|  |
| --- |
|  |



|  |
| --- |
|  |

**Projet 4°Année: 2023 / 2024**

**Développement d’une application de gestion de dépenses.**

**Ingénierie Informatique et Réseaux**

Réalisé par :

Nom de l’étudiant 1 : Maroua Chegri.

Nom de l’étudiant 2 : Bensalk Ilias Mohamed.

Nom de l’étudiant 3 : Ouahdani Mohamed Taha.

Encadré par :

Nom de l’encadrant : Dr. Driss Essabar & Dr. Sanae Chakri & Dr. Abdeljalil Nadiri & Dr. Ayoub Charef.

# Dédicace

On dédie ce modeste travail :

**A Dieu**

Le tout puissant mon créateur.

Une image contenant écriture manuscrite, croquis, Police, dessin

Description générée automatiquement

Dans la dédicace de ce travail, nous souhaitons honorer nos familles, nos amis, nos collègues et nos mentors, qui ont été des piliers essentiels dans cette aventure. Leur soutien indéfectible, leurs encouragements constants et leurs précieux conseils ont façonné ce projet. Un hommage particulier est dédié à nos professeurs, dont l'enseignement enrichissant a grandement contribué à notre réussite. À chacun d'entre vous, nous exprimons notre profonde reconnaissance pour avoir joué des rôles si importants dans cette réalisation. Merci pour votre soutien indéfectible et vos contributions inestimables.

# Remerciement

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce projet. Leur soutien, leur expertise et leur encouragement ont été essentiels à chaque étape de ce processus.

Tout d'abord, je tiens à remercier Driss Essabar, Ayoub Charef, Sanae Chakri et Nadiri Abdeljalil pour leurs conseils éclairés, leur expertise technique et leur soutien constant. Leurs suggestions précieuses et leur encadrement m'ont permis de comprendre en profondeur les tenants et aboutissants du projet.

Je tiens également à remercier chaleureusement mes collègues et mes camarades de classe pour leurs discussions enrichissantes, leurs idées novatrices et leur coopération tout au long du projet. Leur contribution a grandement enrichi mes réflexions et a favorisé un environnement propice à l'apprentissage collaboratif.

Un grand merci à l'EMSI pour avoir fourni les ressources nécessaires et pour avoir soutenu ce projet depuis ses débuts. Leur engagement envers l'excellence et leur soutien logistique ont grandement facilité sa réalisation.

Je souhaite exprimer ma reconnaissance envers mes proches et ma famille pour leur soutien inconditionnel, leur compréhension et leurs encouragements tout au long de ce projet. Leur soutien moral m'a été d'une grande aide pour surmonter les défis et atteindre mes objectifs.

Enfin, je tiens à remercier tous ceux qui, de près ou de loin, ont contribué à ce projet et à ma formation. Votre soutien et vos conseils ont été inestimables.

# Résume

Le projet consiste en la conception et le développement d'une application de gestion des dépenses. Avant d'entamer la phase de réalisation, une analyse approfondie des objectifs et des fonctionnalités du système a été effectuée afin de définir clairement les services et les solutions offerts aux utilisateurs.

La simulation du système a été réalisée à l'aide de diverses illustrations, telles que des diagrammes, permettant une description précise de sa structure et de son fonctionnement. L'application offre plusieurs fonctionnalités, notamment l'ajout, la modification, la suppression et l'affichage des dépenses.

L'architecture du système est basée sur une application web, développée avec React.js pour le front-end et Node.js pour le back-end. Le stockage des données est géré par une base de données relationnelle, tandis que les requêtes sont traitées via des API RESTful.

L'implémentation de la sécurité dans l'application a été assurée par l'utilisation de JSON Web Tokens (JWT) pour l'authentification des utilisateurs et la protection des routes sensibles.

En résumé, ce projet a permis de créer une application de gestion des dépenses complète et fonctionnelle, offrant aux utilisateurs une interface intuitive pour gérer efficacement leurs finances.

# Abstract

The project involves designing and developing an expense management application. Before embarking on the implementation phase, a thorough analysis of the system's objectives and functionalities was conducted to clearly define the services and solutions offered to users.

The system simulation was performed using various illustrations, such as diagrams, allowing for a precise description of its structure and operation. The application offers several features, including adding, modifying, deleting, and displaying expenses.

The architecture of the system is based on a web application, developed with React.js for the front-end and Node.js for the back-end. Data storage is managed by a relational database, while requests are processed through RESTful APIs.

Security implementation in the application was ensured by using JSON Web Tokens (JWT) for user authentication and protecting sensitive routes.

In summary, this project has created a complete and functional expense management application, providing users with an intuitive interface to effectively manage their finances.

# TABLE DE MATIERE

Table des matières

Table des matières

[Dédicace - 2 -](#_Toc166441611)

[Remerciement - 3 -](#_Toc166441612)

[Résume - 4 -](#_Toc166441613)

[Abstract - 5 -](#_Toc166441614)

[TABLE DE MATIERE - 6 -](#_Toc166441615)

[TABLE DE FIGURE - 8 -](#_Toc166441616)

[Chapitre 1 : Contexte Generale - 10 -](#_Toc166441617)

[1.Introduction : - 1 -](#_Toc166441618)

[2.Presentation du Sujet : - 11 -](#_Toc166441619)

[2.1 Problématique : - 11 -](#_Toc166441620)

[2.2 Objectif : - 11 -](#_Toc166441621)

[3. Planification et Conduite du projet - 12 -](#_Toc166441622)

[3.1 Processus de Développement - 12 -](#_Toc166441623)

[Meet the Team : - 12 -](#_Toc166441624)

[Liens Github : - 12 -](#_Toc166441625)

[3.2 Tableau de Taches : - 13 -](#_Toc166441626)

[Partie : Azure Devops : - 14 -](#_Toc166441627)

[-Product Backlogs - 14 -](#_Toc166441628)

[Taches : - 15 -](#_Toc166441629)

[Boards : - 16 -](#_Toc166441630)

[Sprints : - 17 -](#_Toc166441631)

[3.3 Planification - 18 -](#_Toc166441632)

[Conclusion - 18 -](#_Toc166441633)

[4. Analyse et Conception - 18 -](#_Toc166441634)

[4.1 Introduction - 18 -](#_Toc166441635)

[4.2 Spécifications fonctionnelles - 18 -](#_Toc166441636)

[4.3 Spécifications Non Fonctionnelles : - 19 -](#_Toc166441637)

[4.4 La Méthode UML : - 20 -](#_Toc166441638)

[Conclusion - 22 -](#_Toc166441639)

[5.Realisation Et Outils Utilises - 22 -](#_Toc166441640)

[Introduction - 22 -](#_Toc166441641)

[5.1 Outils de Développement - 23 -](#_Toc166441642)

[6.Architecture Globale - 24 -](#_Toc166441643)

[6.1 Introduction - 24 -](#_Toc166441644)

[6.2 Architecture de l'API : - 24 -](#_Toc166441645)

[6.3 Architecture Front-End : - 25 -](#_Toc166441646)

[Conclusion - 25 -](#_Toc166441647)

[Question : Comment On a établi la liaison entre les deux projets : Backend et Frontend ? - 25 -](#_Toc166441648)

[Conclusion - 26 -](#_Toc166441649)

[7.Illustrations - 26 -](#_Toc166441650)

[7.1 Front - 26 -](#_Toc166441651)

[7.2 MySql Workbench - 30 -](#_Toc166441652)

[Conclusion - 32 -](#_Toc166441653)

[Conclusion Generale - 33 -](#_Toc166441654)

[Conclusion Generale - 34 -](#_Toc166441655)

# TABLE DE FIGURE

Figure 1 : Logo de L’EMSI …………………………………………………………………………………………………………1

Figure 2 : Tableau de Tache……………………………………………………………………………………………………..8

**Introduction Generale**

Dans un contexte où la gestion financière personnelle devient de plus en plus cruciale dans nos vies quotidiennes, les outils de gestion de dépenses se révèlent être des alliés indispensables pour aider les individus à maintenir un contrôle efficace sur leurs finances. Dans cette ère numérique, où les interactions avec les services bancaires et financiers se font de plus en plus en ligne, la nécessité d'une application de suivi des dépenses bien conçue et conviviale est plus grande que jamais. Le monde moderne est en constante évolution, et les défis posés par la gestion financière personnelle sont nombreux. Entre la multiplicité des transactions, la nécessité de catégoriser les dépenses, la gestion des budgets et la visualisation des tendances de dépenses, il est facile de se sentir submergé. C'est là qu'intervient notre projet de développement d'une application de gestion de dépenses. Notre objectif est de fournir aux utilisateurs une plateforme intuitive et complète qui simplifie la tâche de suivre et de gérer leurs dépenses financières. Cette application offrira une gamme de fonctionnalités, allant de la saisie des transactions à la création de budgets en passant par la visualisation des tendances de dépenses. En permettant aux utilisateurs de catégoriser leurs dépenses et de suivre l'évolution de leurs soldes financiers, notre application vise à leur fournir les

outils nécessaires pour prendre des décisions éclairées en matière de gestion financière. Pour répondre à ces besoins, notre équipe s'est engagée dans le développement d'une application Web et mobile qui soit à la fois conviviale et performante. En utilisant des technologies modernes telles que React et Spring Boot, nous avons travaillé à la création d'une expérience utilisateur fluide et réactive. De plus, notre approche agile nous a permis de répondre de manière flexible aux besoins changeants des utilisateurs tout au long du processus de développement. Dans cette introduction, nous allons explorer en détail les fonctionnalités, les technologies utilisées, les contraintes et les défis rencontrés lors du développement de notre application de gestion de dépenses. Nous allons également examiner les impacts potentiels de notre application sur la gestion financière personnelle et discuter des leçons apprises tout au long du processus de développement. En fin de compte, notre objectif est de fournir aux utilisateurs un outil pratique et efficace pour les aider à prendre le contrôle de leurs finances et à atteindre leurs objectifs financiers.

# Chapitre 1 : Contexte Generale

# 1.Introduction :

Ce chapitre constitue une introduction au projet de développement d'une application de gestion de dépenses, détaillant le contexte général dans lequel le projet s'inscrit, les objectifs visés, ainsi que les méthodologies adoptées pour sa réalisation.

# 2.Presentation du Sujet :

## 2.1 Problématique :

Face à la digitalisation croissante des services financiers et à l'importance croissante de la gestion financière personnelle, comment concevoir et développer une application de gestion de dépenses qui réponde efficacement aux besoins des utilisateurs tout en garantissant la sécurité et la confidentialité de leurs données.

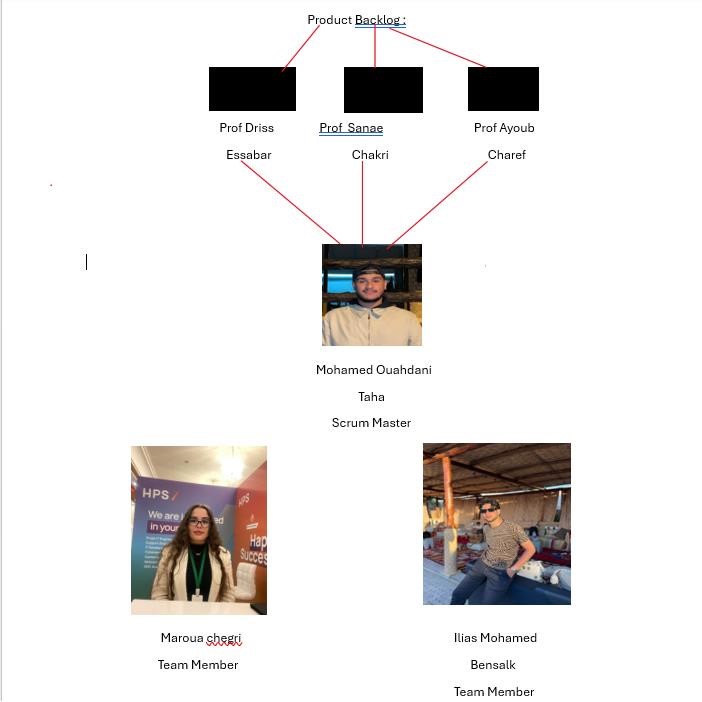
## 2.2 Objectif :

L'objectif de ce rapport est de présenter de manière détaillée le processus de conception et de développement d'une application de gestion de dépenses, en mettant en évidence les différentes étapes du projet, les technologies utilisées, les choix de conception, ainsi que les défis rencontrés et les solutions adoptées. Ce rapport vise également à fournir une analyse approfondie des fonctionnalités de l'application, de son interface utilisateur et de sa sécurité, ainsi qu'à évaluer sa conformité aux besoins des utilisateurs et aux exigences du cahier des charges. Enfin, ce rapport servira de documentation complète pour permettre une compréhension approfondie du projet et de ses implications.

# 3. Planification et Conduite du projet

## 3.1 Processus de Développement

### Meet the Team :



### Liens Github :

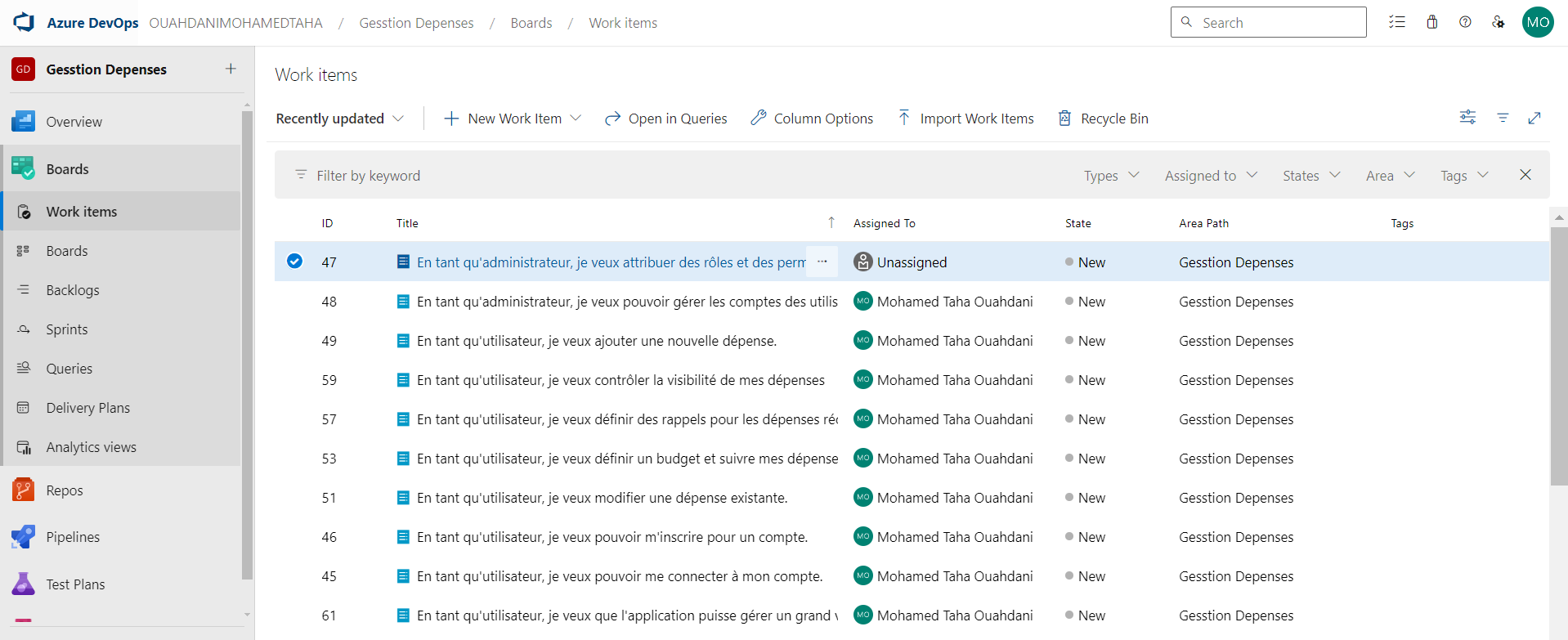
https://github.com/iir-projets/gestion\_depenses\_IIR4G2324G5G7.git

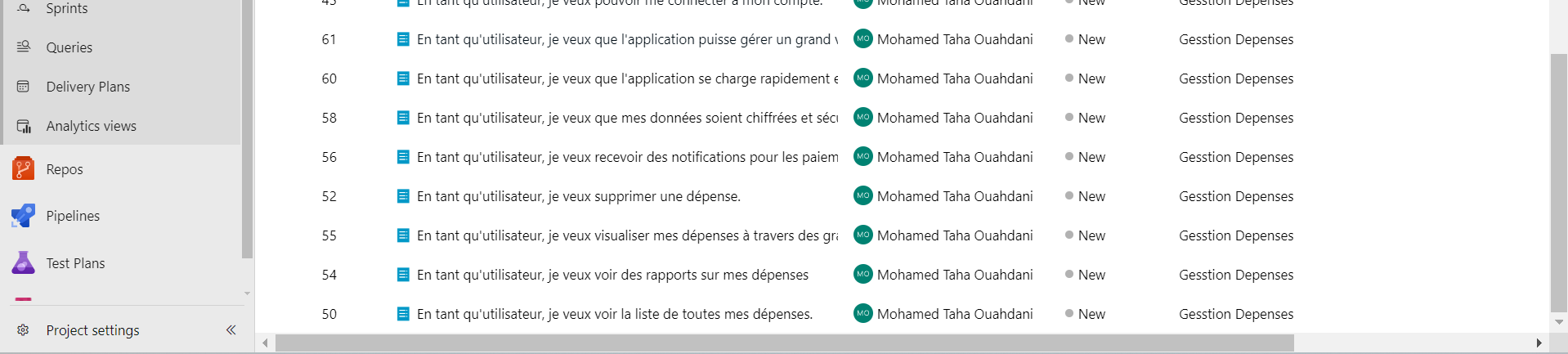
## 3.2 Tableau de Taches :

|  |  |  |
| --- | --- | --- |
|  | **Taches** | **Date** |
| **Tache 1** | . Formation sur Spring Boot et  MySQL | Le 01/03/2024 |
| **Tache 2** | . Configuration de l'environnement de développement | | |
| **Tache 3** | . Conception de la base de données  MySQL | | |
| **Tache 4** | Création des entités et des repositories JPA | | |
| **Tache 5** | Développement des API RESTful pour le backend | | |
| **Tache 6** | Tests unitaires et d'intégration du backend | | |
| **Tache 7** | Mise en place de l'authentification et de l'autorisation | | |
| **Tache 8** | Conception de l'interface utilisateur (Wireframes) | | |
| **Tache 9** | Implémentation de l'interface utilisateur et Dépenses en  React | | |
| **Tache 10** | Intégration des API RESTful avec l'interface React | | |
| **Tache 11** | Développement de l'application mobile avec  Android Studio | | |
| **Tache 12** | Intégration de l'interface utilisateur et Dépenses avec l'application Android | Le 08/05/2024 |

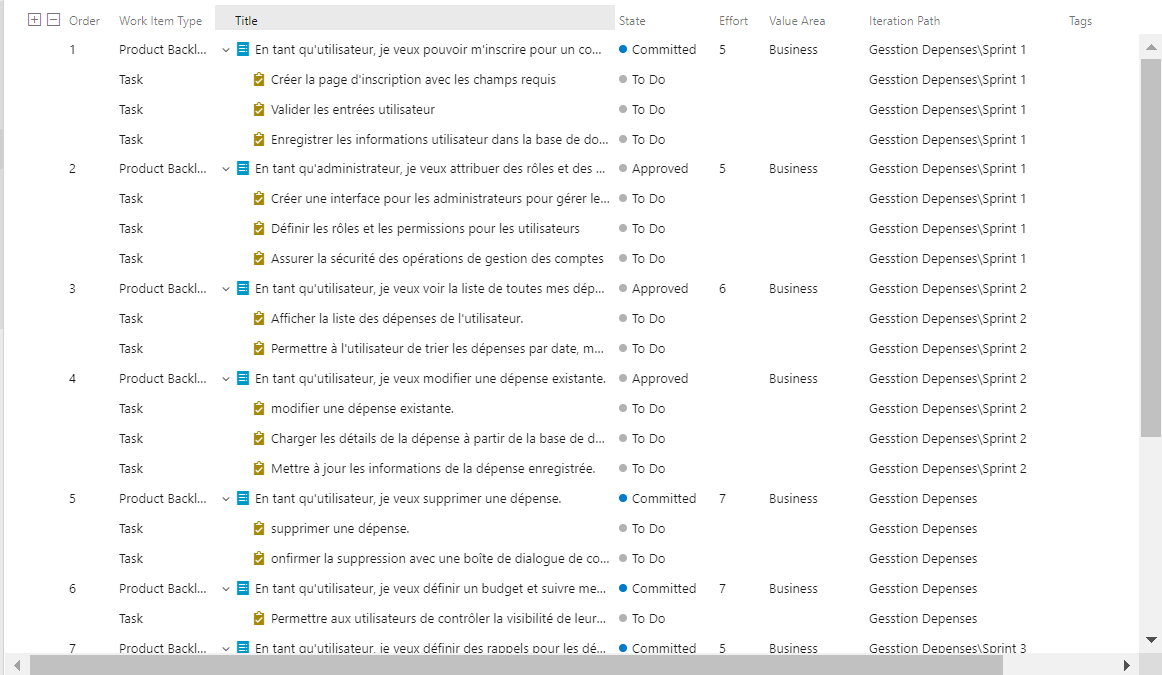
# Partie : Azure Devops :

## -Product Backlogs





## Taches :



Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

## Boards :

Une image contenant texte, logiciel, Icône d’ordinateur, nombre

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

## Sprints :

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

## 3.3 Planification

Pour garantir le bon déroulement du projet et répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible, nous organisions des réunions chaque semaine afin de suivre l'avancement du projet. Ces réunions servaient également à discuter de la fiabilité et du fonctionnement des outils utilisés, en identifiant et en corrigeant les bugs pour mettre à jour les besoins et ainsi atteindre l'objectif de livrer le projet dans les délais impartis.

## Conclusion

En conclusion, ce projet de gestion des dépenses a permis de créer une application web et mobile robuste et conviviale, offrant aux utilisateurs une expérience intuitive et efficace. Grâce à une collaboration étroite entre les équipes backend, frontend et Android, nous avons pu répondre aux besoins fonctionnels et non fonctionnels du cahier des charges. La méthodologie agile adoptée a joué un rôle crucial dans la gestion efficace du projet, en permettant une adaptation continue aux changements et en assurant la livraison du produit dans les délais impartis.

# 4. Analyse et Conception

## 4.1 Introduction

Ce chapitre est consacré à l'étude préliminaire des besoins, qui consiste essentiellement à analyser et diagnostiquer le système d'informations global, ainsi qu'à identifier et modéliser le contexte.

## 4.2 Spécifications fonctionnelles

**Gestion des Budgets :**

* Permettre aux utilisateurs de créer, modifier et supprimer des budgets pour différentes catégories de dépenses.
* Les budgets doivent spécifier un montant total et une période (mensuelle, hebdomadaire, etc.).

**Visualisation des Tendances :**

* Afficher les dépenses sous forme de graphiques (histogrammes, camemberts, etc.).
* Les graphiques doivent montrer les dépenses par catégorie, par période (jour, semaine, mois) et par type (revenus, dépenses).

**Gestion des Catégories :**

* Permettre aux utilisateurs de créer, modifier et supprimer des catégories pour classer leurs dépenses.

**Suivi des Soldes :**

* Afficher le solde financier actuel.
* Afficher l'historique des soldes.

**Gestion des Comptes Utilisateurs :**

* Permettre aux utilisateurs de créer, modifier et supprimer leur compte.
* L'authentification est requise pour accéder à l'application.

**Notifications :**

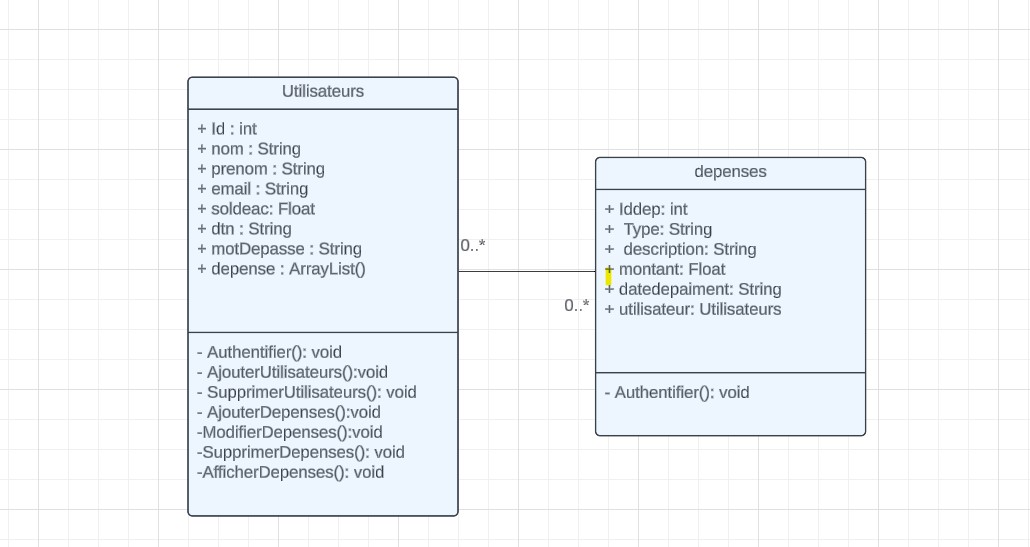
* Envoyer des notifications pour rappeler aux utilisateurs de saisir leurs dépenses, lors du dépassement de budgets, etc.

## 4.3 Spécifications Non Fonctionnelles :

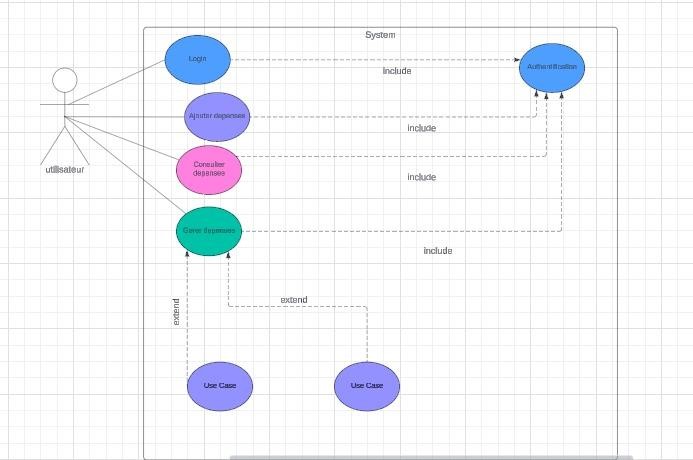
1. **Performance :**
   * L'application doit être réactive même avec de grandes quantités de données.
   * Temps de réponse rapide pour toutes les interactions de l'utilisateur.
2. **Sécurité et Confidentialité :**
   * Stockage sécurisé des données utilisateur avec cryptage.
   * Protection des données personnelles des utilisateurs.
   * Authentification sécurisée des utilisateurs pour accéder à leur compte.
3. **Fiabilité :**
   * Fonctionnement sans erreur et avec une disponibilité élevée.
   * Gestion appropriée des erreurs et des exceptions.
4. **Compatibilité :**
   * L'application doit être compatible avec les navigateurs web modernes (Chrome, Firefox, Safari, Edge) ainsi qu'avec les appareils mobiles (Android, iOS).
5. **Accessibilité :**
   * Respect des normes d'accessibilité pour garantir que l'application est utilisable par tous les utilisateurs, y compris ceux ayant des besoins spécifiques.

## 4.4 La Méthode UML :

#### 4.4.1 Diagramme de Classe

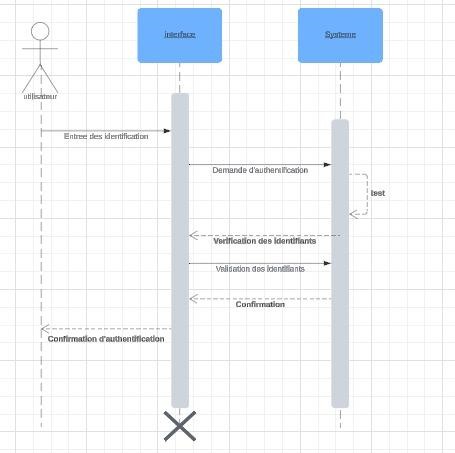


#### 4.4.2 Diagramme de Cas d’utilisation

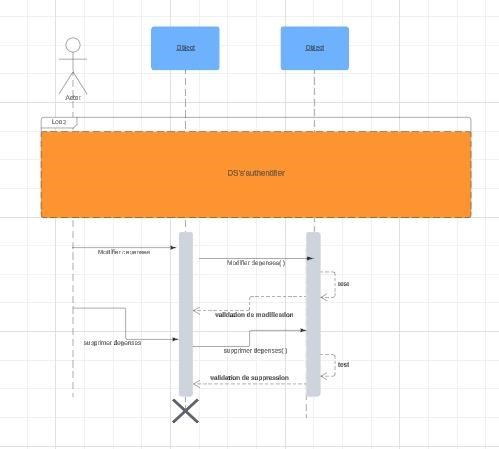


#### 4.4.3 Diagramme de Séquence

-Authentification « Sign up »



-Gérer Dépense :



## Conclusion

La conception d'une application de gestion des dépenses, intégrant les acteurs principaux tels que l'Utilisateur et les Dépenses, offre une vue d'ensemble des interactions et des fonctionnalités essentielles du système.En identifiant ces acteurs et en définissant leurs rôles spécifiques ainsi que les cas d'utilisation associés, nous avons établi les bases nécessaires pour concevoir et développer efficacement l'application. La présence de l'acteur "Utilisateur" souligne son rôle central dans l'utilisation de l'application, en ajoutant, modifiant et supprimant les dépenses ainsi qu'en consultant les rapports.

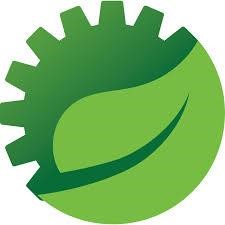
# 5.Realisation Et Outils Utilises

## Introduction

Dans cette section, nous abordons la phase de réalisation de notre projet de gestion des dépenses. Nous décrirons les outils et technologies que nous avons utilisés pour développer l'application, ainsi que les différentes étapes du processus de développement. Cette partie met en lumière les choix technologiques et les méthodes de développement qui ont permis de concrétiser notre projet.

## 5.1 Outils de Développement

Spring Tool Suite : STS



Description : STS est un environnement de développement intégré (IDE) basé sur Eclipse, conçu spécifiquement pour le développement d'applications Spring.

JAVA :

Description : Java est un langage de programmation polyvalent et populaire, largement utilisé pour le développement d'applications backend.

Postman :

Description : Postman est une plateforme de développement d'API qui permet de créer, tester et déboguer des API.

Visuel Studio Code :



Description : Visual Studio Code est un éditeur de code source léger et très performant développé par Microsoft.

React :

Description : React est une bibliothèque JavaScript open-source développée par Facebook, principalement utilisée pour construire des interfaces utilisateur interactives et réactives.



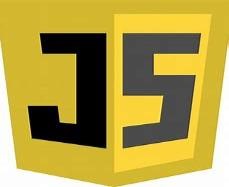
MySQL Workbench

Description : MySQL Workbench est un outil visuel de conception de bases de données MySQL et de gestion de serveur de base de données.



Android Studio :

Description : Android Studio est l'environnement de développement intégré (IDE) officiel pour le développement d'applications Android. Il est développé par Google et basé sur IntelliJ IDEA.

 JAVASCRIPT :

Description : JavaScript est un langage de programmation de haut niveau, interprété et orienté objet, principalement utilisé pour créer des applications web interactives et dynamiques.

UML :

Description **:** est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d’un système.

# 6.Architecture Globale

## 6.1 Introduction

L'architecture globale d'un projet constitue le fondement sur lequel reposent tous les aspects du développement, de la conception à la mise en œuvre. Elle définit la structure générale de l'application, les interactions entre ses différentes composantes et les technologies utilisées pour garantir son bon fonctionnement. Dans le cadre de notre projet de gestion de dépenses, l'architecture globale est conçue pour assurer une gestion efficace des données, une communication fluide entre le front-end et le back-end, ainsi qu'une expérience utilisateur optimale. Cette architecture repose sur trois piliers principaux : l'architecture de l'API, l'architecture front-end et l'architecture back-end.

## 6.2 Architecture de l'API :

L'architecture de l'API (Application Programming Interface) définit la manière dont le front-end communique avec le back-end pour échanger des données. Elle est construite autour de points d'entrée bien définis qui permettent au front-end d'accéder aux fonctionnalités et aux données du back-end de manière sécurisée et efficace.

Dans notre projet, l'architecture de l'API repose sur des principes RESTful (Représentationnel State Transfer) qui utilisent des requêtes HTTP standard (GET, POST, PUT, DELETE) pour effectuer des opérations CRUD (Create, Read, Update, Delete) sur les ressources. Cela garantit une interaction uniforme et prévisible entre le front-end et le back-end, facilitant ainsi le développement et la maintenance de l'application.

## 6.3 Architecture Front-End :

L'architecture front-end concerne la structure et l'organisation des composants visuels de l'application, ainsi que leur interaction avec l'utilisateur. Elle définit la manière dont les données sont présentées et manipulées dans l'interface utilisateur, tout en assurant une expérience utilisateur intuitive et cohérente. Dans notre projet, l'architecture front-end est basée sur le framework React.js, qui permet de créer des applications web dynamiques et réactives. Nous utilisons une architecture de composants réutilisables, où chaque composant est responsable d'une partie spécifique de l'interface utilisateur. Cela permet une gestion efficace de l'état de l'application et une mise à jour dynamique de l'interface en fonction des interactions de l'utilisateur.

## Conclusion

En résumé, l'architecture globale de notre projet de gestion de dépenses repose sur une communication fluide entre le front-end et le back-end via une API RESTful, combinée à une architecture front-end basée sur React.js. Cette approche garantit une gestion efficace des données, une expérience utilisateur optimale et une évolutivité future de l'application.

# Question : Comment On a établi la liaison entre les deux projets : Backend et Frontend ?

→Pour lier la base de données à notre application, nous utilisons MySQL Workbench comme système de gestion de base de données relationnelle. MySQL Workbench nous permet de concevoir, modéliser et gérer les bases de données MySQL de manière visuelle. Nous utilisons cet outil pour créer et gérer les schémas de base de données, définir les tables, les colonnes, les clés primaires et étrangères, ainsi que les contraintes d'intégrité référentielle.

* Dans le back-end de notre application développée avec Spring Boot, nous utilisons Spring Data JPA pour interagir avec la base de données MySQL. Nous créons des entités Java qui représentent les tables de la base de données, en les annotant avec des annotations spécifiques de JPA pour mapper les colonnes de la table. Les repositories JPA sont utilisés pour effectuer des opérations CRUD sur les entités. Lorsqu'une requête est reçue du front-end, le service correspondant dans le back-end utilise les repositories pour effectuer des opérations sur la base de données.

* Dans le front-end, développé avec React, nous utilisons Axios pour envoyer des requêtes HTTP au back-end afin d'accéder aux données de la base de données. Lorsqu'une page est chargée ou qu'une action est effectuée par l'utilisateur, le frontend envoie une requête appropriée au back-end pour récupérer ou modifier les données. Une fois les données récupérées, le front-end les affiche à l'utilisateur en utilisant des composants React pour la mise en forme et l'affichage des données.

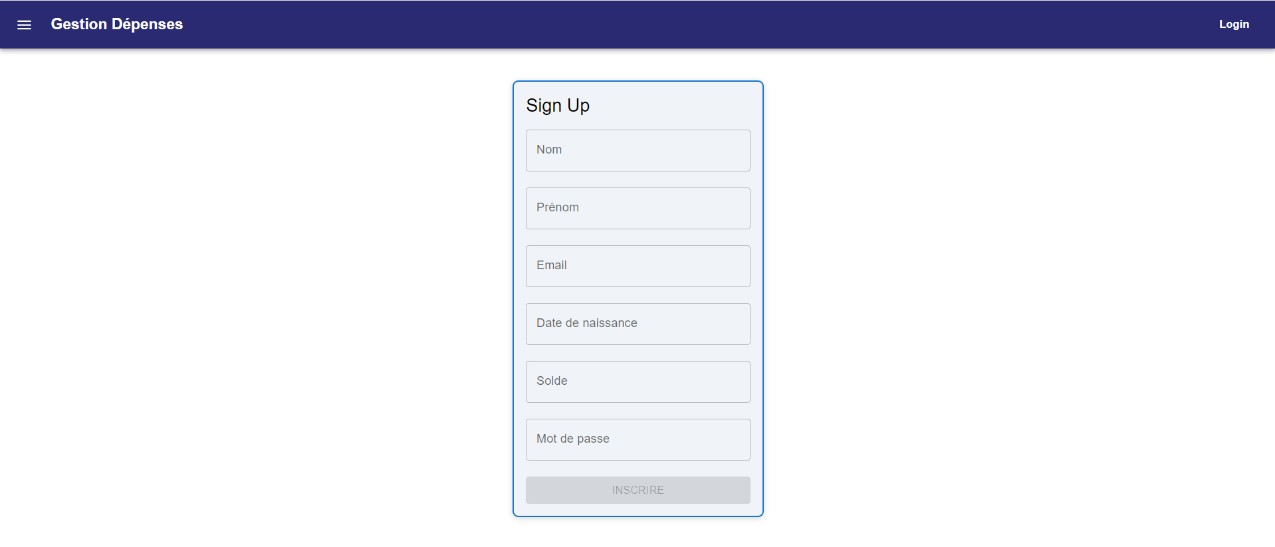
# Conclusion

La liaison entre la base de données et notre application est réalisée via MySQL Workbench pour la gestion de la base de données, et Spring Data JPA dans le back-end pour interagir avec la base de données. Cette approche assure une gestion efficace des données et garantit la cohérence entre la base de données et l'application.

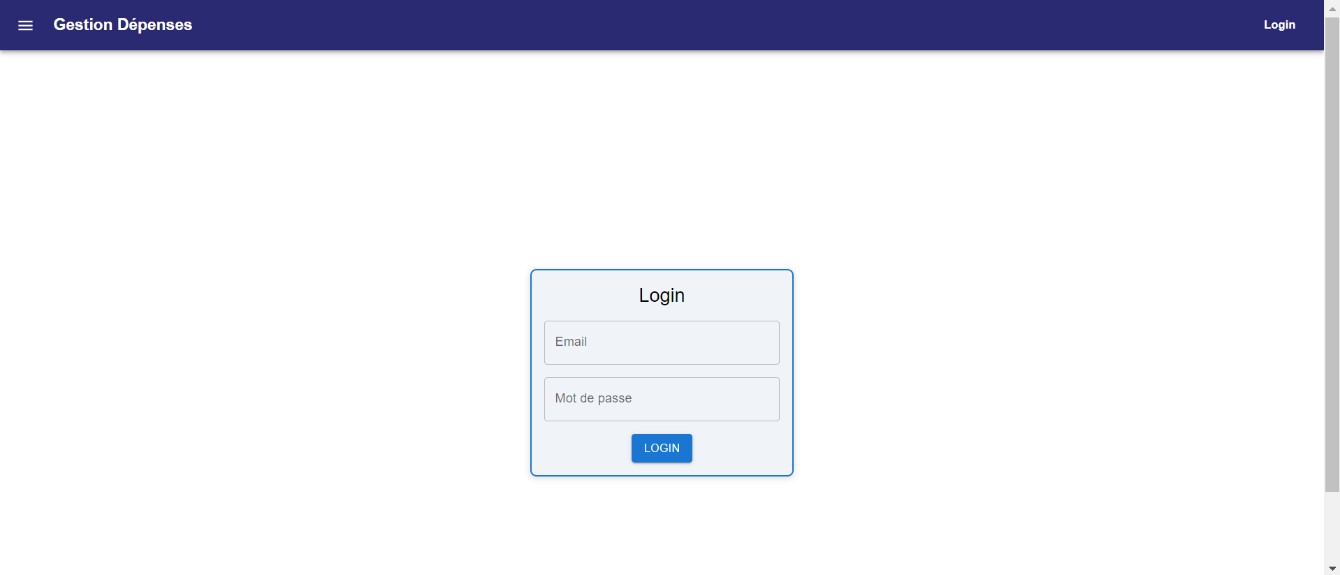
# 7.Illustrations

## 7.1 Front

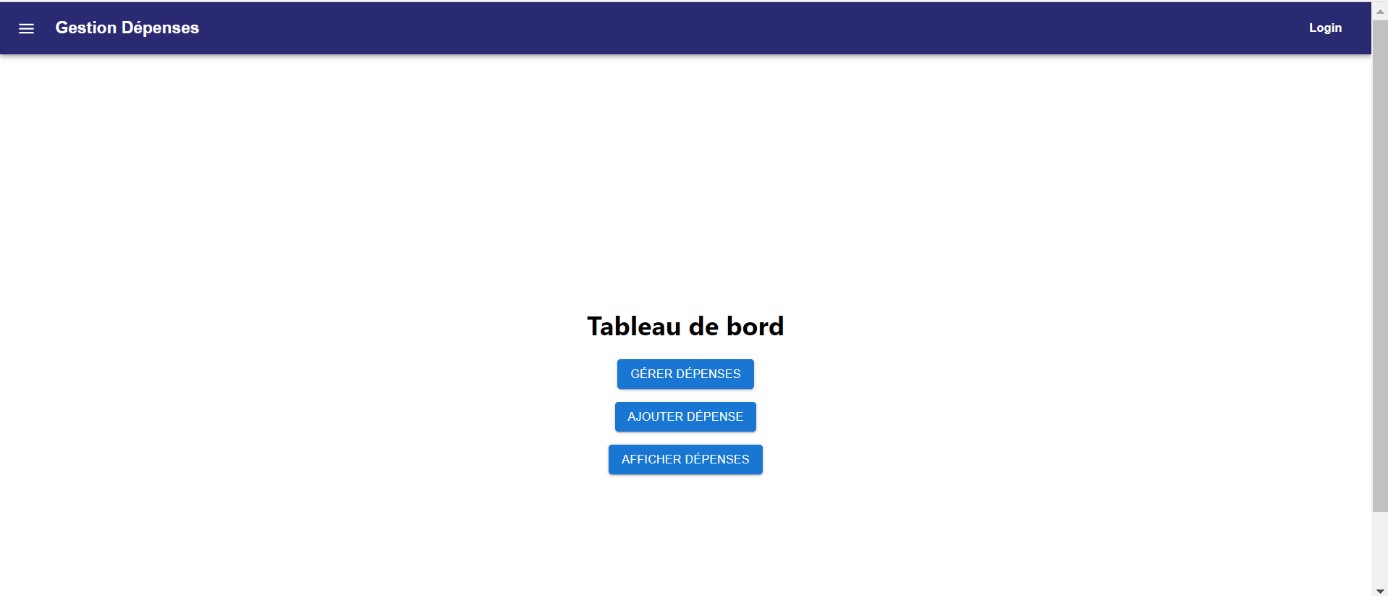
#### 7.1.1 Sign up



7.1.2 Login



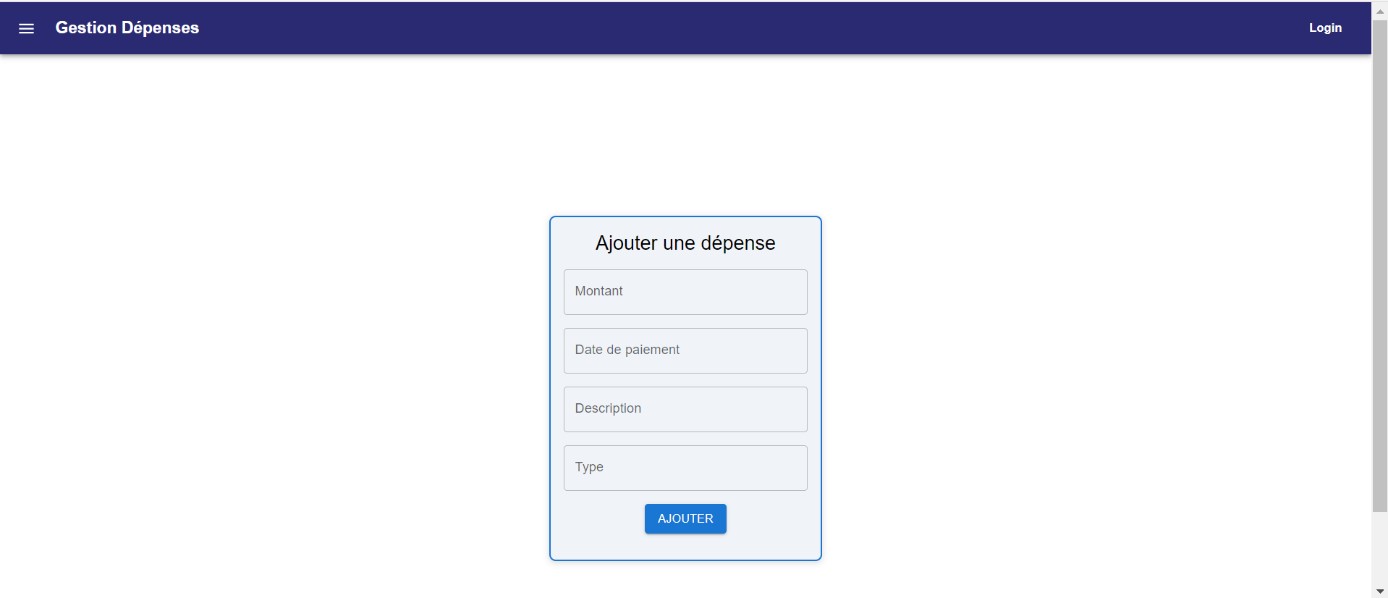
##### 7.1.3 Tableau de Bord



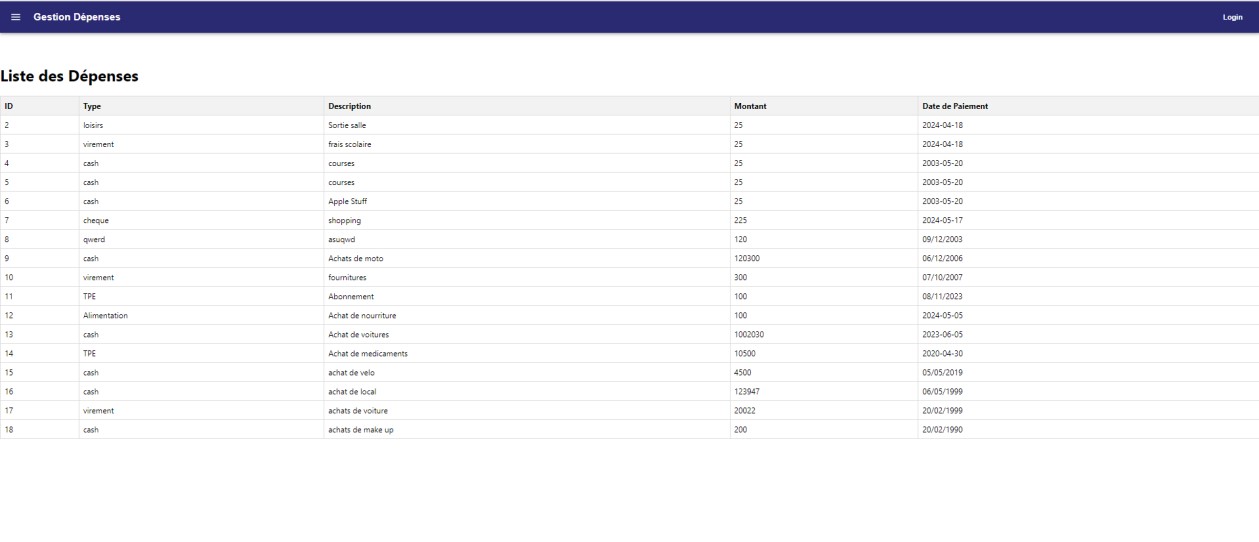
#### 7.1.4 Gérer Dépenses



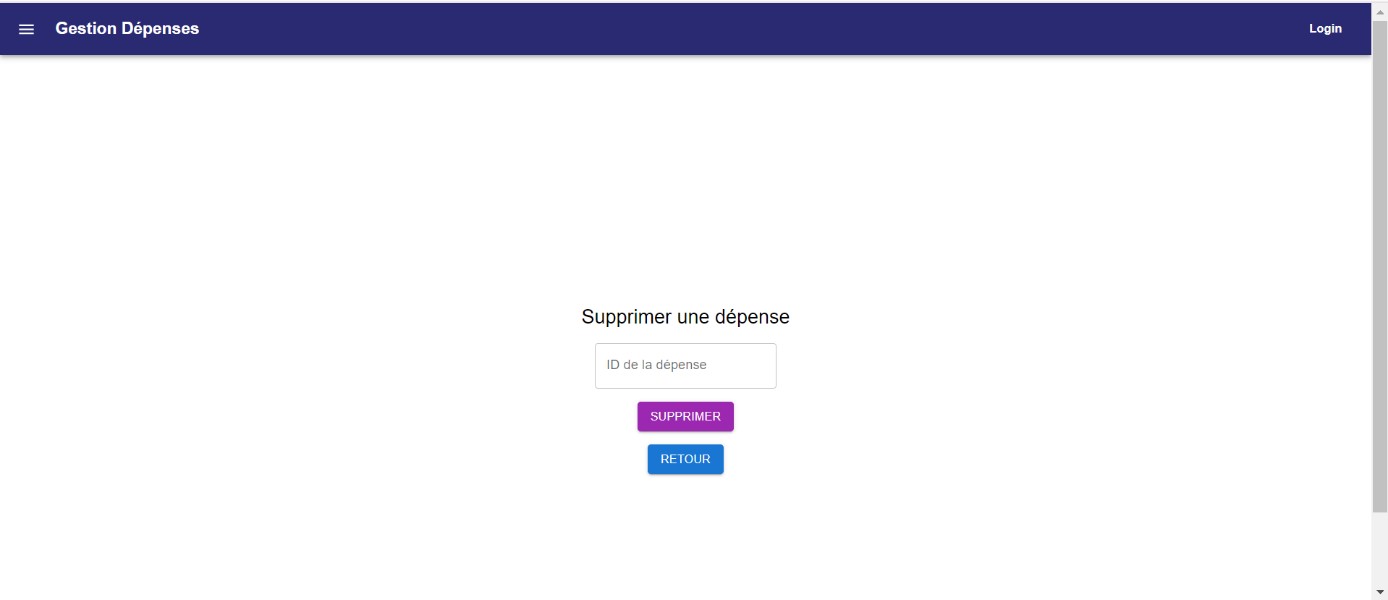
###### 7.1.5 Ajouter Dépenses



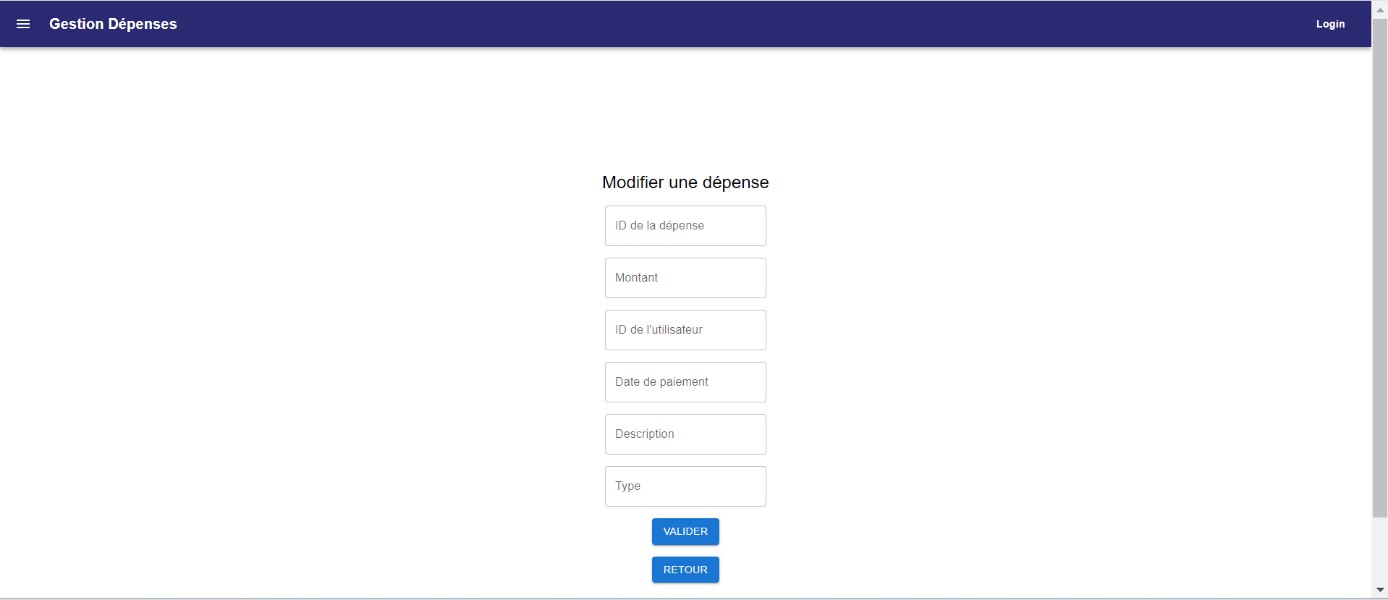
#### 7.1.6 Afficher Dépenses



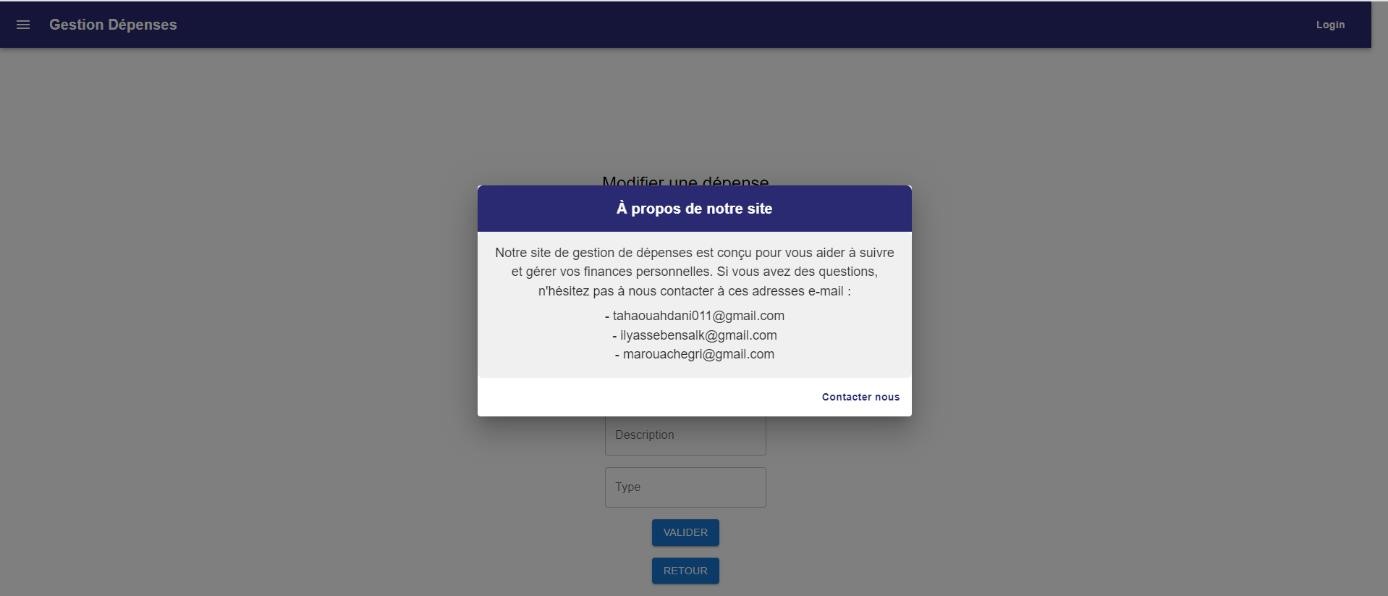
#### 7.1.7 SupprimerDepense



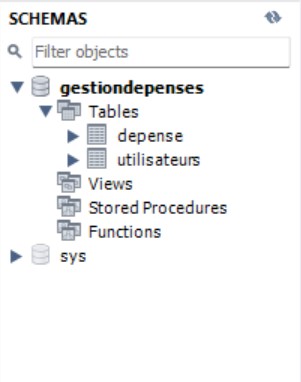
#### 7.1.8 ModifierDepense



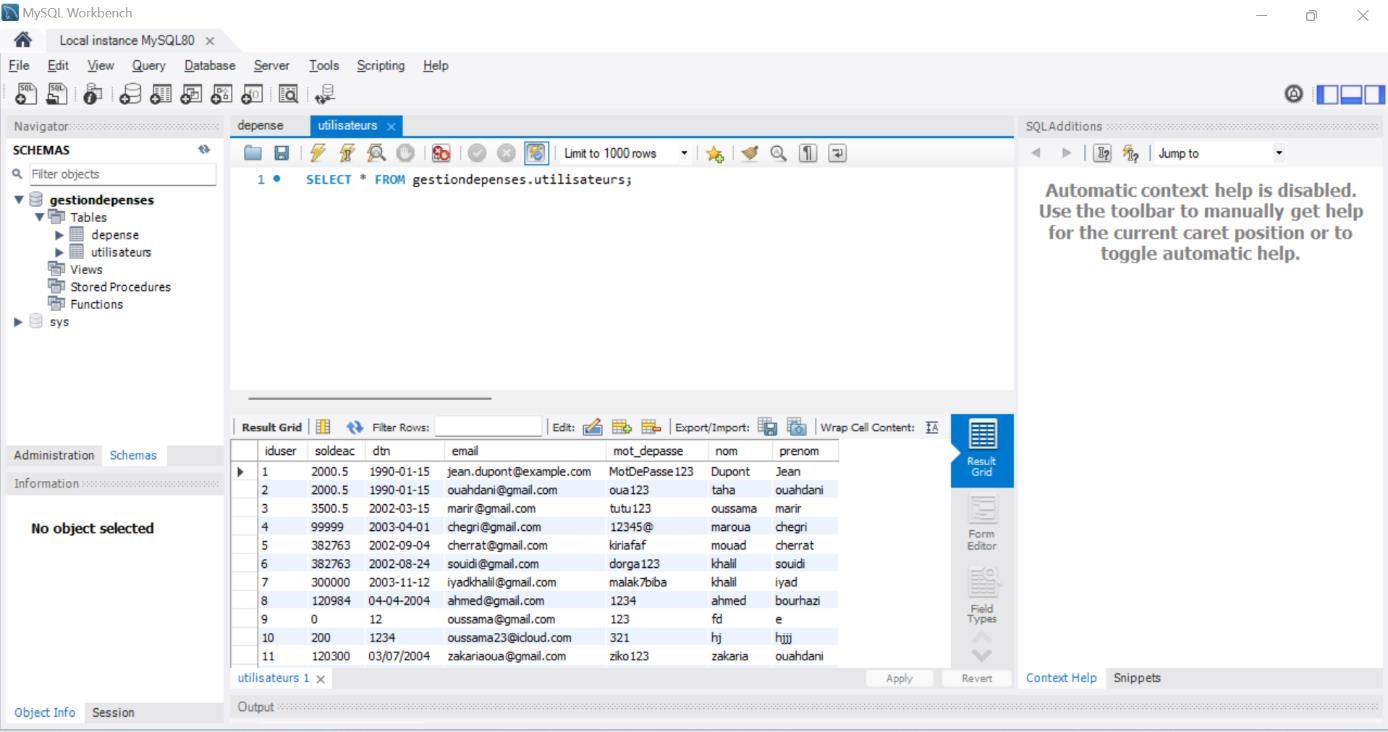
#### 7.1.9 A propos de nous



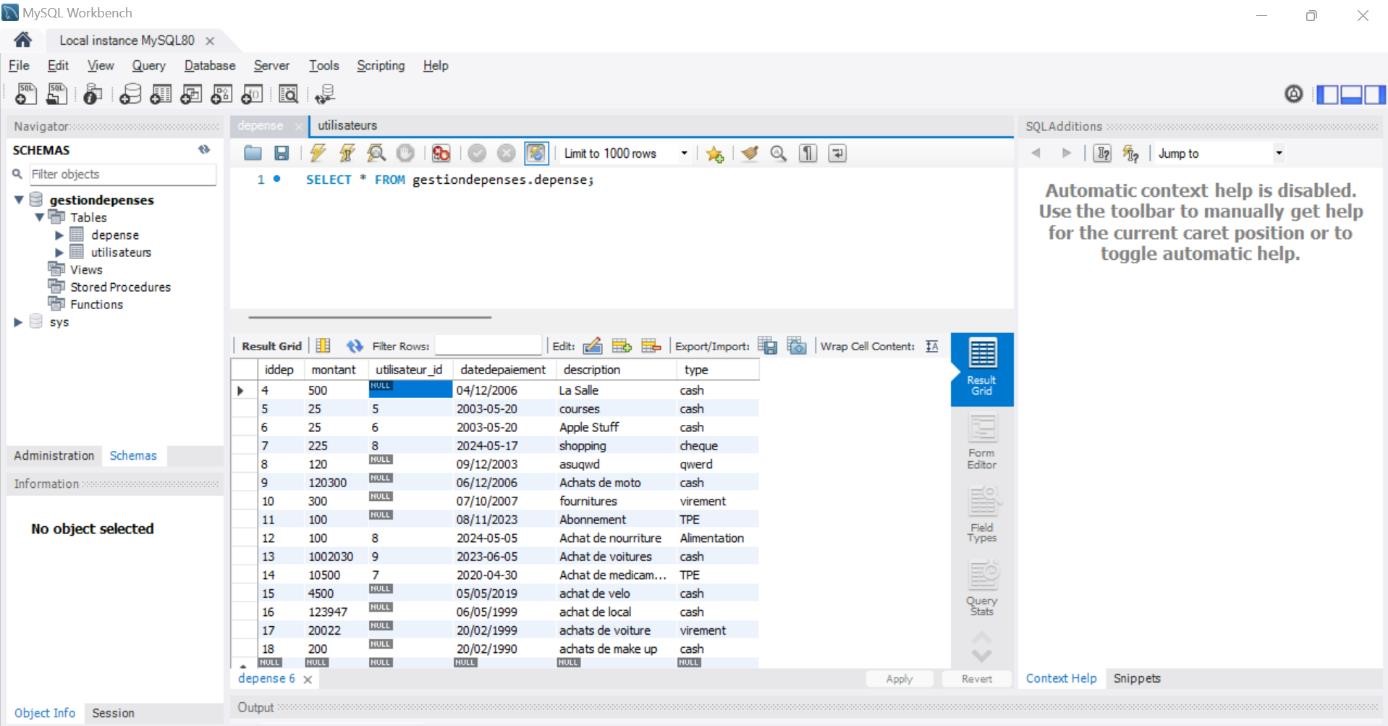
### 7.2 MySql Workbench



#### 7.2.1 Utilisateur



#### 7.2.2 Dépense



## Conclusion

En conclusion, avant d'entamer la phase de réalisation de notre projet, il est crucial de définir clairement les objectifs et les fonctionnalités du système. Cette étape permet de comprendre les limites d'utilisation, ainsi que les services et les solutions offerts aux utilisateurs. Grâce à une analyse approfondie, nous avons pu définir les besoins et les attentes des utilisateurs, ce qui a guidé la conception de notre système.

# Conclusion Generale

## Conclusion Generale

En conclusion, ce projet représente une étape significative dans le développement d'un système de gestion des dépenses personnelles. Avant d'entrer dans les détails de la réalisation, une analyse approfondie des besoins des utilisateurs a été effectuée afin de définir clairement les objectifs et les fonctionnalités du système. Cette étape initiale s'est révélée cruciale pour orienter le processus de conception et assurer la pertinence du système par rapport aux attentes des utilisateurs.

La phase de conception a été caractérisée par la création de diverses illustrations, telles que des diagrammes de cas d'utilisation, des diagrammes de séquence, et des maquettes d'interface utilisateur. Ces illustrations ont permis de visualiser et de simuler le fonctionnement du système, offrant ainsi une vue d'ensemble claire de sa structure et de ses interactions.

L'implémentation du système s'est déroulée en plusieurs étapes, mettant en œuvre les fonctionnalités essentielles identifiées lors de la phase de conception. L'utilisation de technologies modernes et l'adoption de bonnes pratiques de développement ont contribué à la création d'un système fiable, évolutif et sécurisé.

Enfin, une série de tests approfondis a été réalisée pour garantir le bon fonctionnement du système et sa conformité aux exigences spécifiées. Des tests d'acceptation utilisateur ont également été menés pour valider l'expérience utilisateur et s'assurer que le système répond aux besoins des utilisateurs finaux.

Au terme de ce projet, nous avons pu développer un système de gestion des dépenses personnelles efficace et intuitif, offrant aux utilisateurs une solution pratique pour contrôler et suivre leurs dépenses au quotidien. Ce projet a été une opportunité précieuse d'appliquer nos connaissances théoriques à un cas pratique, et de développer nos compétences en conception et en développement logiciel.