

Département Mathématiques et Informatiques

**Rapport de Stage de Fin d’Etudes**

Pour l’obtention du titre

**Master de l’ENSET de Mohammedia**

Filière :

**Ingénierie Informatique Big Data et Cloud Computing**

**IIBDCC**

Développement d’un module ERP SaaS

En utilisant Flutter et Firebase

Lieu du stage: **WE ARE beebay**



Réalisé par :

+ **Zakaria El moumnaoui**

Encadré par :

Encadrants Entreprise :

+ **Mustapha Jabrane**

+ **Said EL Haddati**

Encadrant Académique:

+ **Pr. Mohammed Youssfi**

Année Universitaire :

**2023/2024**

Table des matières

[Dédicace 3](#_Toc166509536)

[Remerciements 4](#_Toc166509537)

[Résumé 5](#_Toc166509538)

[Abstract 6](#_Toc166509539)

[Table des figures 7](#_Toc166509540)

[Table des acronymes 8](#_Toc166509541)

[I. Introduction générale 9](#_Toc166509542)

[II. Chapitre 1 : Contexte générale 11](#_Toc166509543)

[1. Introduction 11](#_Toc166509544)

[2. Présentation de l’organisme d’accueil 11](#_Toc166509545)

[a) Présentation Générale 11](#_Toc166509546)

[b) Secteur d’activité 12](#_Toc166509547)

[c) Présentation de l'équipe à refaire 13](#_Toc166509548)

[d) Palette technologique à refaire 13](#_Toc166509549)

[e) Domaines d’expertises 14](#_Toc166509550)

[f) Client de l’entreprise 14](#_Toc166509551)

[3. Contexte du projet 15](#_Toc166509552)

[a) Vue d'ensemble de ERP Saas 15](#_Toc166509553)

[b) Flutter et Firebase 15](#_Toc166509554)

[c) Problématique 18](#_Toc166509555)

[d) Objectif 19](#_Toc166509556)

[4. Gestion du projet 19](#_Toc166509557)

[III. Chapitre 2 : Analyse initiale et exigence 21](#_Toc166509558)

[1. Introduction 21](#_Toc166509559)

[2. Développement Cross plateforme 22](#_Toc166509560)

[a) Adoption de la solution Cloud SaaS 22](#_Toc166509561)

[b) Architecture Flutter 22](#_Toc166509562)

[c) Pourquoi utiliser Firebase dans le Backend 22](#_Toc166509563)

[3. Conclusion 23](#_Toc166509564)

[IV. Chapitre 3 : Analyse technique et conception 24](#_Toc166509565)

[1. Introduction 24](#_Toc166509566)

[2. Phase d’analyse 25](#_Toc166509567)

[3. Phase fonctionnelle 26](#_Toc166509568)

[a. Etude fonctionnelle 26](#_Toc166509569)

[b. Etude technique 26](#_Toc166509570)

[4. Conclusion 27](#_Toc166509571)

[V. Chapitre 4 : Mise en œuvre 28](#_Toc166509572)

[1. Introduction 28](#_Toc166509573)

[2. Stack technique 29](#_Toc166509574)

[3. Etapes de réalisation 30](#_Toc166509575)

[4. Conclusion 31](#_Toc166509576)

[Conclusion générale : 32](#_Toc166509577)

[Webographie : 33](#_Toc166509578)

[Bibliographie 34](#_Toc166509579)

## Dédicace

Cette dédicace est un témoignage sincère de reconnaissance envers ceux qui ont joué un rôle crucial dans la réalisation de ce projet. Nous tenons à exprimer notre gratitude envers Dieu, le Tout-Puissant et Miséricordieux, pour Sa guidance bienveillante et la persévérance qu'Il nous a accordée.

À nos proches, nous adressons un immense merci pour leur patience infinie, leur soutien indéfectible et l'amour qui nous ont constamment inspirés à donner le meilleur de nous-mêmes.

Nous ne pouvons oublier nos mentors académiques, dont la sagesse et l'expérience partagées ont éclairé notre chemin tout au long de notre parcours universitaire. Leurs conseils précieux ont été une boussole dans les moments de doute et d'incertitude.

À nos collaborateurs en entreprise, nous exprimons notre profonde reconnaissance pour leur guidance dans le dédale complexe du monde professionnel. Leurs conseils éclairés et leur collaboration ont été essentiels pour atteindre nos objectifs.

À nos compagnons de route, nous disons merci pour leur soutien constant et leurs encouragements qui ont été des piliers solides tout au long de ce voyage. Leurs encouragements nous ont motivés à persévérer même lorsque les défis semblaient insurmontables.

Enfin, à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, nous vous adressons nos plus sincères remerciements. Vos contributions ont été précieuses et ont permis de donner vie à ce projet modeste mais significatif.

Que cette dédicace soit le témoignage humble de notre gratitude envers ceux qui ont rendu cette réalisation possible. Que nos mots expriment la profondeur de notre reconnaissance et l'ampleur de notre gratitude envers tous ceux qui ont été des partenaires précieux dans cette entreprise.

## Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude envers ceux qui ont joué un rôle essentiel dans l'élaboration de ce mémoire de recherche. Leur soutien et leurs conseils ont été inestimables tout au long de ce parcours.

Je souhaite exprimer ma reconnaissance à mes encadrants internes, Monsieur YOUSSFI Mohamed, ainsi qu'à mes encadrants de l’entreprise We Are Beebay, Monsieur EL HADDATI Said et Monsieur JABRANE Mustapha. Leurs orientations éclairées, leur disponibilité et leur expertise ont grandement contribué à l'aboutissement de ce travail. Leur guidance précieuse a été un moteur essentiel dans sa réalisation.

Je tiens également à remercier l'ensemble du corps professoral de l’École Normale Supérieure de l’Enseignement Technique de Mohammedia. Leurs méthodes pédagogiques innovantes m'ont permis de développer et d'appliquer mes connaissances de manière efficace.

Je n'oublie pas de mentionner le personnel technique de la société We Are Beebay, dont l'engagement et la disponibilité ont été remarquables tout au long de mon stage. Leur support constant a été d'une valeur inestimable.

Enfin, mes remerciements s'adressent à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail. Votre soutien et vos encouragements ont été une source de motivation constante.

## Résumé

## Abstract

## Table des figures

## Table des acronymes

## Introduction générale

Dans le domaine de la technologie, les solutions ERP SaaS (Enterprise Resource Planning Software as à Service) occupent une place de plus en plus centrale dans la gestion des entreprises modernes. Ces solutions, hébergées dans le cloud, offrent une approche innovante pour intégrer et gérer efficacement les ressources d'une organisation, couvrant un large éventail de fonctions allant de la gestion financière à la logistique en passant par les ressources humaines.

L'utilité de cette dernière réside dans la capacité à centraliser et automatiser les processus métier, offrant ainsi une vue d'ensemble complète des opérations de l'entreprise. En simplifiant ainsi la gestion des ressources et en améliorant l'efficacité opérationnelle, ces solutions permettent aux entreprises de rester compétitives dans un environnement commercial en constante évolution.

Notamment, l'intégration de technologies Flutter et Firebase dans le développement d'applications ERP SaaS offre un avantage significatif. Flutter, avec son framework de développement d'interfaces utilisateur réactives et dynamiques, permet de créer des expériences utilisateur fluides et homogènes sur toutes les plateformes, qu'il s'agisse de mobiles, de desktop ou du web. D'autre part, Firebase fournit une infrastructure cloud robuste et évolutive, offrant une gamme de services tels que l'authentification des utilisateurs, la base de données en temps réel et le stockage cloud, qui sont essentiels pour le fonctionnement efficace des applications ERP SaaS. En combinant la puissance de Flutter pour la création d'interfaces utilisateur modernes et réactives avec les fonctionnalités riches de Firebase pour la gestion des données et des services cloud, les développeurs peuvent créer des applications ERP SaaS hautement performantes et évolutives.

Cette approche permet non seulement d'améliorer l'expérience utilisateur, mais également de garantir la fiabilité, la sécurité et l'évolutivité des applications, répondant ainsi aux besoins croissants des entreprises en matière de gestion des ressources et d'optimisation des processus métier.

Le rapport entame le premier chapitre qui concerne le cadre global du projet commençant par une introduction de l’entité hôte, une explication détaillée et la méthodologie de la gestion de projet.

Le deuxième chapitre traite l’aspect SaaS dans le développement des logiciels, exhibant Flutter et Firebase comme des solutions utilise pour développer les modules SaaS

Le troisième chapitre explore la conception et la modélisation de l’application ainsi développée, en expliquant comment aboutir ces deux points dans un contexte SaaS.

Le quatrième chapitre couvre la parité de la réalisation du projet en particulier l’étude technique, incluant l’architecture du système et les différentes technologies ainsi que mentionner les différents sprints d la mise en œuvre du système.

En finale, la dernière partie est dédiée a la présentation des résultats obtenus et aux perspectives du projet.

# Chapitre 1 : Contexte générale

## Introduction

Ce chapitre représente une mise dans le contexte du projet, il met le point en premier

lieu sur l’entreprise d’accueil, ensuite il expose la conduite du projet incluant la méthodologie de travail et la planification du stage.

## Présentation de l’organisme d’accueil

### Présentation Générale

**We Are BEEBAY** est une société spécialisée dans les technologies de l'information, se concentrant sur le conseil et la mise en œuvre des plateformes d'intégration, avec une expertise particulière dans les Hybrid Integration Platforms (HIP) avec des clients principalement basés en Europe et au Moyen Orient depuis plus de 10 ans d’après leurs bureaux à Paris et Casablanca.



Figure 1 : Logo de We Are beebay

### Secteur d’activité

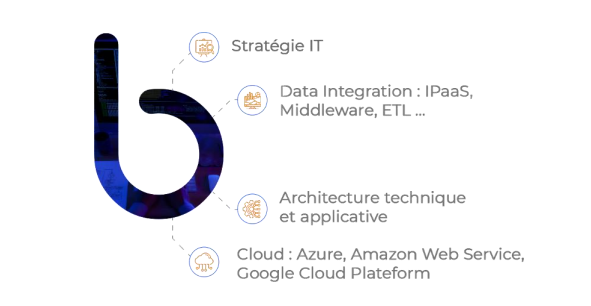


Figure 2: Sécteur d'activité de We Are Beebay

L’activité de We Are Beebay se base sur 2 axes principaux :

* *Conseil en architecture des systèmes distribués et technologies Cloud :*

We Are Beebay offre des services de consultation spécialisés dans la conception et l'optimisation des architectures de systèmes distribués, en mettant l'accent sur les technologies cloud. Cela comprend l'évaluation des besoins de l'entreprise, la conception d'architectures adaptées, la sélection des technologies cloud appropriées et la mise en place de stratégies d'intégration efficaces. Leur expertise permet aux clients de bénéficier d'infrastructures robustes et évolutives, alignées sur leurs objectifs métier et optimisées pour une utilisation efficace des ressources cloud.

* *Conception et implémentation de plateformes d'intégration et plateformes d'intégration hybrides :*

We Are BEEBAY excelle dans la conception et l'implémentation de plateformes d'intégration, y compris les plateformes d'intégration hybrides. Cela comprend la mise en place de solutions telles que IPaaS (Integration Platform as a Service), la gestion d'API, les middlewares orientés messages, etc. Leur approche holistique englobe la compréhension des besoins spécifiques de l'entreprise, la sélection des technologies appropriées, le développement sur mesure et la mise en œuvre, ainsi que l'intégration avec les systèmes existants. En offrant des solutions d'intégration flexibles et robustes, BEEBAY aide les entreprises à optimiser leurs flux de données et à améliorer leur agilité opérationnelle.

### Présentation de l'équipe à refaire



La présentation de l'équipe de We Are Beebay est l'occasion de mettre en lumière les talents qui constituent le cœur de l'entreprise. C'est une équipe diversifiée de professionnels passionnés, chacun apportant son expertise unique pour enrichir la qualité des

services offerts. Ensemble, ils incarnent les valeurs de We Are Beebay et travaillent de concert pour délivrer des solutions innovantes en intégration de données, reflétant un engagement envers l'excellence et la satisfaction client.

### Palette technologique





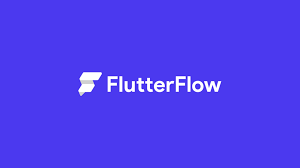








Figure 3 : Les technologies d’expertise

We Are BEEBAY excelle dans la maîtrise d'un éventail varié de technologies de pointe, ce qui lui permet de répondre efficacement aux besoins complexes de ses clients en matière de développement et d'architecture informatique.

Parmi ces technologies figurent des solutions leaders du marché telles que Flutter, Firebase, et FlutterFlow, ainsi que des plateformes Cloud de renommée mondiale comme AWS, Google Cloud Platform et Azure.

De plus, BEEBAY possède une expertise avancée dans des frameworks spécialisés tels que Spring, offrant ainsi à ses clients une gamme complète de solutions pour leurs défis de développement.

Cette diversité de compétences technologiques permet à We Are BEEBAY de concevoir et de mettre en œuvre des architectures informatiques sur mesure, répondant aux exigences spécifiques de chaque entreprise et favorisant un développement avancé des systèmes et des applications.

### Domaines d’expertises

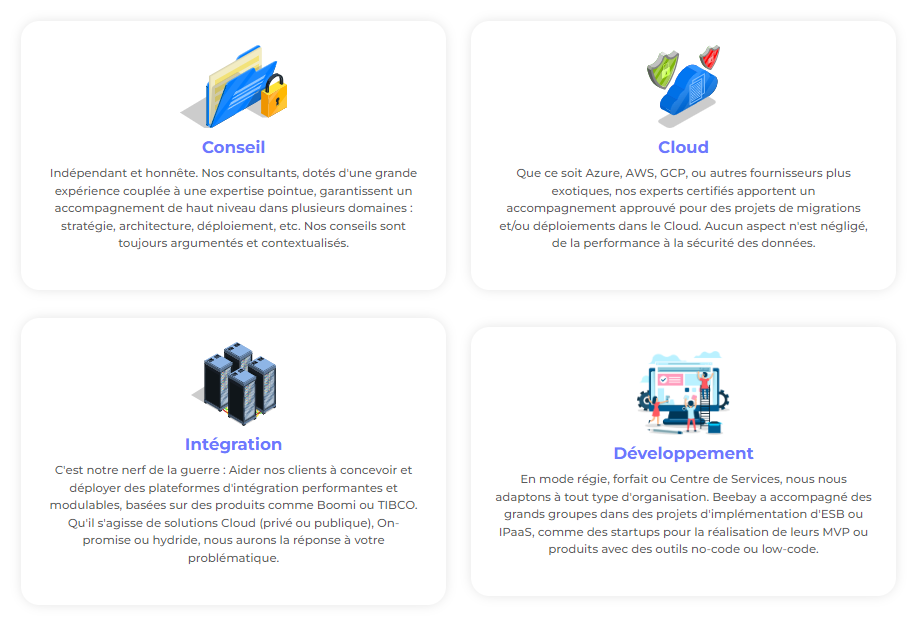


Figure 4 : Domaines d'expertises

### Client de l’entreprise



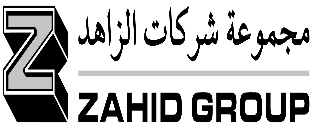


Figure 5 : Clients de l’entreprise

We Are Beebay coopère avec des clients divers entre des start-up innovantes aux grandes entreprises internationales, montrant leur capacité à fournir des solutions innovantes pour différents secteurs et besoins.

## Contexte du projet

### Vue d'ensemble de ERP SaaS

Dans le paysage dynamique de la gestion d'entreprise moderne, les solutions ERP SaaS (Enterprise Resource Planning Software as a Service) émergent comme des éléments fondamentaux pour les organisations cherchant à optimiser leurs opérations, à accroître leur efficacité et à rester compétitives sur un marché en constante évolution. Ces solutions, hébergées dans le cloud, offrent une approche novatrice pour la gestion intégrée des ressources, couvrant un large éventail de fonctions allant de la comptabilité à la gestion des ressources humaines.

En résumé, l'utilisation d'une solution ERP SaaS offre aux entreprises une approche complète pour la gestion efficace de leurs ressources, en offrant des fonctionnalités avancées pour automatiser les processus opérationnels et améliorer leur agilité.

### Flutter et Firebase

* 1. Dart

Dart est un langage de programmation moderne polyvalent et de haut niveau développé à l'origine par Google. Il est apparu en 2011 comme nouveau langage de programmation, mais sa version stable n'a été publiée qu'en juin 2017. À ses débuts, Dart n'était pas très populaire, mais il a gagné en popularité lorsqu'il a été utilisé avec Flutter.

Dart est un langage de programmation dynamique, orienté objet et basé sur les classes, avec des fermetures et une portée lexicale. Sur le plan syntaxique, il est assez similaire à Java, C et JavaScript. Si vous connaissez l'un de ces langages de programmation, vous pouvez facilement apprendre le langage de programmation Dart.

Dart est un langage de programmation open source largement utilisé pour développer des applications mobiles, des applications web modernes, des applications de bureau et l'Internet des objets (IoT) en utilisant le framework Flutter. Il prend également en charge quelques concepts avancés tels que les interfaces, les mixins, les classes abstraites, les generics réfléchis et l'inférence de types. C'est un langage compilé qui prend en charge deux types de techniques de compilation.

**AOT (Ahead of Time)** - Il convertit le code Dart en code JavaScript optimisé à l'aide du compilateur dar2js et s'exécute sur tous les navigateurs web modernes. Il compile le code au moment de la construction.

**JIT (Just-In-Time)** - Il convertit le bytecode en code machine (code natif), mais seulement le code qui est nécessaire.

**Avant d'apprendre Dart, il est important de garder à l'esprit ces concepts :**

1. Tout en Dart est traité comme un objet, y compris les nombres, les booléens, les fonctions, etc., comme en Python. Tous les objets héritent de la classe Object.

2. Les outils Dart peuvent signaler deux types de problèmes lors de la programmation : les avertissements et les erreurs. Les avertissements indiquent qu'il peut y avoir un problème dans votre code, mais cela n'interrompt pas l'exécution du code, tandis que les erreurs peuvent empêcher l'exécution du code.

3. Dart prend en charge le typage fort.

4. Dart prend en charge les types génériques, comme List<int> (une liste d'entiers) ou List<dynamic> (une liste d'objets de n'importe quel type).

* 1. Flutter

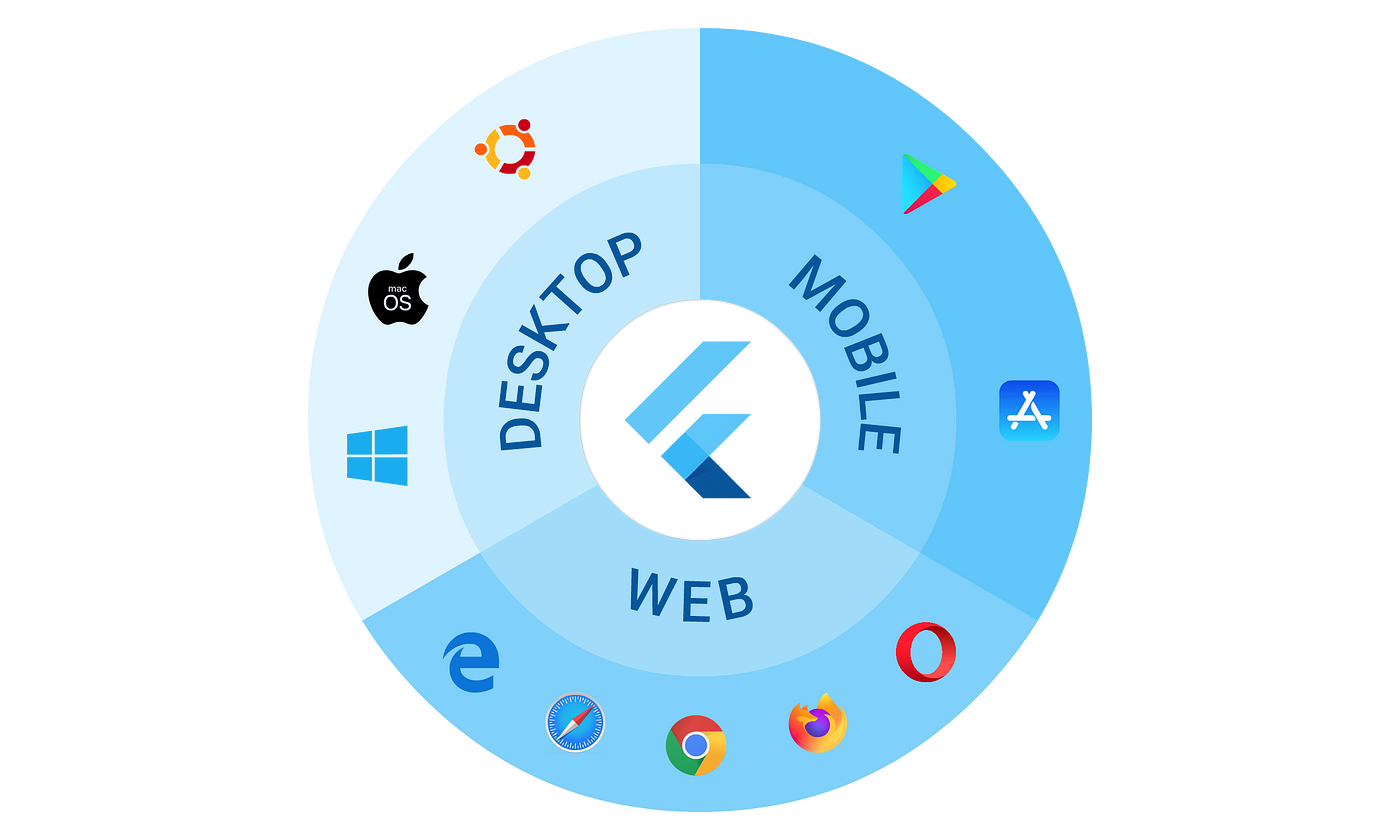


Figure 5 : Flutter cross-platform promises: mobile, web, and desktop

Flutter est un framework open-source et polyvalent pour créer des applications mobiles et web hautes performances. Développé par Google, le développement d’applications Flutter permet aux développeurs d’utiliser une seule base de code pour créer des applications pour iOS et Android, ainsi que des plateformes web et de bureau. Il utilise un modèle de programmation réactif pour construire des interfaces utilisateur, ce qui permet aux développeurs de créer facilement des interfaces belles et interactives.

Flutter framework se distingue par son SDK qui offre un style réactif, sans compter sur un pont JavaScript. Cette approche permet à Flutter d’atteindre des niveaux de performance comparables à ceux de son concurrent direct, React Native. Flutter s’intègre également de manière transparente à diverses plateformes, notamment les applications Android, iOS, Linux, macOS, Windows et Google Fuchsia.

Il dispose d’une communauté intéressante de développeurs qui contribuent à son développement et créent des paquets qui peuvent être facilement intégrés dans une application Flutter. Il s’agit d’un excellent choix pour les développeurs qui veulent créer des applications multi-plateformes visuellement superbes avec un processus de développement rapide.

* 1. Firebase

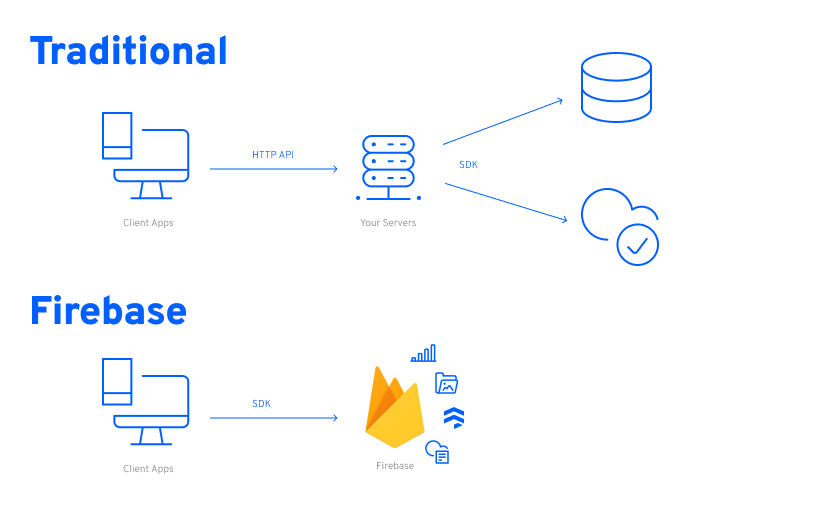


Figure 5 : Firebase Backend

Firebase est une suite intégrée d’outils de développement qui permettent aux utilisateurs de créer de puissantes applications web et mobiles. Maintenu par Google et exécuté sur l’infrastructure cloud de Google, Firebase interagit directement avec les services backend, contournant le besoin d’un middleware entre l’application et le service.

En tant que sous-ensemble organisé de la plateforme Google Cloud (GCP), Firebase rend plusieurs services Google plus accessibles pour les développeurs d’applications. La console Firebase est beaucoup plus concentrée et intuitive que celle de GCP. Et le kit de développement logiciel (SDK) bien documenté – un ensemble d’outils, de bibliothèques, de documentation, d’exemples de code et de guides que les développeurs téléchargent directement et suivent – connecte votre application à ces services avec un effort de développement minimal.

De plus, l’ensemble d’outils intègre une grande partie des services que les développeurs doivent normalement configurer manuellement, comme l’authentification, la configuration, le stockage de fichiers et les analyses. S’appuyer sur l’infrastructure Google permet aux utilisateurs (et à nos clients) de savoir avec certitude que les applications créées avec Firebase sont bien conçues, fiables et évolutives.

### Problématique

À l'ère numérique, le développement de plateformes web et d'applications mobiles est devenu un élément clé de la stratégie commerciale de nombreuses entreprises. Ces plateformes doivent être à la fois performantes, sécurisées et évolutives pour répondre aux besoins changeants des utilisateurs et aux défis technologiques émergents.

Dans ce contexte dynamique, le besoin de rapidité de développement est crucial pour rester compétitif sur le marché. C'est là que des outils et des technologies modernes comme Flutter et Firebase entrent en jeu. Flutter, un framework de développement d'interfaces utilisateur multiplateforme, offre une approche rapide et efficace pour la création d'applications web et mobiles. Grâce à son architecture basée sur des widgets réactifs, Flutter permet aux développeurs de concevoir des interfaces utilisateur fluides et attrayantes tout en réduisant le temps de développement.

D'autre part, Firebase, une plateforme de développement d'applications mobiles et web de Google, fournit une suite complète de services cloud, y compris l'authentification des utilisateurs, la base de données en temps réel, le stockage de fichiers, et bien plus encore. L'intégration de Firebase dans le processus de développement permet aux développeurs de gagner du temps en évitant la mise en place d'une infrastructure backend complexe et en se concentrant plutôt sur la création de fonctionnalités clés pour l'application.

Cependant, malgré les avantages de rapidité offerts par Flutter et Firebase, les développeurs sont toujours confrontés à des défis complexes liés à la conception et à l'implémentation d'une architecture logicielle robuste et flexible. Ils doivent trouver des stratégies efficaces pour intégrer harmonieusement ces technologies tout en garantissant la réutilisabilité du code et la maintenabilité du système. De plus, assurer des performances optimales et la sécurité des applications reste une priorité majeure pour garantir une expérience utilisateur fluide et sécurisée.

Dans ce contexte, les questions clés qui se posent sont les suivantes :

* Comment concevoir une architecture logicielle flexible et évolutive pour une plateforme web ou une application, en tirant parti des avantages de Flutter et Firebase tout en respectant les principes de conception logicielle et les meilleures pratiques de développement ?
* Quelles sont les stratégies efficaces pour intégrer de manière modulaire Flutter et Firebase dans une plateforme web ou une application, en favorisant la factorisation du code et la maintenabilité du système ?
* Comment assurer des performances pour l'application, en mettant en œuvre des techniques d'optimisation du code, de gestion des ressources et de mise en cache des données ?
* Quels sont les enjeux majeurs liés à la sécurité des logiciels et à la protection des données, et quelles approches peuvent être adoptées pour identifier les risques de sécurité potentiels, tout en garantissant la conformité aux normes de sécurité et de confidentialité ?

### Objectif

Ce projet permet de répondre aux objectifs ci-dessous liés aux problématiques mentionnées :

1.Mettre en place une architecture logicielle flexible et intégrée pour les applications web et mobiles, en adaptant les technologies Flutter et Firebase en fonction des besoins spécifiques de chaque plateforme.

2.Utiliser des méthodes de développement efficaces qui utilisent les capacités de Flutter et Firebase, tout en garantissant la qualité du code, la continuité du système et la réutilisabilité des composants.

3. Améliorer les performances de l'application en utilisant des techniques avancées d'optimisation , de gestion des ressources et de mise en cache des données, pour garantir une expérience utilisateur fluide et réactive.

4. Implémenter des mesures de sécurité robustes pour préserver la confidentialité des données utilisateurs, en respectant les normes et réglementations en vigueur, et en anticipant les risques de piratage ou de violation de la vie privée.

## Gestion du projet

1. Ressources Humaines

Dans l'ensemble, la puissance de l'équipe de projet réside dans l'alliance stratégique entre les encadrants internes de l'ENSET Mohammédia et les encadrants externes de We Are Beebay, comme en témoigne notre tableau de référence. Cette collaboration dynamique et polyvalente est essentielle pour mener à bien et réussir le projet.

|  |  |  |
| --- | --- | --- |
| Pr. Youssfi Mohamed | Prof a ENSET Mohammédia | Encadrant interne |
| Mr. El haddati Said | Directeur Technique | Encadrant externe |
| Mr. Jabrane Mustapha | Technical Leader | Encadrant externe |

Figure 5 : L'équipe du projet

# Chapitre 2 : Analyse initiale et exigence

## Introduction

Ce chapitre commence par explorer les solutions utilisées pour développer une application SaaS, en examinant l'analyse initiale et les exigences fondamentales. Au cœur de cette démarche se trouvent Flutter et Firebase, deux piliers technologiques qui ont façonné notre approche de développement. Explorerons en détail cette étape préliminaire, où chaque décision ouvre la voie à une application solide et performante.

## Développement Cross plateforme

### Adoption de la solution Cloud SaaS

Le logiciel en tant que service (SaaS) est une offre de cloud computing qui donne aux utilisateurs l'accès à un logiciel basé sur le cloud d'un fournisseur. Quels sont les principaux avantages de l'utilisation du SaaS ?

Ce tableau résume les principaux avantages de l'utilisation des solutions SaaS dans le développement et la gestion des applications.

|  |  |
| --- | --- |
| Avantage | Description |
| Réduction du délai de rentabilisation | Contrairement au modèle traditionnel, le logiciel en tant que service (SaaS) est déjà installé et configuré. Il suffit de provisionner un serveur pour une instance dans le cloud, et en quelques heures, l'application est prête à l'emploi. Cela réduit le temps consacré à l'installation et à la configuration, et peut minimiser les problèmes qui entravent le déploiement du logiciel. |
| Coûts réduits | Le SaaS peut offrir des économies de coûts intéressantes car il réside  généralement dans un environnement partagé ou multi-tenant, où les coûts des licences de matériel et de logiciels sont faibles par rapport au modèle traditionnel. Un autre avantage est que vous pouvez rapidement étendre votre clientèle puisque le SaaS permet aux petites et moyennes entreprises d'utiliser un logiciel qu'elles n'utiliseraient pas autrement en raison du coût élevé des licences. Les coûts de maintenance sont également réduits, car l'environnement appartient au fournisseur SaaS et  est réparti entre tous les clients qui utilisent cette solution. |
| Évolutivité et intégration | Les solutions SaaS résident généralement dans des environnements  cloud évolutifs et s'intègrent à d'autres offres SaaS. Par rapport au  modèle traditionnel, vous n'avez pas besoin d'acheter un autre serveur  ou logiciel. Il vous suffit d'activer une nouvelle offre SaaS et le fournisseur  SaaS s'occupe de la planification de la capacité du serveur. Vous avez  également la flexibilité d'augmenter ou de diminuer votre utilisation du  SaaS en fonction de vos besoins spécifiques. |
| Nouvelles versions (mises à niveau) | Avec le SaaS, le fournisseur met à niveau la solution et elle devient disponible pour ses clients. Les coûts et les efforts associés aux mises à niveau et aux nouvelles versions sont inférieurs à ceux du modèle traditionnel qui vous oblige généralement à acheter un package de mise à niveau et à l'installer (ou à payer des services spécialisés pour mettre à niveau l'environnement). |
| Facilité d'utilisation et réalisation de preuves de concept | Les offres SaaS sont faciles à utiliser car elles intègrent déjà les meilleures pratiques et des exemples. Les utilisateurs peuvent réaliser des preuves de concept et tester la fonctionnalité du logiciel ou une nouvelle fonctionnalité avant son lancement. De plus, vous pouvez avoir plusieurs instances avec des versions différentes et effectuer une migration en douceur. Même pour les environnements importants, vous pouvez utiliser des offres SaaS pour tester le logiciel avant de l'acheter. |

### Architecture Flutter

Flutter est conçu comme un système extensible et en couches. Il se présente sous la forme d'une série de bibliothèques indépendantes, chacune dépendant de la couche sous-jacente. Aucune couche n'a un accès privilégié à la couche inférieure, et chaque partie du niveau du Framework est conçue pour être optionnelle et remplaçable.

Cette figure illustre les éléments essentiels de flutter :

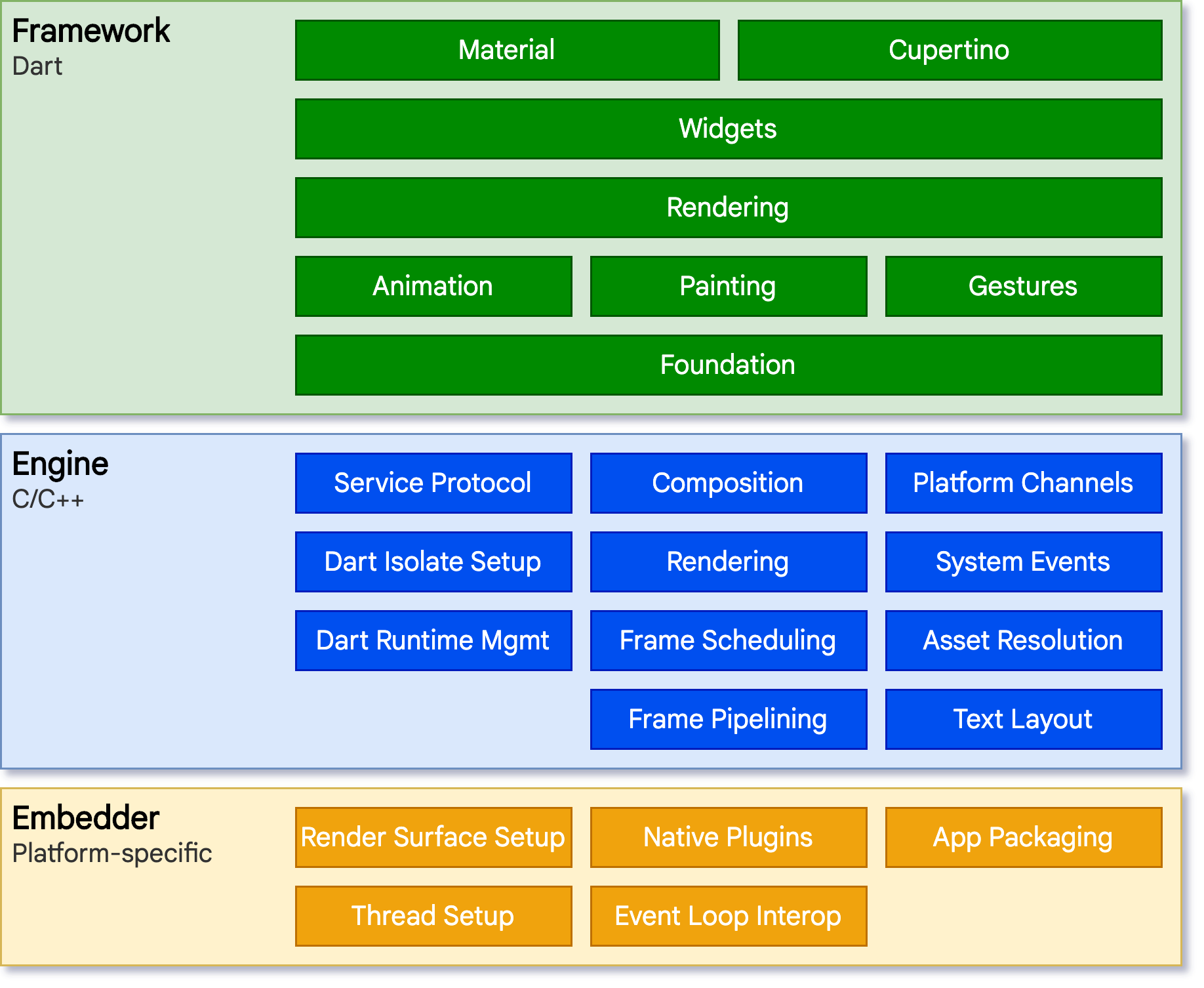


Figure 6 : architecture Flutter

Pour le système d'exploitation sous-jacent, les applications Flutter sont empaquetées de la même manière que n'importe quelle autre application native. Un lanceur spécifique à la plateforme fournit un point d'entrée ; il coordonne avec le système d'exploitation sous-jacent pour l'accès à des services tels que les surfaces de rendu, l'accessibilité et les entrées ; et gère la boucle d'événements de messages. Le lanceur est écrit dans un langage approprié à la plateforme : actuellement Java et C++ pour Android, Objective-C/Objective-C++ pour iOS et macOS, et C++ pour Windows et Linux. En utilisant le lanceur, le code Flutter peut être intégré dans une application existante en tant que module, ou le code peut constituer l'intégralité du contenu de l'application. Flutter inclut un certain nombre de lanceurs pour les plates-formes cibles courantes, mais d'autres lanceurs existent également.

Au cœur de Flutter se trouve le moteur Flutter, principalement écrit en C++, qui prend en charge les primitives nécessaires pour tous les types d'applications Flutter. Le moteur est responsable de la rastérisation des scènes composites chaque fois qu'une nouvelle image doit être peinte. Il assure la mise en œuvre de bas niveau de l'API principale de Flutter, notamment les graphiques (à travers Impeller sur iOS et à venir sur Android, et Skia sur d'autres plates-formes), la mise en page de texte, les E/S de fichiers et de réseau, la prise en charge de l'accessibilité, l'architecture des plugins, ainsi qu'une chaîne d'outils d'exécution et de compilation Dart.

Le moteur est exposé au Framework Flutter via **dart:ui**, qui enveloppe le code C++ sous-jacent dans des classes Dart. Cette bibliothèque expose les primitives de plus bas niveau, telles que les classes pour piloter les entrées, les graphiques et les sous-systèmes de rendu de texte.

Typiquement, les développeurs interagissent avec Flutter qui fournit un Framework moderne et réactif écrit dans le langage Dart. Il inclut un ensemble riche de bibliothèques de plateforme, de mise en page et fondamentales, composées d'une série de couches. En partant du bas vers le haut, nous avons :

Les bibliothèques **Material** et **Cupertino** offrent des ensembles complets de contrôles qui utilisent les primitives de composition de la couche de widgets pour implémenter les langages de conception Material ou iOS

Des classes fondamentales de base et des services de blocs de construction tels que l'animation, le dessin et les gestes qui offrent des abstractions couramment utilisées sur la fondation sous-jacente

**Flutter**

La couche de widgets est une abstraction de composition. Chaque objet de rendu dans la couche de rendu a une classe correspondante dans la couche de widgets. De plus, la couche de widgets vous permet de définir des combinaisons de classes que vous pouvez réutiliser. C'est à cette couche que le modèle de programmation réactif est introduit

La couche de rendu fournit une abstraction pour traiter la mise en page. Avec cette couche, vous pouvez construire un arbre d'objets rendables. Vous pouvez manipuler ces objets dynamiquement, l'arbre mettant automatiquement à jour la mise en page pour refléter vos modifications.

Le Framework Flutter est relativement petit; de nombreuses fonctionnalités de niveau supérieur que les développeurs pourraient utiliser sont implémentées sous forme de packages, y compris les plugins de plateforme comme la caméra et le webview, ainsi que des fonctionnalités agnostiques à la plateforme comme les caractères, http, et les animations qui s'appuient sur les bibliothèques Dart et Flutter de base. Certains de ces packages proviennent de l'écosystème plus large, couvrant des services tels que les paiements in-app, l'authentification Apple et les animations.

Le reste de cet aperçu navigue largement à travers les couches, en commençant par le paradigme réactif du développement d'interface utilisateur. Ensuite, nous décrivons comment les widgets sont composés ensemble et convertis en objets pouvant être rendus dans le cadre d'une application. Nous décrivons comment Flutter interagit avec d'autres codes au niveau de la plateforme, avant de donner un bref résumé de la façon dont le support Web de Flutter diffère des autres cibles.

### Firebase pour le Backend

Utiliser Firebase comme backend pour le développement d'une application web présente plusieurs avantages, particulièrement en conjonction avec une architecture SaaS. Voici les principales raisons pour lesquelles Firebase est une excellente option :

|  |  |
| --- | --- |
| Raison | Description |
| Facilité d'utilisation | Firebase propose une gamme de services prêts à l'emploi qui simplifient le développement, comme Firestore pour la gestion de base de données en temps réel, Authentication pour la gestion des utilisateurs, et Hosting pour le déploiement d'applications web. Cela permet de démarrer rapidement sans voir à configurer une infrastructure complexe. |
| Scalabilité | Firebase est conçu pour évoluer automatiquement avec la demande.  Que vous ayez quelques utilisateurs ou des millions, Firebase s'adapte pour gérer le trafic sans nécessiter d'interventions manuelles pour la mise à l'échelle. |
| Intégration avec Flutter | Firebase offre une excellente intégration avec Flutter, facilitant la création d'applications cross-platforms avec une seule base de code. Cela permet de partager des fonctionnalités entre les versions web, iOS et Android de l'application. |
| Sécurité | Firebase propose des fonctionnalités de sécurité robustes comme Firebase  Authentication pour sécuriser l'accès utilisateur et Firebase Security Rules  pour protéger les données dans Firestore et Realtime Database. Cela assure une protection élevée des données des utilisateurs. |
| Services en Temps Réel | Firestore et Realtime Database de Firebase permettent de synchroniser  les données en temps réel, offrant une expérience utilisateur fluide et  réactive. Cela est particulièrement utile pour les applications nécessitant  des mises à jour en temps réel comme les messageries instantanées ou  les tableaux de bord collaboratifs. |
| Analytique et Surveillance | Firebase Analytics fournit des insights détaillés sur l'utilisation de l'application, permettant d'optimiser l'expérience utilisateur. De plus, Firebase Performance Monitoring et Crashlytics aident à surveiller les performances et à identifier les problèmes rapidement. |
| Infrastructure sans serveur (Serverless) | Avec Firebase Functions, vous pouvez exécuter du code backend en réponse à des événements déclenchés par Firebase ou d'autres services Google Cloud, sans gérer de serveurs. Cela simplifie le déploiement et la gestion des services backend. |
| Déploiement et Hébergement | Firebase Hosting offre un déploiement rapide et sécurisé pour les applications web, avec des fonctionnalités comme le SSL automatique et la gestion des versions. Cela simplifie la mise en ligne et la gestion continue des applications. |

Ces avantages font de Firebase une solution backend attrayante pour les développeurs cherchant à créer des applications web modernes et performantes, tout en bénéficiant de l'efficacité et de la scalabilité des services cloud.

## Conclusion

# Chapitre 3 : Analyse technique et conception

## Introduction

## Phase d’analyse

## Phase fonctionnelle

### Etude fonctionnelle

### Etude technique

## Conclusion

# Chapitre 4 : Mise en œuvre

## Introduction

## Stack technique

## Etapes de réalisation

## Conclusion

# Conclusion générale :

# Webographie :

<https://www.javatpoint.com/dart-programming>

<https://docs.flutter.dev/resources/architectural-overview>

<https://www.ibm.com/think/insights/software-as-a-service-advantages#:~:text=SaaS%20can%20provide%20beneficial%20cost,compared%20with%20the%20traditional%20model>.

# Bibliographie