Toc168436854)

[ Gestion des dépendances : 2](#_Toc168436855)

[ Gestion des risques : 2](#_Toc168436856)

[ Gestion du changement : 2](#_Toc168436857)

[ Contrôle de la qualité : 3](#_Toc168436858)

[ Gestion des parties prenantes : 3](#_Toc168436859)

[II- Présentation du sujet : 3](#_Toc168436860)

[ Développer un plan de projet global : 3](#_Toc168436861)

[ Coordonner les parties prenantes : 3](#_Toc168436862)

[ Gérer les risques : 4](#_Toc168436863)

[**CHAP 2 : Conception :** **5**](#_Toc168436865)

[I. Diagramme des classes : 5](#_Toc168436867)

[II. Architecture logicielle : 6](#_Toc168436867)

[III. Conception de l'interface utilisateur (UI) : 6](#_Toc168436869)

[IV. Expérience utilisateur (UX) : 6](#_Toc168436871)

[V. Prototypes et tests : 6](#_Toc168436873)

[VI. Les besoins fonctionnels : 7](#_Toc168436875)

**[CHAP 3 : Réalisation](#_Toc168436877)****[9](#_Toc168436877)**

[I. Environnement : 9](#_Toc168436878)

[I.1. Environnement matériel : 9](#_Toc168436879)

[I.2. Environnement logiciels : 9](#_Toc168436880)

[I.2.1. Les outils : 9](#_Toc168436881)

[II. Interface graphique du projet 13](#_Toc168436882)

[**Conclusion :** **20**](#_Toc168436883)

**Les références : ..…………………………………..………….………………21**

**Table des figures**

[**Figure 1 :** **DIAGRAMME - CLASS 5**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511914)

[**Figure 2 : VS CODE - LOGO 9**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511914)

[**Figure 3 : ANDROID STUDIO - LOGO 10**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511915)

[**Figure 4 : MONGODB - LOGO 10**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511916)

[**Figure 5 : MODELIO - LOGO 10**](#_Toc140511917)

[**Figure 6 : GITHUB - LOGO 11**](#_Toc140511917)

[**Figure 7 : DART – LOGO 11**](#_Toc140511918)

[**Figure 8 :** **APP-LOGO 13**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511920)

[**Figure 9 : ACCUEIL 13**](#_Toc140511921)

[**Figure 10 : POETES 14**](#_Toc140511922)

[**Figure 11 : RECUEILS PAR POETE 14**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511923)

[**Figure 12 : POEMES PAR RECUEIL 15**](#_Toc140511924)

[**Figure 13 : POEME CONTENU 15**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511925)

[**Figure 14 : POETES FAVORIS 16**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511926)

[**Figure 15 : POEMES FAVORIS 16**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511927)

[**Figure 16 : RECUEILS (DEEWANS) 17**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511928)

[**Figure 17 : RECHERCHE 17**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511929)

[**Figure 18 : RECHERCHE RESULTAT 18**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511929)

[**Figure 19 : INFO POETE 18**](file:///C:\xampp\htdocs\tp%20php\Projet_PI\rapport%20du%20projet.docx#_Toc140511929)

**Figure 20 : INFO POEME 19**

# **Introduction**

Dans le cadre de mon stage au sein du bureau d'étude de [Nom de l'entreprise], j'ai eu l'opportunité de travailler sur un projet commandé par la Fédération des fabriques de glace. Cette mission avait pour objectif de concevoir et développer une application mobile et une interface administrateur destinées à faciliter la gestion des demandes entre les clients et les propriétaires des fabriques de glace.

L'importance de ce projet réside dans la nécessité d'améliorer l'efficacité des interactions entre les différents acteurs du secteur de la fabrication de glace. Traditionnellement, les demandes étaient traitées de manière manuelle, ce qui entraînait des délais de traitement plus longs, une plus grande marge d'erreur et une insatisfaction générale des clients. Grâce à une solution informatique centralisée, notre projet visait à automatiser et sécuriser ces échanges, offrant ainsi une meilleure expérience utilisateur et une gestion optimisée.

Le projet s'est articulé autour de deux principaux composants : une application mobile destinée aux clients pour créer et suivre leurs demandes, et une interface administrateur permettant de gérer les opérations CRUD (Create, Read, Update, Delete) sur l'ensemble des données de la base de données. L'application mobile a été conçue pour être intuitive, sécurisée et accessible, tandis que l'interface administrateur devait offrir une vue d'ensemble efficace et des fonctionnalités avancées de gestion.

Notre équipe de stagiaires a travaillé en étroite collaboration avec la Fédération des fabriques de glace pour comprendre leurs besoins spécifiques et développer une solution sur mesure. Le projet a inclus une phase d'analyse et de définition des exigences, une phase de conception et de développement, et enfin une phase de test et de déploiement. Chaque étape a été réalisée en suivant les meilleures pratiques du développement logiciel, assurant ainsi la qualité et la fiabilité du produit final.

Cette expérience de stage a été extrêmement enrichissante, nous permettant de mettre en pratique nos connaissances théoriques dans un contexte réel et de contribuer de manière tangible à un projet de grande envergure. Ce rapport détaille les différentes étapes du projet, les technologies utilisées et les résultats obtenus, en mettant en lumière les défis rencontrés et les solutions apportées.

# **CHAP 1 : Cadre générale du projet**

## **Problématique** :

La mise en place de ce projet a répondu à plusieurs problématiques spécifiques liées à la gestion des demandes entre les clients et les propriétaires des fabriques de glace. Ces problématiques incluent la coordination des équipes, la gestion des dépendances, la gestion des risques, la gestion du changement, le contrôle de la qualité et la gestion des parties prenantes.

Voici quelques problématiques courantes auxquelles on peut être confronté dans le projet :

* Coordination des équipes :

La coordination entre les différentes équipes impliquées dans le projet, telles que les développeurs, les concepteurs et les utilisateurs finaux, est essentielle pour assurer une communication efficace et éviter les malentendus. Chaque équipe doit comprendre ses responsabilités et travailler en harmonie avec les autres pour atteindre les objectifs communs.

* Gestion des dépendances :

Le projet repose sur plusieurs composants interdépendants, tels que l'application mobile, l'interface administrateur et la base de données. Il est crucial de gérer ces dépendances de manière à ce que chaque composant soit développé et testé de manière cohérente, minimisant ainsi les risques de dysfonctionnement.

* Gestion des risques :

Comme tout projet informatique, celui-ci comporte des risques potentiels, tels que des retards dans le développement, des erreurs de programmation, ou des problèmes de sécurité. Une évaluation continue des risques et la mise en place de mesures de mitigation sont nécessaires pour garantir la réussite du projet.

* Gestion du changement :

Les besoins des utilisateurs peuvent évoluer au fil du temps, nécessitant des ajustements dans les fonctionnalités ou la conception du produit. Il est important d'avoir des processus en place pour gérer ces changements de manière structurée, évitant ainsi des perturbations majeures dans le développement.

* Contrôle de la qualité :

Assurer la qualité du produit final est primordial pour répondre aux attentes des utilisateurs. Cela inclut des tests rigoureux à chaque étape du développement, la correction des bugs, et l'optimisation des performances

* Gestion des parties prenantes :

Les parties prenantes du projet incluent les clients finaux, les propriétaires des fabriques, et la Fédération des fabriques de glace. Une communication régulière et transparente avec toutes les parties prenantes est essentielle pour s'assurer que leurs attentes sont comprises et prises en compte.

En répondant à ces problématiques, le projet a permis de développer une solution complète et fiable pour la gestion des demandes de glace. L'application mobile offre aux clients une plateforme sécurisée pour créer et suivre leurs demandes, tandis que l'interface administrateur permet aux propriétaires de gérer efficacement leurs opérations. Ce projet a démontré l'importance de la planification, de la coordination et du contrôle de la qualité dans la réussite d'une initiative de développement logiciel.

## II- Présentation du sujet :

Le projet de développement d'une application mobile et d'une interface administrateur pour la gestion des demandes entre les clients et les propriétaires des fabriques de glace a été initié par la Fédération des fabriques de glace. Cette initiative avait pour but de moderniser et d'optimiser la gestion des demandes, en offrant une solution technologique adaptée aux besoins des différents acteurs du secteur.

La gestion des fabriques de glace implique généralement les responsabilités suivantes :

* Développer un plan de projet global :

La première étape du projet a consisté à élaborer un plan de projet détaillé, définissant les objectifs, les livrables, les délais et les ressources nécessaires. Ce plan a servi de guide tout au long du développement, assurant une coordination efficace des efforts et une gestion proactive des risques.

* Coordonner les parties prenantes :

La Fédération des fabriques de glace, les propriétaires des fabriques et les clients finaux étaient les principales parties prenantes du projet. Une communication régulière et des réunions de suivi ont permis de recueillir leurs retours et de s'assurer que la solution développée répondait à leurs attentes.

* Gérer les risques :

La gestion des risques a été une composante clé du projet. Une analyse des risques a été réalisée au début du projet pour identifier les éventuels obstacles et définir des stratégies de mitigation. Les risques techniques, tels que les problèmes de sécurité ou les erreurs de programmation, ont été suivis de près et des tests rigoureux ont été effectués pour garantir la qualité du produit final.

L'application mobile, destinée aux clients, permet de créer et de suivre les demandes de glace de manière sécurisée. Les fonctionnalités incluent l'authentification par email et mot de passe, la création de demandes avec spécification des quantités, et le suivi de l'état des demandes (attendant, acceptée, refusée). L'interface utilisateur a été conçue pour être intuitive et facile à utiliser, même pour les utilisateurs non technophiles.

L'interface administrateur, quant à elle, permet aux propriétaires de fabriques de gérer les opérations CRUD sur l'ensemble des données de la base de données. Les administrateurs peuvent consulter, ajouter, modifier et supprimer des enregistrements de clients, de propriétaires et de demandes. Cette interface assure une vue d'ensemble claire et une gestion efficace des données.

En conclusion, la présentation du sujet met en lumière les différentes étapes et composantes du projet, démontrant l'importance d'une planification rigoureuse, d'une communication efficace et d'une gestion proactive des risques pour assurer la réussite d'un projet de développement logiciel.

# **CHAP 2 : Conception**

## La conception du projet de gestion des demandes entre les clients et les propriétaires des fabriques de glace s'est structurée autour de plusieurs aspects clés : l'architecture logicielle, la conception de l'interface utilisateur (UI), l'expérience utilisateur (UX), les prototypes et tests, et les besoins fonctionnels. Le diagramme de classes fourni (illustré ci-dessus) a été essentiel pour comprendre et structurer les relations entre les différentes entités du système.

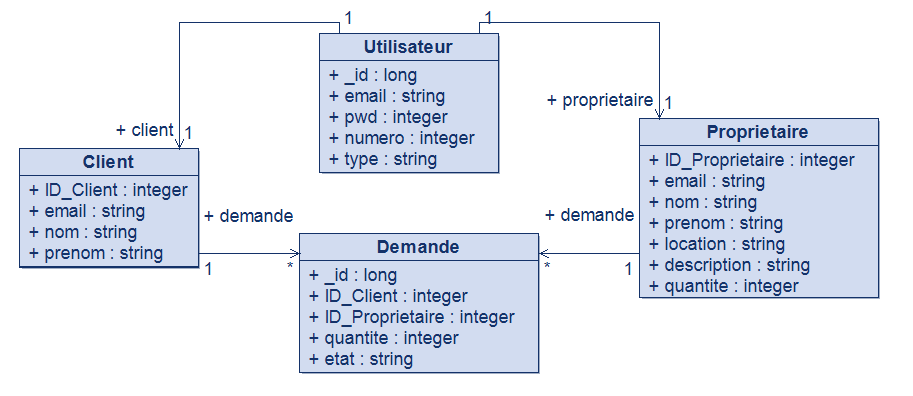
1. Le diagramme des classes :

Figure 1 : Diagramme - class

Figure 1 : VS CODE - LOGO

Le diagramme de classes présente les différentes entités impliquées dans le projet de gestion des demandes entre clients et propriétaires des fabriques de glace. Chaque classe représente une table dans la base de données, avec ses attributs et ses relations avec les autres classes.

1. Classe Utilisateur

Attributs :

\_id : long : Identifiant unique de l'utilisateur.

Email : string : Adresse email de l'utilisateur.

Pwd : integer : Mot de passe de l'utilisateur (idéalement devrait être stocké sous forme de hachage pour des raisons de sécurité).

Numero : integer : Numéro de téléphone de l'utilisateur.

Type : string : Type d'utilisateur, qui peut être soit "client" soit "propriétaire".

Relations :

1 Utilisateur est lié à 1 Client ou Propriétaire. Cela signifie qu'un utilisateur peut être soit un client soit un propriétaire, mais pas les deux.

1. Classe Client

Attributs :

ID\_Client : integer : Identifiant unique du client.

Email : string : Adresse email du client.

Nom : string : Nom du client.

Prenom : string : Prénom du client.

Relations :

1 Client est lié à 1..\* Demandes. Un client peut créer plusieurs demandes.

1. Classe Propriétaire

Attributs :

ID\_Proprietaire : integer : Identifiant unique du propriétaire.

Email : string : Adresse email du propriétaire.

Nom : string : Nom du propriétaire.

Prenom : string : Prénom du propriétaire.

Location : string : Localisation de la fabrique de glace.

Description : string : Description de la fabrique de glace.

Quantité : integer : Quantité de glace disponible chez le propriétaire.

Relations :

1 Propriétaire est lié à 1..\* Demandes. Un propriétaire peut recevoir plusieurs demandes de différents clients.

1. Classe Demande

Attributs :

\_id : long : Identifiant unique de la demande.

ID\_Client : integer : Identifiant du client qui a fait la demande.

ID\_Proprietaire : integer : Identifiant du propriétaire à qui la demande est faite.

Quantité : integer : Quantité de glace demandée.

Etat : string : État de la demande (par exemple, "attendant", "acceptée", "refusée").

Relations :

1 Demande est liée à 1 Client. Chaque demande est faite par un seul client.

1 Demande est liée à 1 Propriétaire. Chaque demande est adressée à un seul propriétaire.

Résumé des Relations :

1. Utilisateur et Client/Propriétaire :

Un utilisateur peut être soit un client soit un propriétaire, mais pas les deux. Cette relation est indiquée par une ligne reliant la classe Utilisateur à la fois à Client et à Propriétaire, avec une cardinalité de 1 de chaque côté.

1. Client et Demande :

Un client peut créer plusieurs demandes (relation 1 à plusieurs). Cela est représenté par une ligne reliant Client à Demande, avec une cardinalité 1 du côté du client et 1..\* du côté des demandes.

1. Propriétaire et Demande :

Un propriétaire peut recevoir plusieurs demandes (relation 1 à plusieurs). Cette relation est représentée par une ligne reliant Propriétaire à Demande, avec une cardinalité 1 du côté du propriétaire et 1..\* du côté des demandes.

Ce modèle permet une gestion efficace des utilisateurs (clients et propriétaires), ainsi que des demandes de glace, en assurant que chaque demande est correctement associée à un client et à un propriétaire spécifique.

1. Architecture logicielle :

L'architecture logicielle de ce projet repose sur une séparation claire des responsabilités entre le frontend (interface utilisateur) et le backend (logique métier et gestion des données). Cette approche assure une meilleure modularité, facilitant ainsi la maintenance et l'évolution du système.

Backend : Le backend a été développé en utilisant un Framework web (Flask pour Python) qui gère les requêtes HTTP, la logique métier, et l'interaction avec la base de données. La base de données choisie est MongoDB, une base de données NoSQL qui permet une gestion flexible et évolutive des données.

Frontend : Le frontend de l'application mobile a été développé en utilisant un Framework multiplateforme (Flutter), permettant de déployer l'application sur Android et iOS avec un seul codebase. L'interface administrateur a été développée en utilisant des technologies web telles que HTML, CSS, et JavaScript.

API REST : Une API RESTful a été mise en place pour permettre la communication entre le frontend et le backend. Cette API expose des Endpoints pour les opérations CRUD sur les entités de la base de données (Utilisateur, Client, Propriétaire, Demande).

## Conception de l'interface utilisateur (UI) :

## La conception de l'interface utilisateur (UI) a été centrée sur la simplicité et l'intuitivité, permettant aux utilisateurs de naviguer et d'interagir avec l'application de manière fluide.

## Application mobile : L'interface de l'application mobile inclut des écrans pour la connexion, la création de demandes, et le suivi des demandes. Des éléments de design cohérents et des icônes claires ont été utilisés pour guider les utilisateurs à travers les différentes fonctionnalités.

## Interface administrateur : L'interface administrateur comprend des tableaux de bord pour la visualisation des données, des formulaires pour la gestion des enregistrements, et des filtres pour rechercher et trier les informations. L'UI a été conçue pour offrir une vue d'ensemble rapide et faciliter les opérations de gestion.

## Expérience utilisateur (UX) :

## L'expérience utilisateur (UX) a été optimisée en se basant sur les retours des utilisateurs potentiels et en suivant les meilleures pratiques en matière de design d'interfaces.

## Facilité d'utilisation : L'application mobile et l'interface administrateur ont été conçues pour être faciles à utiliser, même pour des utilisateurs non techniques. Les processus de création et de suivi des demandes ont été simplifiés pour minimiser le nombre d'étapes nécessaires.

## Réactivité : L'application mobile et l'interface administrateur ont été développées pour être réactives, offrant une expérience utilisateur fluide et sans latence. L'utilisation de Frameworks modernes a permis d'assurer des temps de réponse rapides et une navigation sans interruptions.

## Accessibilité : Des efforts ont été faits pour rendre l'application accessible à tous les utilisateurs, y compris ceux ayant des besoins spécifiques. Des contrastes de couleurs adéquats, des tailles de police ajustables, et des descriptions alternatives pour les éléments visuels ont été intégrés.

## Prototypes et tests :

Avant le développement complet, des prototypes de l'application mobile et de l'interface administrateur ont été créés pour valider les concepts de design et recueillir des retours d'utilisateurs.

Prototypes : Les prototypes ont été développés en utilisant des outils de design (comme Figma ou Adobe XD) et ont permis de visualiser et d'interagir avec les différentes interfaces de l'application. Ces prototypes ont été présentés aux parties prenantes pour validation.

Tests utilisateurs : Des tests utilisateurs ont été réalisés avec un groupe restreint de clients et de propriétaires de fabriques. Les retours obtenus ont permis d'identifier des points de friction et d'améliorer l'interface avant le lancement.

Tests de performance : Des tests de performance ont été effectués pour s'assurer que l'application et l'interface administrateur peuvent gérer des charges de travail élevées sans dégradation des performances. Des outils de monitoring ont été utilisés pour suivre et optimiser les temps de réponse.

## Les besoins fonctionnels :

Les besoins fonctionnels ont été définis en collaboration avec la Fédération des fabriques de glace et ont servi de base pour le développement des différentes fonctionnalités de l'application.

Authentification : L'application mobile nécessite une authentification sécurisée par email et mot de passe pour accéder aux fonctionnalités. Cela garantit que seules les personnes autorisées peuvent créer et suivre des demandes.

Gestion des demandes : Les clients peuvent créer des demandes en spécifiant la quantité de glace requise et le propriétaire de la fabrique. Ils peuvent également suivre l'état de leurs demandes (attendant, acceptée, refusée) à travers l'application.

Interface administrateur : Les administrateurs de fabriques peuvent accéder à une interface administrateur pour gérer les enregistrements de clients, de propriétaires, et de demandes. Cette interface permet de consulter, ajouter, modifier, et supprimer des enregistrements, offrant une gestion complète et efficace des données.

Sécurité des données : Des mesures de sécurité, telles que le chiffrement des mots de passe et l'utilisation de connexions sécurisées (HTTPS), ont été mises en place pour protéger les données des utilisateurs et garantir la confidentialité des informations échangées.

En conclusion, la conception de ce projet a impliqué une planification rigoureuse, une attention particulière aux détails de l'interface utilisateur et de l'expérience utilisateur, ainsi que des tests approfondis pour garantir la qualité et la performance du produit final. Le diagramme de classes a été un outil précieux pour structurer les relations entre les différentes entités et s'assurer que toutes les exigences fonctionnelles et non fonctionnelles étaient correctement adressées. Le résultat est une solution intégrée qui répond aux besoins des clients et des propriétaires de fabriques, facilitant ainsi la gestion des demandes de manière efficace et sécurisée.

# **CHAP 3 : Réalisation**

# 

## Environnement :

## 

Un environnement désigne, pour une application l’ensemble de matériels et des logiciels système, dont le système d’exploitation, sur lesquels sont exécutés les programmes de l’application.

## I.1. Environnement matériel :

|  |  |
| --- | --- |
| Ordinateur | HP pavillon |
| Processeur | Intel(R) Core(TM) i5-11G |
| Mémoire RAM | 16.00GB |
| Disque dur | 1TB |
| Système d’exploitation | Windows 10 Pro 64bits |

## I.2. Environnement logiciels :

### I.2.1. Les outils :

Le logiciel utilisé pour l’implémentation de notre solution est :

#### Visual Studio Code :

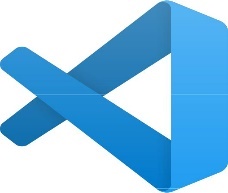
(VSC par la suite) est un éditeur de code open-source, gratuit et multiplateforme (Windows, Mac et Linux), développe par Microsoft, a` ne pas confondre avec Visual Studio, l’IDE propriétaire de Microsoft. VSC est développe avec Electron et exploite des fonctionnalités d’édition avancées du projet Monaco Editor. Principalement conçu pour le d´développement d’application avec JavaScript, l´éditeur peut s’adapter a` d’autres types de langages grâce a` un système d’extension bien fourni [1]

Figure 2 : VS CODE - LOGO

Figure 2 : VS CODE - LOGO

Figure 2 : VS CODE - LOGO

Figure 3 : VS CODE - LOGO

* Android studio :

Android Studio est un environnement de développement intégré (IDE) pour créer des applications Android. Il comprend un éditeur de code, des outils de construction et un gestionnaire de packages, parmi de nombreuses autres ressources. Dans cet article, vous pourrez voir tout son potentiel et en savoir plus pour commencer à développer.[2]



Figure 3 : Android Studio - LOGO

#### MongoDB :

MongoDB est une base de données orientée documents. En clair, vous bénéficiez de la scalabilité et de la flexibilité que vous voulez, avec les fonctions d’interrogation et d’indexation qu’il vous faut. [3]



Figure 4 : mongoDB - LOGO

#### Modelio :

Modelio, l’outil de modélisation pour le développement de logiciels, la gestion de processus métiers et l’ingénierie des systèmes, passe en version 2.1.1, et accentue son ouverture open source.

Modelio est par exemple désormais disponible nativement en format 64 bits ou 32 bits sous les différentes plateformes Linux RedHat, Ubuntu et Debian et gère la documentation Libre Office (en plus du HTML et de Microsoft Word). [4]



### 

Figure 5 : Modelio - LOGO

Figure 4 : Dart – LOGOFigure 5 : Modelio - LOGO

Figure 6 : Dart – LOGO

Figure 7 : Figure 8 : Dart – LOGOFigure 9 : Modelio - LOGO

Figure 10 : Dart – LOGOFigure 11 : Modelio - LOGO

Figure 6 : Dart – LOGO

Figure 12 : Figure 13 : Dart – LOGO

Figure 14 : logo

Figure 15 : accueilFigure 16 : Figure 17 : Dart – LOGO

Figure 18 : Figure 19 : Dart – LOGOFigure 5 : Modelio - LOGO

Figure 20 : Dart – LOGOFigure 21 : Modelio - LOGO

Figure 22 : Dart – LOGO

Figure 23 : Figure 24 : Dart – LOGOFigure 25 : Modelio - LOGO

Figure 26 : Dart – LOGOFigure 27 : Modelio - LOGO

#### GitHub:

A un haut niveau, GitHub est un site web et un service de cloud qui aide les développeurs à stocker et à gérer leur code, ainsi qu’à suivre et contrôler les modifications qui lui sont apportées. [6]



Figure 5 : GitHub - LOGO

Figure 28 : Dart – LOGOFigure 29 : Modelio - LOGO

Figure 30 : Dart – LOGO

Figure 31 : Figure 32 : Dart – LOGOFigure 33 : Modelio - LOGO

Figure 34 : Dart – LOGOFigure 35 : Modelio - LOGO

Figure 6 : Dart – LOGO

Figure 36 : Figure 37 : Dart – LOGO

Figure 38 : logo

Figure 39 : accueilFigure 40 : Figure 41 : Dart – LOGO

Figure 42 : Figure 43 : Dart – LOGOFigure 5 : Modelio - LOGO

Figure 44 : Dart – LOGOFigure 45 : Modelio - LOGO

Figure 46 : Dart – LOGO

Figure 47 : Figure 48 : Dart – LOGOFigure 49 : Modelio - LOGO

Figure 50 : Dart – LOGOFigure 51 : Modelio - LOGO

I.2.2. Les technologies :

#### Introduction :

#### Dans le développement d'applications mobiles modernes, plusieurs langages de programmation et technologies jouent un rôle crucial dans la création d'applications dynamiques et interactives. Ce rapport vise à fournir un bref aperçu des principaux langages utilisés dans le développement d'applications mobiles : Dart, Kotlin, Swift et JavaScript. Chaque langage remplit un objectif spécifique et contribue à différents aspects du développement d'applications mobiles

#### Dart :

#### 

Figure 6 : Dart – LOGO

Figure 52 : Figure 53 : Dart – LOGO

Figure 54 : logo

Figure 55 : accueilFigure 56 : Figure 57 : Dart – LOGO

Figure 58 : Figure 59 : Dart – LOGO

Figure 7 : App-logo

Figure 60 : accueilFigure 61 : logo

Figure 62 : accueil

Figure 63 : Figure 64 : accueilFigure 65 : logo

Figure 66 : accueilFigure 67 : Figure 6 : Dart – LOGO

Figure 68 : Figure 69 : Dart – LOGO

Figure 70 : logo

Figure 71 : accueilFigure 72 : Figure 73 : Dart – LOGO

Figure 74 : Figure 75 : Dart – LOGO

Objectif :

Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, de serveur et web. [5]

* Principales caractéristiques :

Langage compilé : Dart est compilé en code natif, offrant des performances élevées et une exécution rapide des applications mobiles.

* Framework Flutter :

Dart est le langage principal utilisé avec Flutter, un framework de développement d'interface utilisateur multiplateforme, permettant la création d'applications mobiles pour Android, iOS et d'autres plateformes.

* Hot Reload :

Dart offre la fonctionnalité de rechargement à chaud (Hot Reload) dans Flutter, permettant aux développeurs de voir instantanément les modifications apportées au code sans redémarrer l'application.

* Programmation asynchrone :

Dart prend en charge la programmation asynchrone, facilitant le développement d'applications réactives qui peuvent gérer efficacement les opérations IO et les appels réseau sans bloquer l'interface utilisateur.

## II. Interface graphique du projet

* Le logo de l’application :

Le logo de notre application de poésie mauritanienne

incarne l'essence et la culture de la poésie locale.

illustré avec une plume stylisée et des éléments naturels

comme les palmiers et le soleil, il évoque l'inspiration et

la tradition littéraire de la Mauritanie. Ce logo sert de

symbole unificateur pour l'application, rappelant aux

utilisateurs l'authenticité et la richesse de la poésie

mauritanienne à chaque interaction.

Figure 7 : App-logo

Figure 76 : accueilFigure 77 : logo

Figure 78 : accueil

Figure 79 : Figure 80 : accueilFigure 81 : logo

Figure 82 : accueilFigure 83 : logo

Figure 8 : accueil

Figure 84 : Figure 85 : accueil

Figure 86 : Figure 87 : accueil

Figure 88 : Figure 89 : accueilFigure 7 : App-logo

Figure 90 : accueilFigure 91 : logo

Figure 92 : accueil

Figure 93 : Figure 94 : accueilFigure 95 : logo

Figure 96 : accueilFigure 97 : logo



* Page d’accueil :

La page d'accueil de l'application est conçue pour être accueillante

et intuitive. Elle présente un accès rapide aux principales

fonctionnalités telles que la recherche de poèmes, la navigation

par auteurs et recueils, et les favoris. Des bannières dynamiques

mettent en avant les nouveautés, les poèmes populaires et

les événements littéraires en cours, offrant ainsi une expérience

engageante dès le premier contact avec l'application.

Figure 8 : accueil

Figure 98 : Figure 99 : accueil

Figure 100 : Figure 101 : accueil

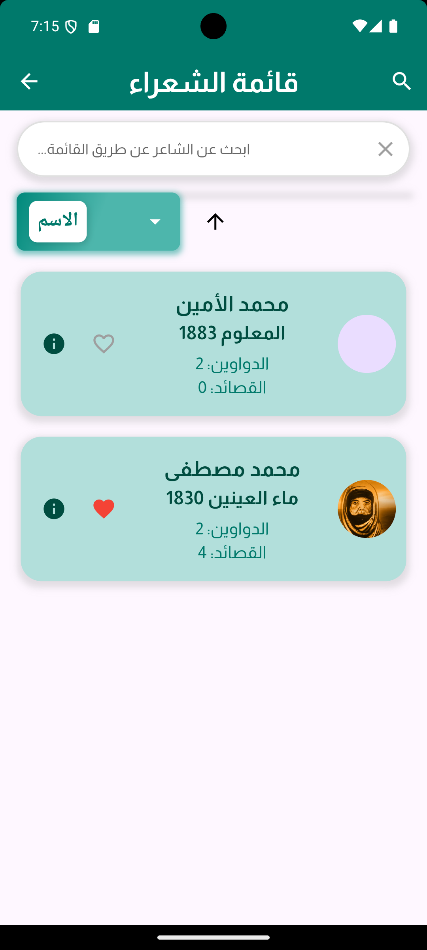
Figure 102 : Figure 103 : accueil

Figure 9 : Figure 8 : accueil

Figure 104 : Figure 105 : accueil

Figure 106 : Figure 107 : accueil

Figure 108 : Figure 109 : accueil

*  Page des poètes :

La page des poètes permet aux utilisateurs de découvrir et

d'explorer les différents auteurs de poèmes.

Chaque poète est présenté avec une photo, une courte biographie

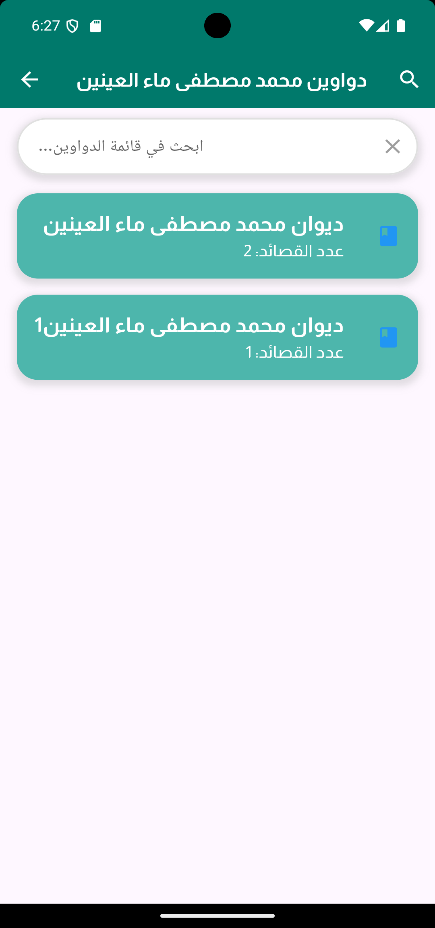
et une liste de ses œuvres. Cette page facilite la navigation et

permet aux utilisateurs d'approfondir leurs connaissances

sur les contributeurs de la poésie mauritanienne.

Figure 9 : poètes

Figure 9 : poètes



* Page des Recueils par poète :

Cette page liste les recueils (deewans) de chaque

poète sélectionné. Chaque recueil est accompagné

d'une courte description et du nombre de poèmes

qu'il contient. Les utilisateurs peuvent cliquer sur

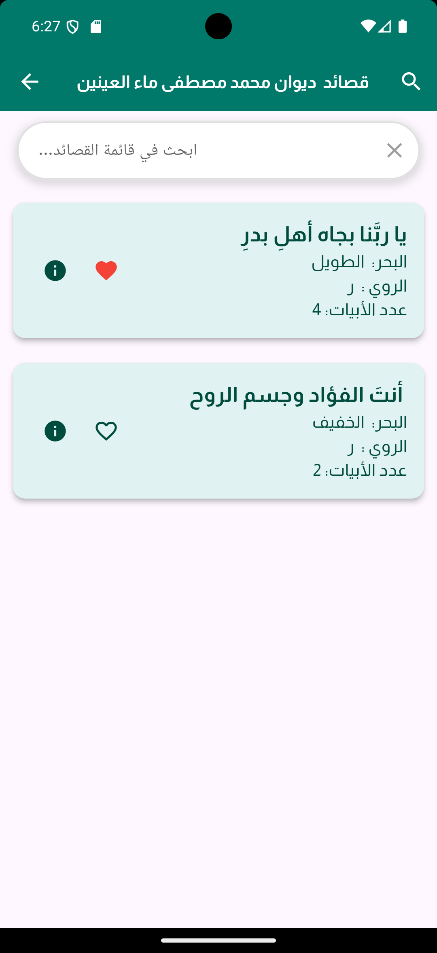
un recueil pour voir la liste détaillée des poèmes

qu'il renferme, favorisant ainsi une exploration

structurée des œuvres d'un poète.

Figure 10 : Recueils par poète

Figure 10 : Recueils par poète

* Page des Poèmes par recueil :

Sur cette page, les utilisateurs peuvent explorer les poèmes

contenus dans un recueil spécifique.

Chaque poème est affiché avec son titre et un extrait,

permettant aux utilisateurs de choisir facilement le poème

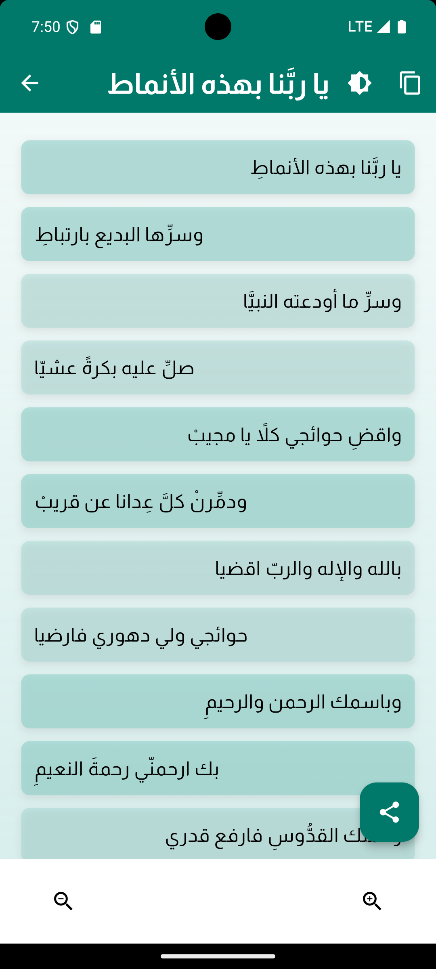
qu'ils souhaitent lire en entier. Cette présentation facilite

la navigation au sein des œuvres et permet une découverte

approfondie de chaque recueil.

Figure 11 : Poèmes par recueil

Figure 11 : Poèmes par recueil



* Page du contenu d’une Poème :

Cette page affiche le contenu complet d'un poème sélectionné.

Avec une mise en page soignée et une typographie agréable,

elle offre une expérience de lecture immersive.

Des options de personnalisation, comme la taille de la police ,

partager ,copier , et le mode nuit, sont disponibles pour

améliorer le confort de lecture.

Cette page peut également afficher des annotations ou

des explications pour enrichir l'expérience de l'utilisateur.

Figure 12 : Poème contenu

Figure 12 : Poème contenu

* Page des poètes favoris :

La page des poètes favoris permet aux utilisateurs de gérer et

d'accéder rapidement à leur liste d'auteurs préférés.

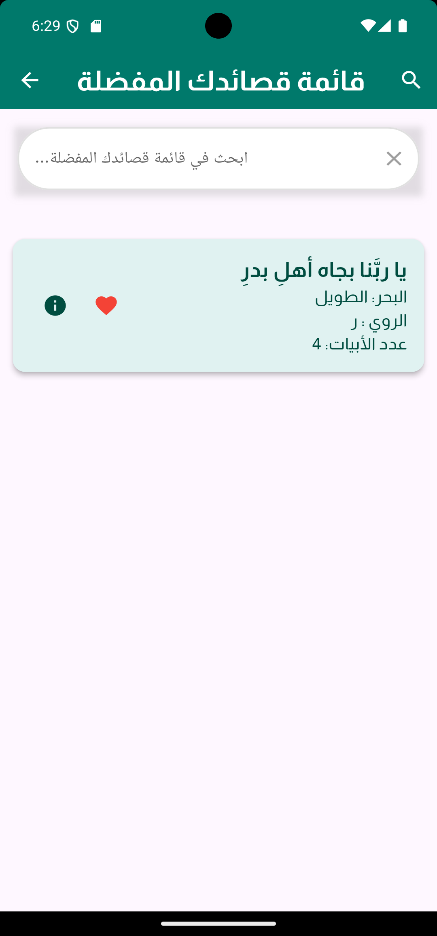
Chaque poète favori est affiché avec un accès rapide à ses œuvres,

facilitant ainsi une navigation personnalisée et une interaction

fréquente avec les poètes les plus appréciés par l'utilisateur.

Figure 13 : Poètes favoris

Figure 13 : Poètes favoris

* Page des Poèmes favoris :

Cette page rassemble tous les poèmes que l'utilisateur a marqués

comme favoris. Les poèmes sont listés avec leurs titres et

les premiers vers, permettant un accès rapide et facile.

Cette fonctionnalité aide les utilisateurs à organiser leurs

lectures et à revenir facilement sur leurs poèmes préférés.

Figure 14 : Poèmes favoris

Figure 14 : Poèmes favoris

* Page des recueils (deewans) :

La page des recueils offre une vue d'ensemble de tous les recueils

disponibles dans l'application. Les recueils sont classés par

ordre alphabétique ou par auteur, avec des filtres disponibles

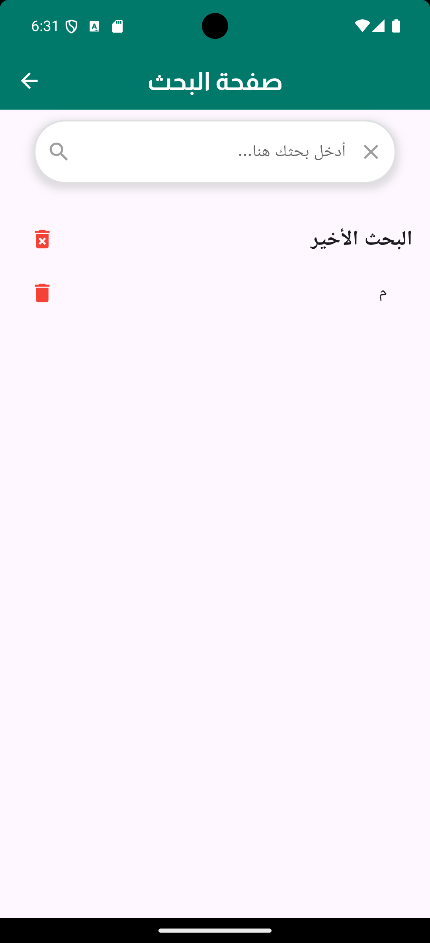
pour affiner la recherche. Chaque recueil est présenté avec

une image de couverture et une brève description, rendant

la navigation visuellement attrayante et informative.

Figure 15 : recueils (deewans)

Figure 15 : recueils (deewans)

* Page de la recherche :

La page de recherche est un outil puissant et convivial qui permet

aux utilisateurs de trouver rapidement des poèmes, des auteurs ou

des recueils. Avec des options de filtrage avancées et une barre

de recherche intuitive, cette page facilite la localisation

des contenus spécifiques et enrichit l'expérience utilisateur

par une navigation simplifiée.

Figure 16 : recherche

Figure 16 : recherche

* Page du résultat de la recherche :

Les résultats de la recherche sont affichés de manière claire et

organisée sur cette page. Chaque résultat est présenté avec

des informations pertinentes comme le titre, l'auteur et un extrait

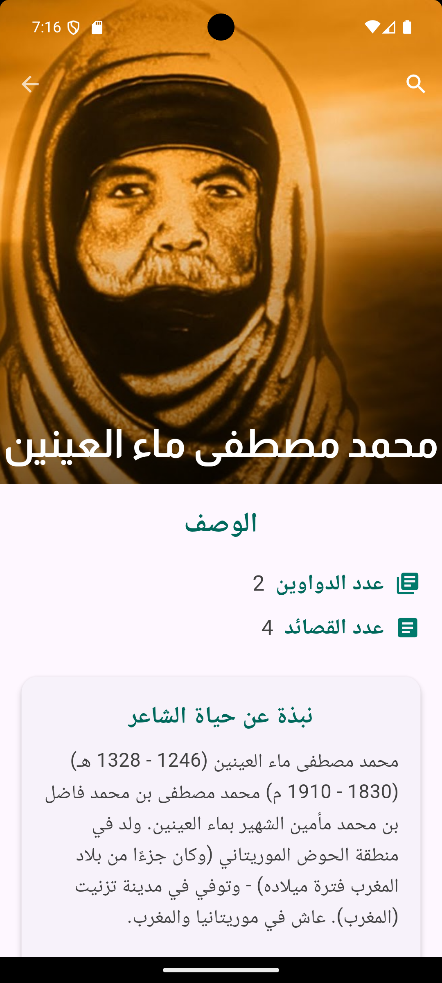
du contenu, permettant aux utilisateurs d'identifier facilement

ce qu'ils recherchent. Les résultats peuvent être triés et filtrés

pour affiner encore plus la recherche.

Figure 17 : recherche résultat

Figure 17 : recherche résultat

*  Page des informations de poète :

La page des informations de poète est conçue pour offrir aux

utilisateurs un aperçu détaillé et complet sur chaque poète de

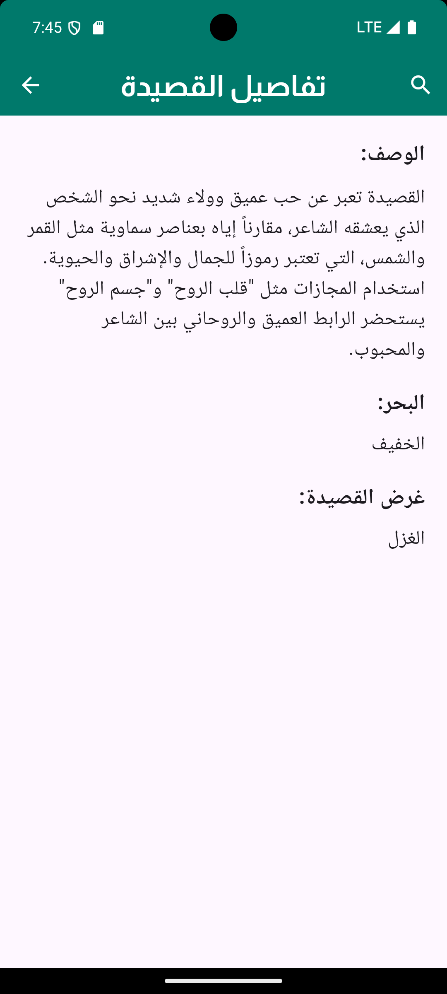
l'application. Elle sert à enrichir l'expérience des utilisateurs en

leur permettant de mieux connaître les auteurs derrière

les poèmes qu'ils apprécient.

Figure 18 : info poète

Figure 18 : info poète

*  Page des informations de poème :

La page des informations de poème est conçue pour offrir aux

utilisateurs un aperçu détaillé et complet sur chaque poème de

l'application. Elle sert à enrichir l'expérience des utilisateurs en

leur permettant de mieux connaître les poèmes qu'ils apprécient.

Figure 19 : info poème

Figure 18 : info poète

**Conclusion :**

En conclusion, ce projet de gestion des demandes de glace pour une Fédération des fabriques de glace a permis de développer une application robuste et sécurisée, répondant aux besoins spécifiques des clients et des propriétaires de fabriques. En tant que stagiaires dans le bureau d'études, nous avons appliqué nos compétences en développement logiciel pour créer une solution qui facilite la création et la gestion des demandes. L'application mobile permet aux clients de soumettre des demandes en toute sécurité grâce à un système d'authentification par email et mot de passe. Parallèlement, la partie administrateur offre une interface intuitive pour gérer les opérations CRUD sur l'ensemble de la base de données, garantissant une gestion efficace des utilisateurs et des demandes.

L'utilisation de technologies modernes et la mise en œuvre de bonnes pratiques de conception ont assuré la fiabilité et la convivialité de l'application. Les relations bien définies entre les différentes entités (clients, propriétaires, demandes) permettent une gestion fluide des données et une meilleure expérience utilisateur. Ce projet a également renforcé notre compréhension des systèmes de gestion de base de données et des architectures logicielles.

Nous remercions notre encadreur et la Fédération des fabriques de glace pour leur soutien tout au long de ce projet. Cette expérience a été enrichissante et formatrice, nous préparant à relever de futurs défis professionnels avec confiance et compétence.

Les références :

[1] <https://code.visualstudio.com/>

[2] <https://androidayuda.com/fr/studio-android/>

[3] <https://www.mongodb.com/fr-fr/company/what-is-mongodb>

[4] <https://www.developpez.com/actu/43205/Modelio-passe-en-version-2-1-1-l-outil-de-modelisation-augmente-son-ouverture-a-l-open-source/>

[5] <https://fr.wikipedia.org/wiki/Dart_(langage)>

[6] <https://kinsta.com/fr/base-de-connaissances/base-de-connaissances-github/>