|  |  |  |
| --- | --- | --- |
|  | **REPUBLIQUE TUNISIENNE**  \*\*\*\*\*  MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  \*\*\*\*\*  DIRECTION GENERALE DES ETUDES TECHNOLOGIQUES  \*\*\*\*\*  INSTITUT SUPERIEUR DES ETUDES TECHNOLOGIQUES DE CHARGUIA  \*\*\*\*\*  **Département Technologies de l’Informatique** | Institut Supérieur des Etudes Technologiques de Charguia ... |

**RAPPORT DE**

|  |
| --- |
| Stage de perfectionnement |

Elaboré par

**Youssef Abid**

**Encadré par :**

**Mr** Hatem AMMAR

**Société d’accueil :** Business Management Group

**Année Universitaire : 2023/2024**

Remerciements

À l'issue de notre stage, nous souhaitons exprimer notre gratitude envers toutes les personnes qui ont contribué à sa réussite.

Tout d'abord, nous remercions sincèrement Monsieur Hatem Ammar, Directeur Général de Business Management Group, pour son accueil chaleureux et son soutien tout au long de notre période au sein de l'entreprise.

Un grand merci également à Madame Nour Derouich, Développeuse Full Stack, pour son précieux soutien et son assistance tout au long du stage. Sa collaboration et ses conseils ont été d'une aide précieuse pour surmonter les défis rencontrés.

Enfin, nous exprimons notre reconnaissance à l'ensemble du personnel de Business Management Group pour leur accueil et leur collaboration tout au long de notre séjour. Votre environnement de travail favorable a grandement contribué à notre apprentissage.

Nous tenons à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce stage.

**Sommaire**

Table des matières

[Introduction générale 1](#_Toc160001250)

[Chapitre 1 : Présentation du cadre du projet 2](#_Toc160001251)

[I. Présentation de la société 2](#_Toc160001252)

[II. Etude de l’existant 3](#_Toc160001253)

[1. Description de l’existant 3](#_Toc160001254)

[2. Critique de l’existant 3](#_Toc160001255)

[3. Solution proposée 4](#_Toc160001256)

[III. Cahier des charges 4](#_Toc160001257)

[1. Besoins fonctionnels 4](#_Toc160001258)

[2. Besoins non fonctionnels 5](#_Toc160001259)

[Chapitre 2 : modélisation comportementale de l’application 7](#_Toc160001260)

[I. Identification des acteurs 7](#_Toc160001261)

[II. Présentation du diagramme de cas d’utilisation [voir Annexes] 7](#_Toc160001262)

[III. Analyse des cas d’utilisation 12](#_Toc160001263)

[Chapitre 3 : Modélisation structurelle de l’application 17](#_Toc160001264)

[I. Diagramme de classes 17](#_Toc160001265)

[1. Descriptif du diagramme 17](#_Toc160001266)

[2. Représentation du diagramme 19](#_Toc160001267)

[II. Modélisation de la base de donnée 20](#_Toc160001268)

[Chapitre 4 : Réalisation 21](#_Toc160001269)

[I. Environnement de développement 21](#_Toc160001270)

[1. Environnement matérielle 21](#_Toc160001271)

[2. Environnement logiciel 21](#_Toc160001272)

[II. Principales interfaces graphiques 25](#_Toc160001273)

[Conclusion générale 29](#_Toc160001274)

[Nétographie 30](#_Toc160001275)

**Liste des figures**

[Figure 1 Calendrier use case 8](#_Toc159969610)

[Figure 2 Enregistrer Utilisateur 13](#_Toc159969611)

[Figure 3 S'authentifier 13](#_Toc159969612)

[Figure 4 afficher calendrier 14](#_Toc159969613)

[Figure 5 créer event 14](#_Toc159969614)

[Figure 6 edition event 15](#_Toc159969615)

[Figure 7 supprition event 15](#_Toc159969616)

[Figure 8 reservation creneau 16](#_Toc159969617)

[Figure 9 diagramme de classe 19](#_Toc159969618)

[Figure 10 modele relationnel 20](#_Toc159969619)

[Figure 11 node js 22](#_Toc159969620)

[Figure 12 express js 22](#_Toc159969621)

[Figure 13 postgreSQL 23](#_Toc159969622)

[Figure 14 postman 23](#_Toc159969623)

[Figure 15 vs code 24](#_Toc159969624)

[Figure 16 starUML 24](#_Toc159969625)

[Figure 17 interface windows 11 25](#_Toc159969626)

[Figure 18 les requetes 26](#_Toc159969627)

[Figure 19 les routes 26](#_Toc159969628)

[Figure 20 les fonctions 27](#_Toc159969629)

[Figure 21 data base connection 27](#_Toc159969630)

[Figure 22 server open 28](#_Toc159969631)

[Figure 23 test un API 28](#_Toc159969632)

# Introduction générale

Dans le monde moderne où la gestion du temps est cruciale, le besoin d'outils efficaces pour organiser les activités et les rendez-vous est devenu primordial. C'est dans ce contexte que s'inscrit notre projet de développement d'une application de calendrier innovante. Notre objectif est de créer une plateforme robuste et conviviale permettant aux utilisateurs de gérer leurs événements et leurs rendez-vous de manière efficace, tout en offrant une expérience utilisateur optimale.

Ce rapport présente en détail le travail réalisé lors de notre stage, axé sur la conception et le développement de cette application de calendrier. Dans un premier temps, nous introduirons le cadre dans lequel s'inscrit ce projet, en mettant en lumière les objectifs visés et les contraintes à considérer. Ensuite, nous aborderons la modélisation comportementale et structurelle de l'application, décrivant ainsi son fonctionnement et son architecture.

Le chapitre dédié à la réalisation mettra en avant les étapes concrètes de développement, les technologies utilisées et les choix techniques effectués pour répondre aux exigences du projet. Enfin, nous conclurons ce rapport en évaluant la conformité de notre solution aux critères établis, notamment en termes de fonctionnalités, de qualité du code et de performance du système.

Dans cette optique, nous nous attacherons à présenter de manière claire et concise chaque aspect de notre travail, en mettant en évidence les contributions et les défis rencontrés tout au long du processus de développement

# Chapitre 1 : Présentation du cadre du projet

Le chapitre introductif présente le contexte du projet, débutant par une présentation de la société d'accueil, éclairant ses activités et son positionnement. Il se poursuit par une analyse des pratiques actuelles de gestion des calendriers, offrant ainsi un aperçu des besoins spécifiques à adresser par notre application.

## Présentation de la société

Business Management Group est une entreprise dynamique, animée par un esprit visionnaire et innovateur, dédiée à répondre aux besoins de ses clients dans un monde en perpétuelle évolution. Spécialisée dans les domaines de la Formation Professionnelle et du Conseil, BMG met son expertise au service de ses clients pour renforcer leurs compétences et accroître leur performance dans un environnement économique et social exigeant.

Ses objectifs sont clairs :

Assurer la durabilité et la croissance de votre entreprise.

Vous permettre d'atteindre vos objectifs professionnels et personnels.

Renforcer vos compétences et accroître vos performances professionnelles.

Ses domaines d'activités couvrent divers aspects :

BMG Consultancy : optimisation des processus grâce à des services de conseil personnalisés.

BMG Talents : offrir des formations intra et interentreprises adaptées à vos besoins.

BMG Purple : gestion et développement d'une communauté autour de votre marque pour renforcer votre présence sur le marché.

BMG Technologies : accompagner la digitalisation de votre entreprise pour rester compétitif dans un monde numérique en constante évolution.

## Etude de l’existant

### Description de l’existant

Actuellement au sein de la société BMG, la gestion des événements et des rendez-vous se fait de manière traditionnelle, principalement par des moyens manuels tels que des calendriers physiques ou des applications tierces. Il n'existe pas de solution interne dédiée à cette gestion spécifique. Les employés utilisent des méthodes individuelles pour gérer leurs rendez-vous et leurs emplois du temps, ce qui peut entraîner des inefficacités et des conflits d'horaires.

En ce qui concerne les solutions existantes sur le marché, il existe plusieurs applications de calendrier, telles que Google Agenda, Outlook Calendar et Calendly. Cependant, aucune de ces solutions ne répond parfaitement aux besoins spécifiques de la société BMG en termes de fonctionnalités et de sécurité.

### Critique de l’existant

La méthode actuelle de gestion des événements et des rendez-vous présente plusieurs lacunes importantes. Tout d'abord, l'utilisation de calendriers physiques ou d'applications tierces non adaptées à l'activité de l'entreprise peut entraîner une perte d'efficacité et une augmentation des risques d'erreurs. De plus, l'absence de centralisation des données et de fonctionnalités spécifiques telles que la gestion des créneaux horaires disponibles et les rappels automatiques limite la productivité et la coordination au sein de l'entreprise.

La problématique principale de ce projet réside dans la nécessité de mettre en place une solution de gestion des événements et des rendez-vous qui soit à la fois efficace, sécurisée et adaptée aux besoins spécifiques de la société BMG.

### Solution proposée

Pour répondre à cette problématique, nous proposons le développement d'une application de calendrier personnalisée, utilisant PostgreSQL comme base de données et Node.js avec Express comme framework pour le backend. Cette application offrira une interface utilisateur intuitive similaire à Calendly tout en intégrant les fonctionnalités spécifiques nécessaires à la gestion des événements et des rendez-vous au sein de la société BMG.

Les fonctionnalités clés de cette solution incluront la gestion des utilisateurs avec authentification sécurisée, un calendrier avec affichage mensuel, hebdomadaire et quotidien des événements, la possibilité de créer, éditer et supprimer des événements avec fonctionnalité de glisser-déposer, la réservation de créneaux horaires disponibles, ainsi que des notifications par e-mail pour les nouveaux événements et les réservations, avec des rappels automatiques pour les événements à venir.

## Cahier des charges

### Besoins fonctionnels

Gestion des utilisateurs :

Enregistrement des utilisateurs avec leurs informations de base.

Authentification sécurisée pour accéder au calendrier.

Calendrier :

Affichage mensuel, hebdomadaire et quotidien des événements et rendez-vous.

Possibilité de créer, éditer et supprimer des événements.

Fonctionnalité de glisser-déposer pour faciliter la gestion des événements.

Réservation de créneaux :

Permettre aux utilisateurs de définir des créneaux horaires disponibles.

Les autres utilisateurs peuvent réserver ces créneaux disponibles.

Notifications :

Notification par e-mail pour les nouveaux événements et les réservations.

Rappels automatiques pour les événements à venir.

### Besoins non fonctionnels

Performance :

Réduire le temps de réponse de l'application pour garantir une expérience utilisateur fluide.

Optimiser les requêtes à la base de données pour minimiser les temps de chargement.

Sécurité :

Mettre en place des mesures de sécurité robustes pour protéger les données des utilisateurs.

Implémenter des mécanismes d'authentification et de gestion des accès sécurisés.

Maintenabilité :

Assurer une architecture modulaire et bien structurée pour faciliter la maintenance et les évolutions futures de l'application.

Documenter le code de manière exhaustive pour permettre aux développeurs de comprendre rapidement son fonctionnement.

Contraintes techniques :

Respect des bonnes pratiques de développement, de sécurité et de performance.

Ce cahier des charges reflète les besoins essentiels à la conception et au développement de l'application de calendrier pour répondre aux exigences spécifiques de la société BMG.

En conclusion, ce chapitre a mis en lumière le contexte du projet en présentant la société BMG, ses activités ainsi que les lacunes de ses pratiques actuelles de gestion des calendriers. La proposition d'une solution personnalisée et l'énoncé des besoins fonctionnels et non fonctionnels témoignent de l'engagement à répondre efficacement aux besoins spécifiques de BMG tout en garantissant des standards élevés de performance, sécurité et convivialité.

# Chapitre 2 : modélisation comportementale de l’application

Le chapitre de modélisation comportementale de l'application vise à définir les actions à entreprendre pour répondre aux besoins identifiés. À travers l'identification des acteurs, la présentation du diagramme de cas d'utilisation et l'analyse des cas les plus pertinents, ce chapitre offre une compréhension claire des interactions entre les utilisateurs et le système.

## Identification des acteurs

Dans le cadre de notre application de calendrier, les acteurs principaux sont les suivants :

* Utilisateur : Tout individu utilisant l'application de calendrier pour gérer ses événements et rendez-vous.
* Systeme : Responsable de la gestion des utilisateurs, des autorisations et de la maintenance de l'application.

## Présentation du diagramme de cas d’utilisation [voir Annexes]

Remarque : Si le diagramme de cas d'utilisation principal est illisible ou difficile à interpréter, veuillez vous référer à un diagramme plus clair disponible dans les annexes de ce document.

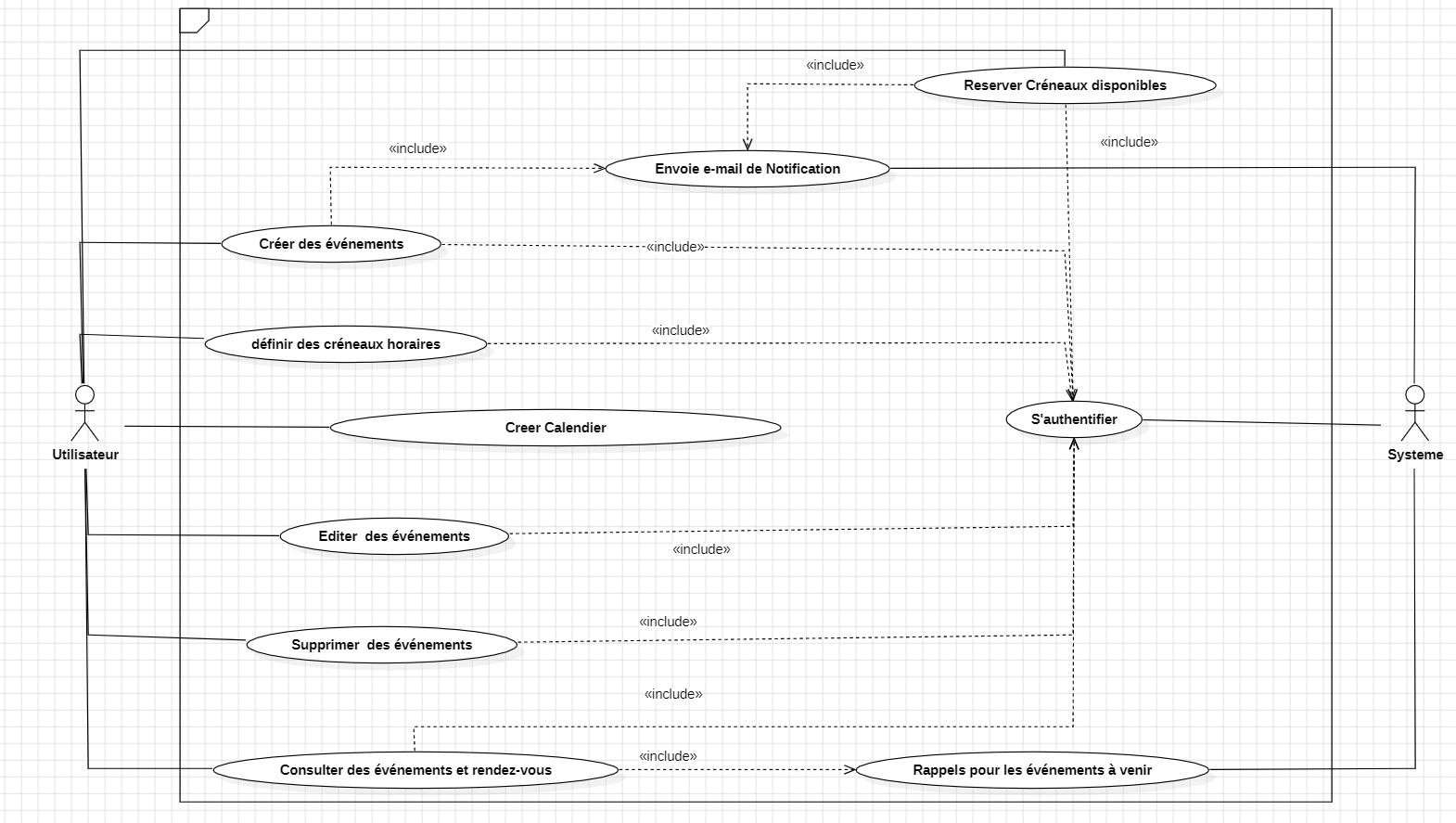


Figure 1 Calendrier use case

* **Description textuelle des cas d’utilisation**

1. Enregistrer un utilisateur:

- Acteur principal : Utilisateur

- Préconditions : L'utilisateur accède à l'application pour la première fois.

- Postconditions: L'utilisateur est enregistré dans la base de données.

- Scénario principal :

1. L'utilisateur accède à la page d'inscription.

2. L'utilisateur saisit ses informations de base (nom, prénom, adresse e-mail, mot de passe, etc.).

3. L'utilisateur valide le formulaire.

4. Le système vérifie les informations et enregistre l'utilisateur dans la base de données.

5. L'utilisateur reçoit une confirmation d'inscription.

2. Authentifier un utilisateur :

- Acteur principal : Utilisateur

- Préconditions : L'utilisateur a déjà un compte enregistré dans l'application.

- Postconditions : L'utilisateur est connecté à son compte.

- Scénario principal :

1. L'utilisateur accède à la page de connexion.

2. L'utilisateur saisit son adresse e-mail et son mot de passe.

3. L'utilisateur valide le formulaire.

4. Le système vérifie les informations et authentifie l'utilisateur.

5. L'utilisateur est redirigé vers son calendrier.

3. Afficher calendrier:

- Acteur principal: Utilisateur

- Préconditions: L'utilisateur est authentifié.

- Postconditions: Le calendrier de l'utilisateur est affiché avec les événements correspondants.

- Scénario principal:

1. L'utilisateur accède à son tableau de bord.

2. Le système charge les données du calendrier de l'utilisateur.

3. Le système affiche le calendrier avec les événements et rendez-vous.

4. Créer événement:

- Acteur principal: Utilisateur

- Préconditions: L'utilisateur est authentifié et accède à son calendrier.

- Postconditions: Le nouvel événement est ajouté au calendrier de l'utilisateur.

- Scénario principal:

1. L'utilisateur sélectionne l'option pour créer un nouvel événement.

2. L'utilisateur saisit les détails de l'événement (titre, date, heure, description, etc.).

3. L'utilisateur valide la création de l'événement.

4. Le système enregistre l'événement dans la base de données.

5. Le système met à jour le calendrier de l'utilisateur pour afficher le nouvel événement.

5. Éditer événement:

- Acteur principal: Utilisateur

- Préconditions: L'utilisateur est authentifié et accède à son calendrier.

- Postconditions: Les modifications apportées à l'événement sont enregistrées.

- Scénario principal:

1. L'utilisateur sélectionne l'événement à modifier dans son calendrier.

2. L'utilisateur accède à l'option de modification de l'événement.

3. L'utilisateur modifie les détails de l'événement selon ses besoins.

4. L'utilisateur valide les modifications.

5. Le système met à jour les informations de l'événement dans la base de données.

6. Supprimer événement:

- Acteur principal: Utilisateur

- Préconditions: L'utilisateur est authentifié et accède à son calendrier.

- Postconditions: L'événement est retiré du calendrier de l'utilisateur.

- Scénario principal:

1. L'utilisateur sélectionne l'événement à supprimer dans son calendrier.

2. L'utilisateur accède à l'option de suppression de l'événement.

3. L'utilisateur confirme la suppression de l'événement.

4. Le système supprime l'événement de la base de données.

5. Le système met à jour le calendrier de l'utilisateur pour refléter la suppression de l'événement.

7. Réserver créneau:

- Acteur principal: Utilisateur

- Préconditions: L'utilisateur est authentifié et accède à la section des créneaux horaires disponibles.

- Postconditions: Le créneau est réservé et enregistré dans le système.

- Scénario principal:

1. L'utilisateur consulte les créneaux horaires disponibles.

2. L'utilisateur sélectionne un créneau qui lui convient.

3. L'utilisateur valide la réservation du créneau.

4. Le système enregistre la réservation dans la base de données.

5. Le système envoie une confirmation de réservation à l'utilisateur.

## Analyse des cas d’utilisation

* Le diagramme de séquence d’enregistrer un utilisateur

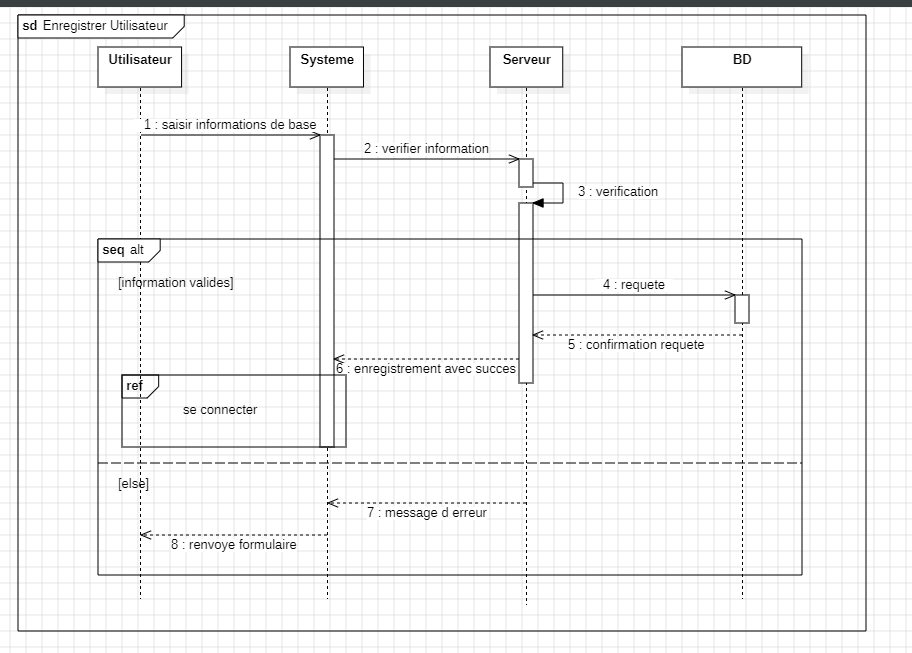


Figure 2 Enregistrer Utilisateur

* Le diagramme de séquence d’Authentification

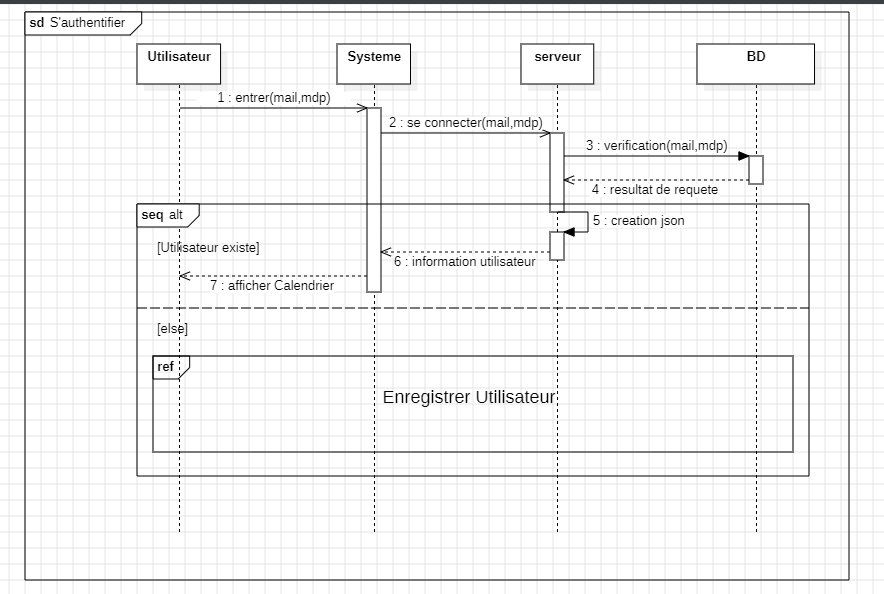


Figure 3 S'authentifier

* Le diagramme de séquence d’afficher calendrier

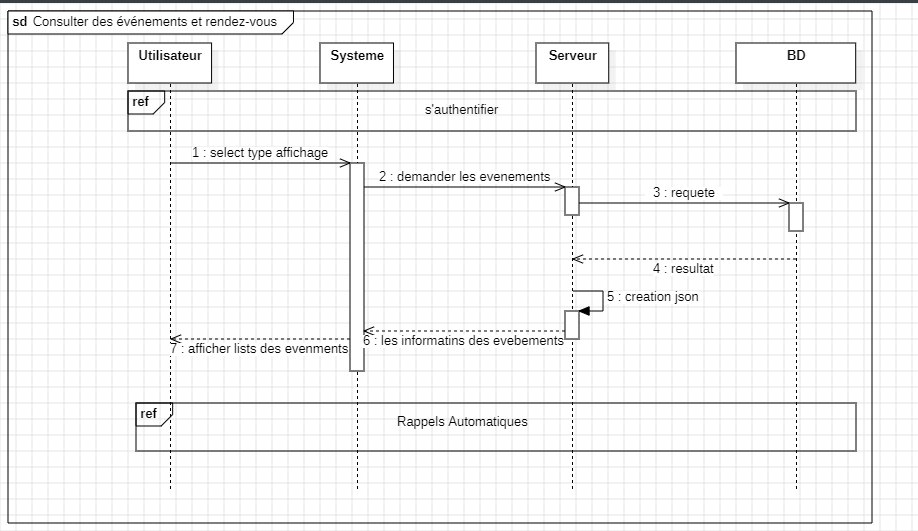


Figure 4 afficher calendrier

* Le diagramme de séquence d’crée événement

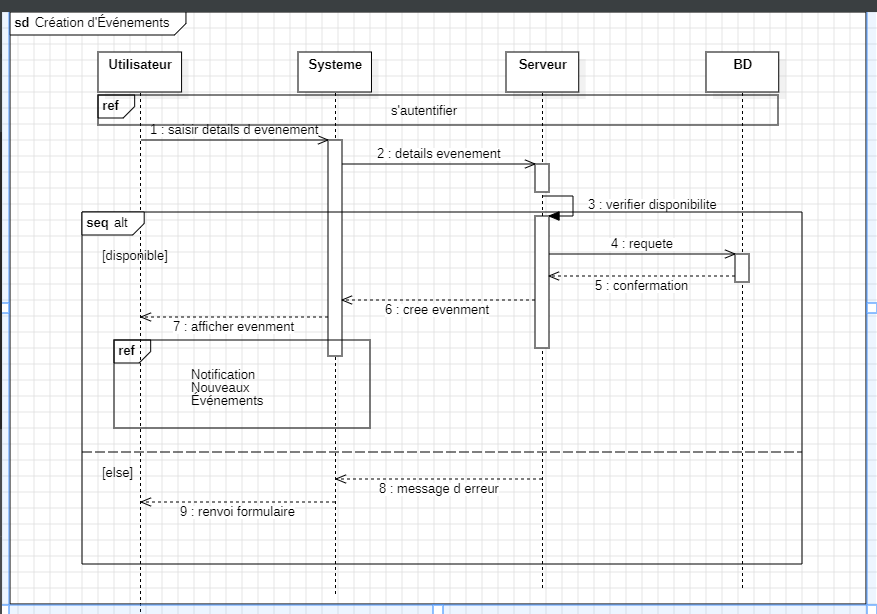


Figure 5 créer event

* Le diagramme de séquence d’editer evenement

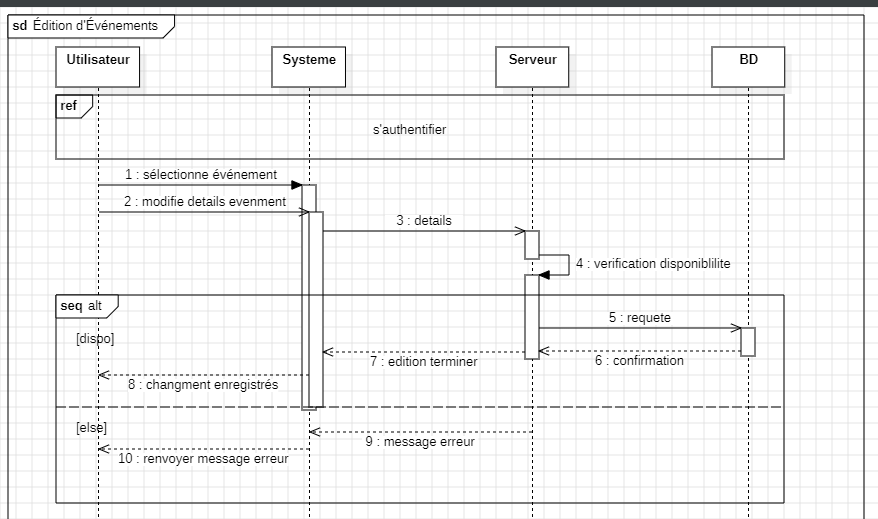


Figure 6 edition event

* Le diagramme de séquence d’supprimer evenement

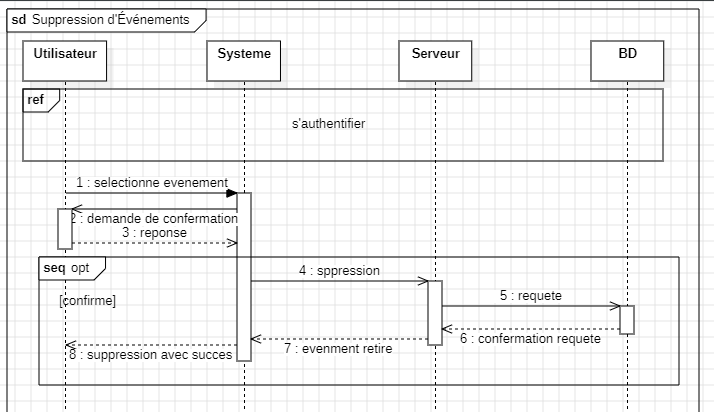


Figure 7 supprition event

* Le diagramme de séquence d’reserver créneau

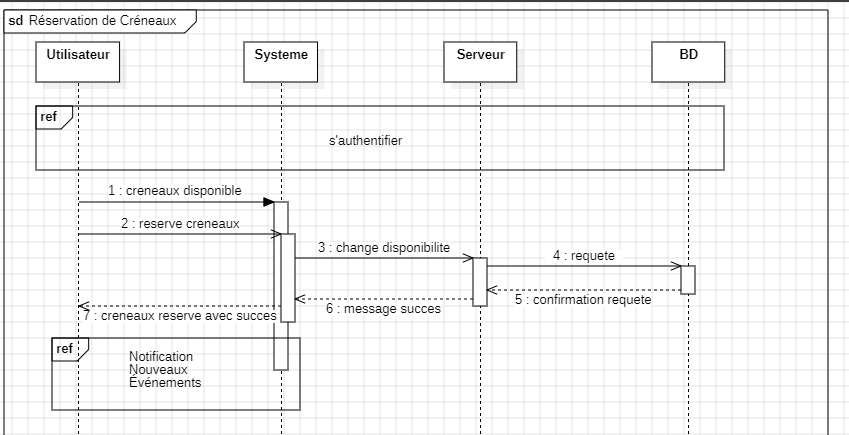


Figure 8 reservation creneau

En conclusion de ce chapitre, la modélisation comportementale de l'application de calendrier a permis d'identifier les actions principales des utilisateurs ainsi que les interactions avec le système. À travers l'identification des acteurs, la présentation du diagramme de cas d'utilisation, et l'analyse des cas d'utilisation les plus pertinents, nous avons clarifié les fonctionnalités essentielles de l'application.

# Chapitre 3 : Modélisation structurelle de l’application

Ce chapitre vise à répondre à la question "Comment faire" en présentant la conception détaillée de la solution proposée pour l'application de calendrier. Il comprend la représentation du diagramme de classes ainsi que la modélisation de la base de données.

## Diagramme de classes

### Descriptif du diagramme

**Classe Utilisateur:**

Cette classe représente les utilisateurs de l'application de calendrier.

Chaque utilisateur est identifié par un numéro unique (id\_utilisateur).

Les attributs nom, prénom, email et mdp stockent les informations personnelles de l'utilisateur.

Cette classe permet de gérer les comptes des utilisateurs, y compris leur inscription et leur authentification.

**Classe Evenement:**

Cette classe représente les événements ajoutés au calendrier par les utilisateurs.

Chaque événement est identifié par un numéro unique (id\_evenement).

Les attributs titre, description, date\_heure\_debut et date\_heure\_fin définissent les détails de l'événement.

**Classe Creneau:**

Cette classe représente les créneaux horaires disponibles pour la réservation.

Chaque créneau est identifié par un numéro unique (id\_creneau).

Les attributs date\_heure\_debut et date\_heure\_fin définissent le début et la fin du créneau.

**Classe association Reservation:**

Cette classe représente les réservations de créneaux horaires par les utilisateurs.

Chaque réservation est identifiée par un numéro unique (id\_reservation).

**Classe Notification:**

Cette classe représente les notifications envoyées aux utilisateurs pour les informer des événements ou des rappels.

Chaque notification est identifiée par un numéro unique (id\_notification).

L'attribut date\_heure\_notification spécifie le moment où la notification est envoyée.

### Représentation du diagramme

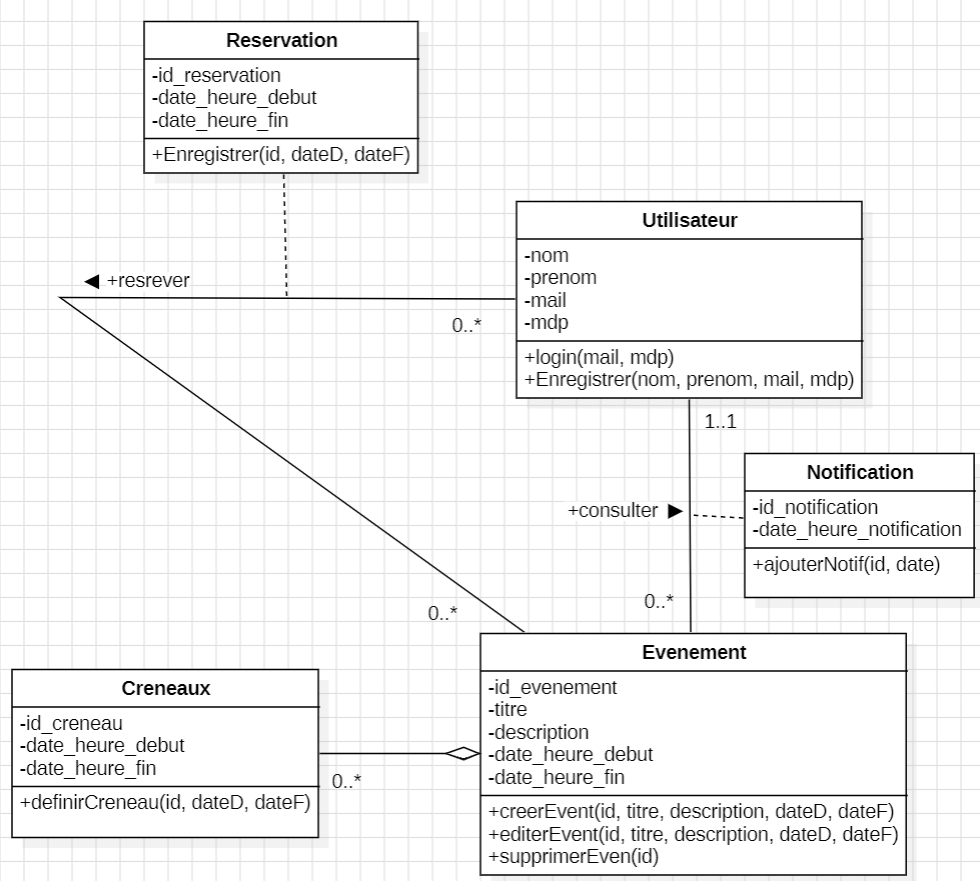


Figure 9 diagramme de classe

## Modélisation de la base de donnée

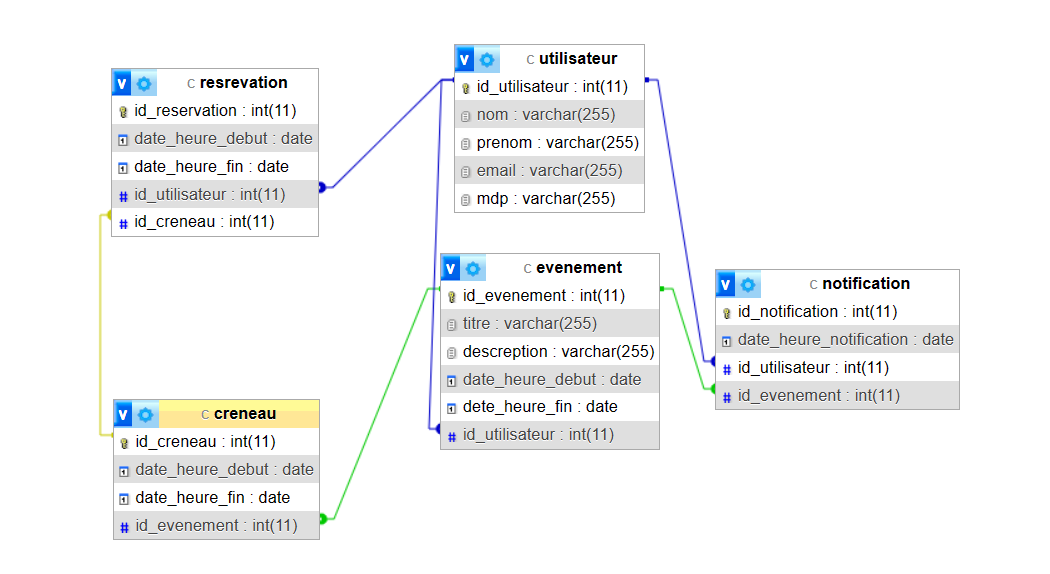


Figure 10 modele relationnel

* **utilisateur (** id\_utilisateur , nom , prenom , email , mdp )
* **evenement (** id\_evenement , titre , description , date\_heure\_debut , date\_heure\_fin , #id\_utilisateur )
* **creneau (** id\_creneau , date\_heure\_debut , date\_heure\_fin , #id\_evenement )
* **reservation (** id\_reservation , #id\_utilisateur , #id\_creneau , date\_heure\_debut , date\_heure\_fin )
* **notification (** id\_notification , #id\_utilisateur , #id\_evenement , date\_heure\_notification )

# Chapitre 4 : Réalisation

Dans ce chapitre, nous donnons vie à l'application de calendrier de BMG en dévoilant son écosystème de développement et ses interfaces graphiques principales. Ces éléments convergent pour concrétiser notre vision, transformant les concepts en une expérience utilisateur tangible.

## Environnement de développement

### Environnement matérielle

|  |  |
| --- | --- |
| Description matérielle | |
| Marque | Dell |
| Processeur | **Intel Core i5-13450HX**(Up to **4.60 GHz Turbo max, 20 Mo**de mémoire cache,**10-Cores**) |
| Disque dur | 512 GO SSD |
| Mémoire vive | **DDR5-4800 MHz** |
| Carte graphique | **NVIDIA GeForce RTX3050** (**6Go** de mémoire dédiée **GDDR6**) |

### Environnement logiciel

|  |  |
| --- | --- |
| Description logicielle | |
| Système d’exploitation | **Windows 11 Famille** |
| Serveur base de donnée | Postgresql |
| Outils de modélisation | StarUML |
| Outils de développement | Visual Studio code |
| Technologies | Node js , Express |
| Outils de Test | Postman |

* **Node.js** est un environnement d'exécution JavaScript côté serveur, basé sur le moteur JavaScript V8 de Google Chrome. Il permet d'exécuter du code JavaScript côté serveur, plutôt que dans un navigateur web. Node.js est conçu pour être événementiel et non bloquant, ce qui le rend efficace pour les applications en temps réel avec une grande quantité d'I/O (Entrées/Sorties), telles que les applications web, les serveurs d'API, les applications de messagerie instantanée, les jeux en ligne, etc.



Figure 11 node js

* **Express.js** est un framework web pour Node.js, qui est un environnement d'exécution JavaScript côté serveur. Express.js simplifie le processus de création d'applications web et d'API en fournissant une interface simple et flexible pour gérer les requêtes HTTP, les routes, les cookies, les sessions, etc.



Figure 12 express js

* **PostgreSQL** est un système de gestion de base de données relationnelle (SGBDR) open source très puissant et robuste. Il est également connu sous le nom de "Postgres". Conçu pour offrir une fiabilité élevée, la conformité aux standards SQL et la prise en charge des fonctionnalités avancées, PostgreSQL est largement utilisé dans de nombreuses applications et entreprises à travers le monde.



Figure 13 postgreSQL

* **Postman** est un outil populaire utilisé par les développeurs pour tester, développer et documenter des API (Application Programming Interfaces). Il offre une interface utilisateur conviviale qui permet aux développeurs de créer, partager, tester et déboguer des API plus facilement.



Figure 14 postman

* **VS Code**, abréviation de **Visual Studio Code**, est un éditeur de code source développé par Microsoft. C'est un outil populaire utilisé par de nombreux développeurs pour écrire, éditer et déboguer du code.



Figure 15 vs code

* StarUML est une application de modélisation UML (Unified Modeling Language) largement utilisée pour créer des diagrammes de modèles logiciels.

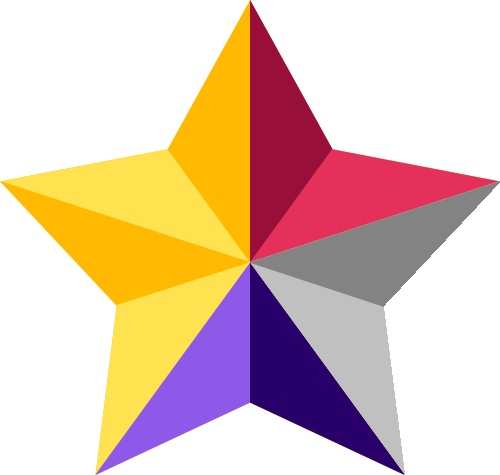


Figure 16 starUML

* Windows 11 est la dernière version majeure du système d'exploitation Microsoft Windows, succédant à Windows 10. Annoncé en juin 2021 et officiellement publié en octobre 2021, Windows 11 présente plusieurs changements visuels, fonctionnels et de performances par rapport à son prédécesseur.



Figure 17 interface windows 11

## Principales interfaces graphiques

Dans cette partie, je vais décrire les principales fonctionnalités des API que j'ai développées pour l'application de calendrier de BMG. Bien que mon implication se concentre sur la conception et la mise en œuvre du backend, il est important de noter qu'aucune interface graphique n'a été développée dans le cadre de ce projet. Mes efforts se sont concentrés sur la création d'un backend robuste et fonctionnel, fournissant les fonctionnalités nécessaires pour gérer les événements et les rendez-vous de manière efficace, même en l'absence d'une interface utilisateur.

* Dans cette section, nous présentons la manière de se connecter à la base de données ainsi que les requêtes nécessaires et les routes d'API formées, ainsi que les fonctions associées.

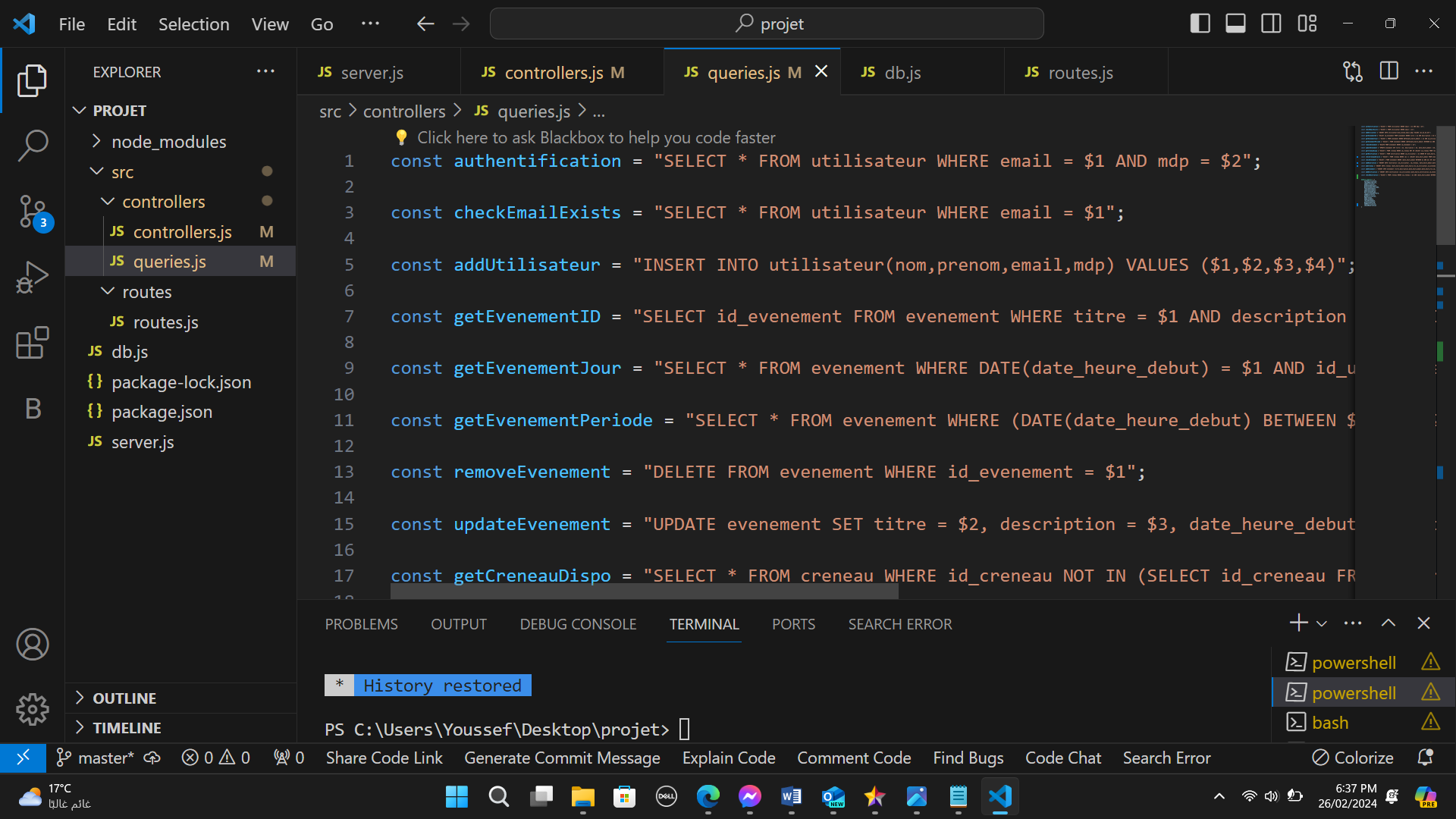


Figure 18 les requetes

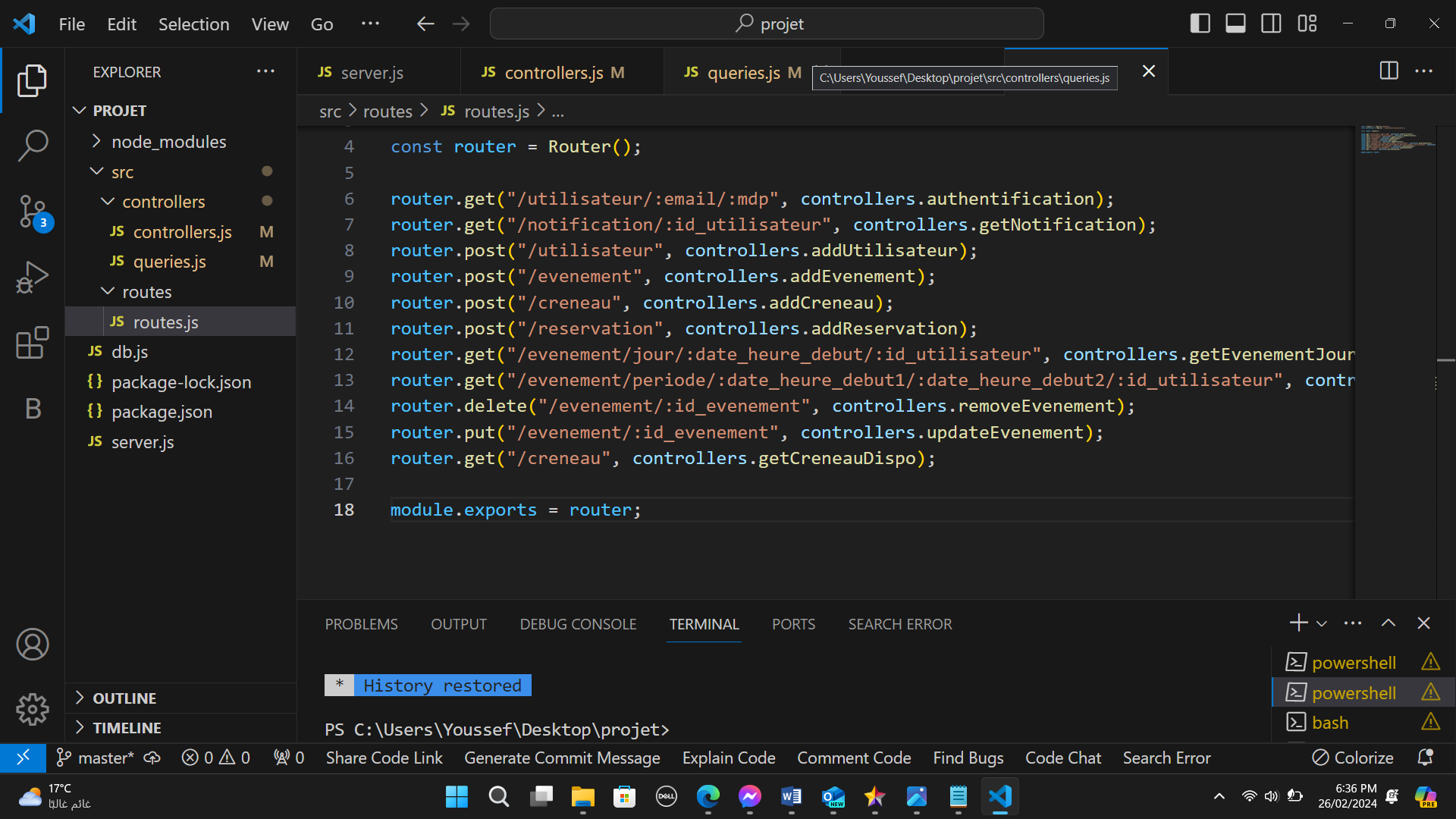


Figure 19 les routes

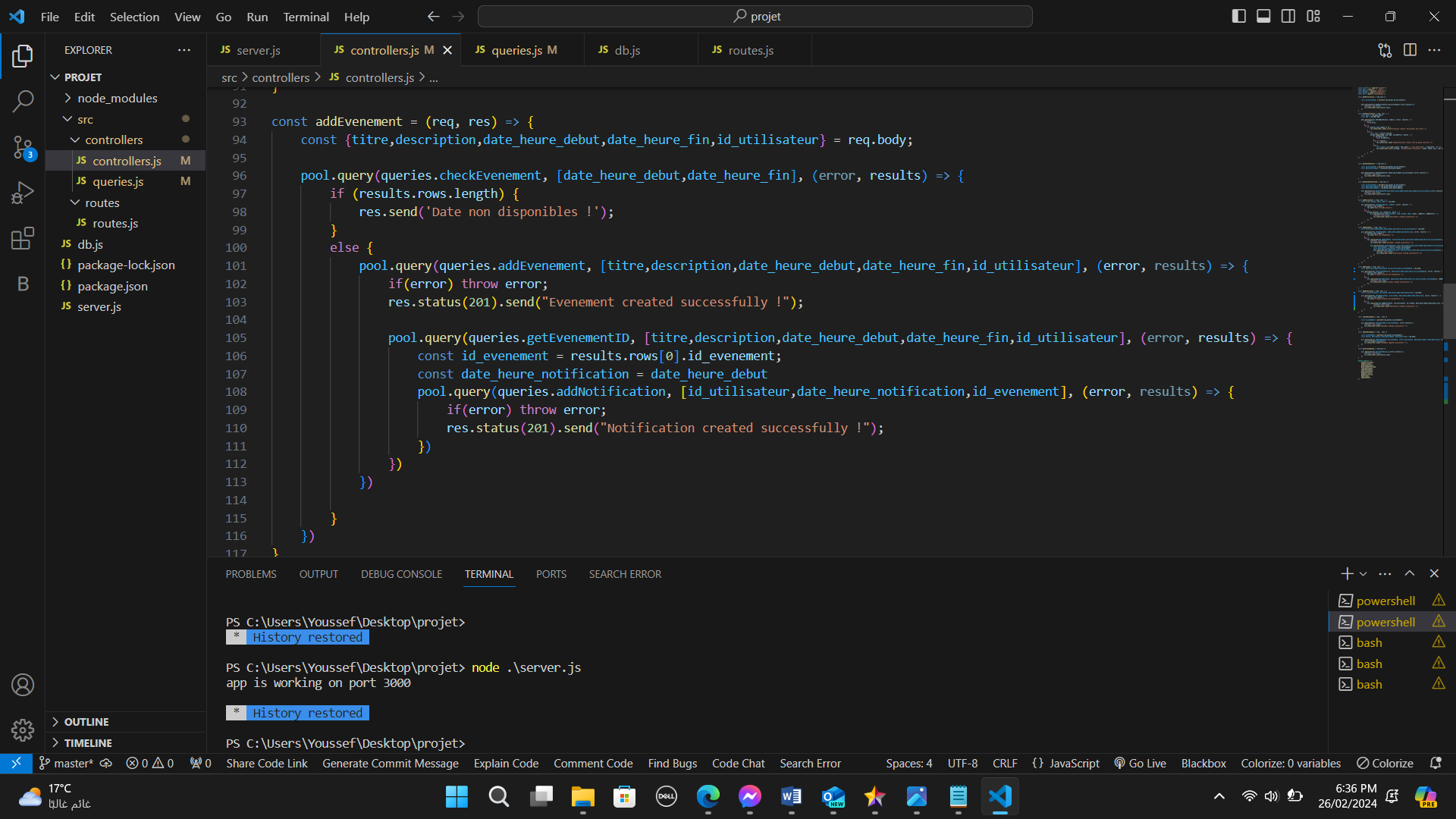
****

Figure 20 les fonctions

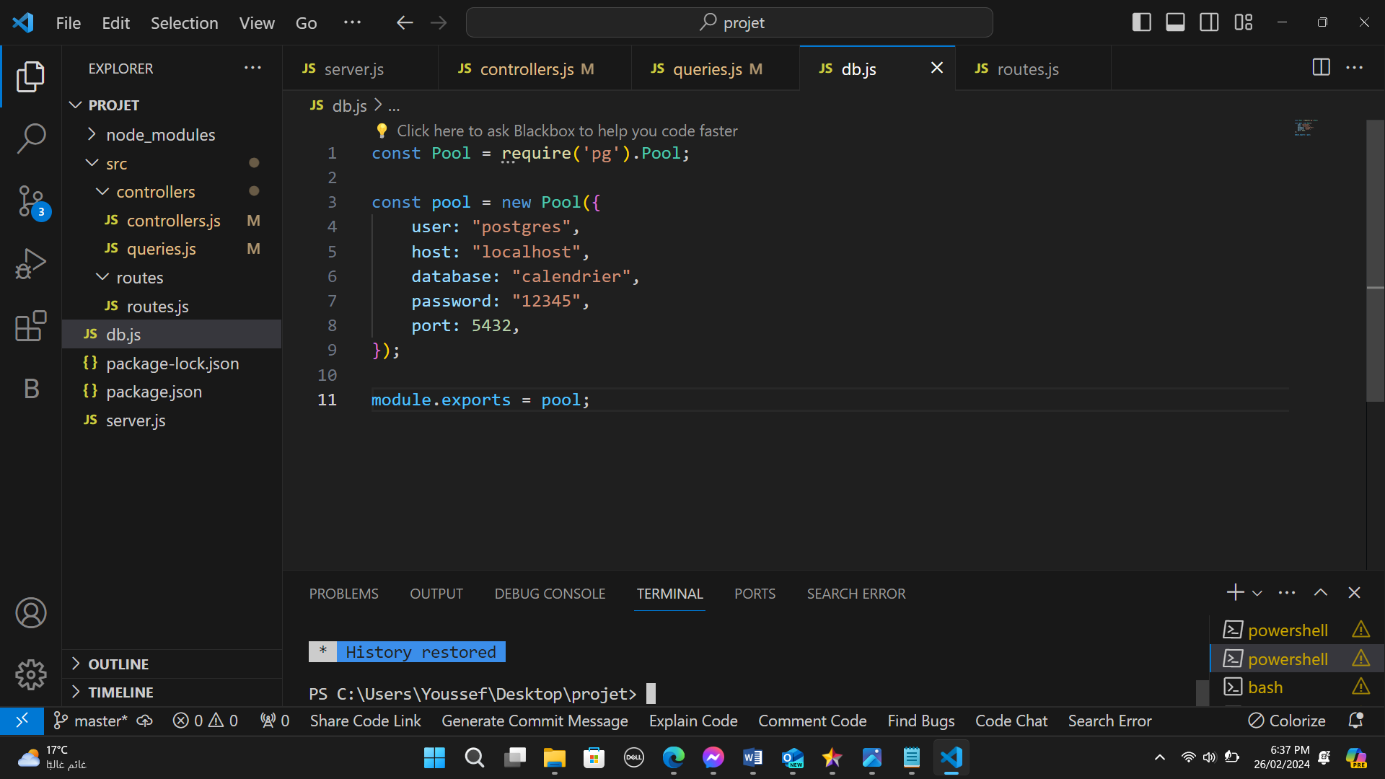


Figure 21 data base connection

* Maintenant, nous procédons à l'ouverture du serveur afin de tester nos APIs.

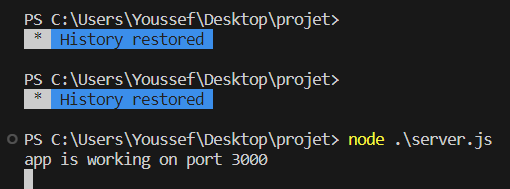


Figure 22 server open

* Un exemple de test d'API avec Postman.

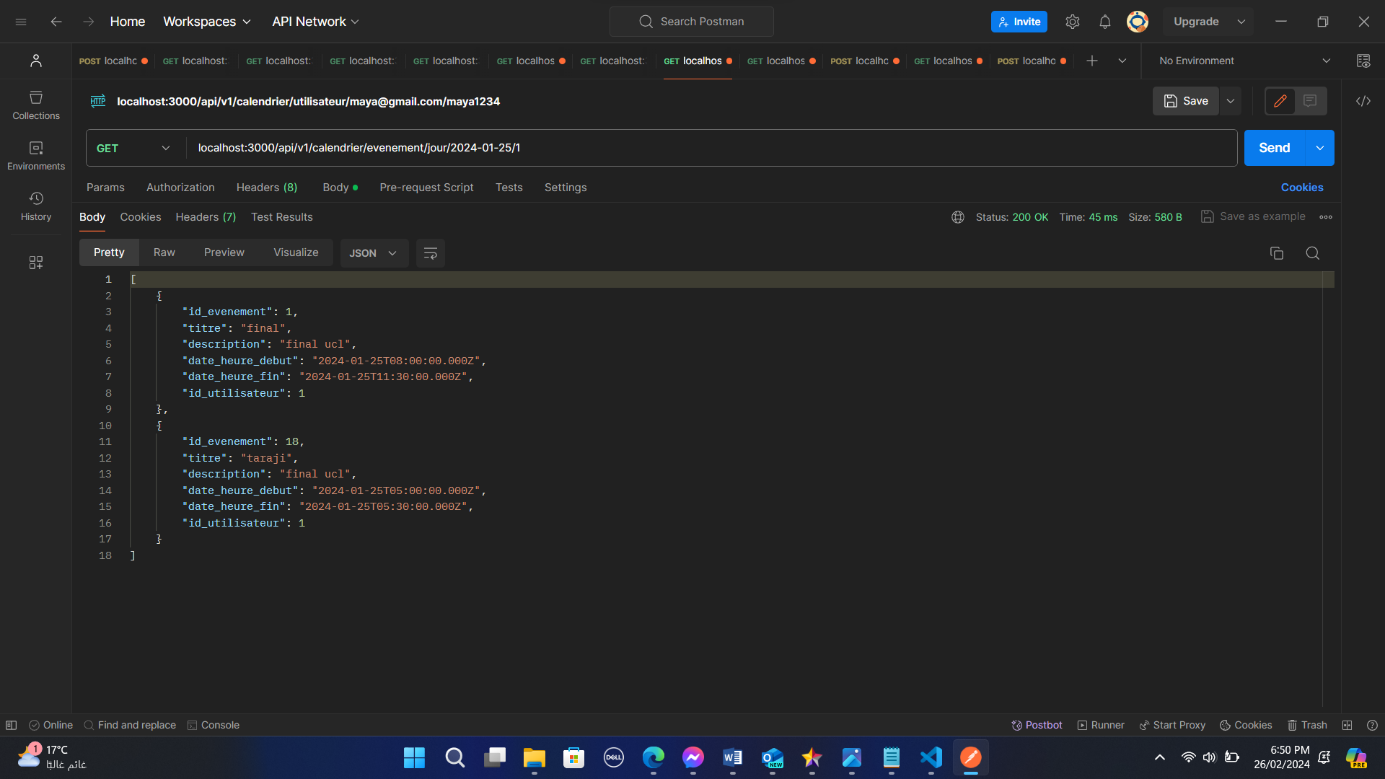


Figure 23 test un API

l'application de calendrier de BMG en présentant son développement et en exposant ses principales interfaces graphiques. Malgré l'absence d'une interface utilisateur, l'accent a été mis sur la création d'un backend robuste et fonctionnel, mettant en lumière les requêtes, les routes d'API et les tests effectués avec succès.

# Conclusion générale

En conclusion, ce rapport a présenté en détail le processus de développement de l'application de calendrier pour BMG. À travers les différents chapitres, nous avons exploré le contexte du projet, identifié les besoins spécifiques de l'entreprise, et proposé une solution personnalisée pour répondre à ces besoins.

La modélisation comportementale et structurelle de l'application a permis de définir clairement les fonctionnalités et l'architecture de la solution. Nous avons utilisé des outils et des techniques appropriés pour concevoir et développer le backend de l'application, en mettant l'accent sur la robustesse, la sécurité et la performance.

Bien que l'absence d'une interface utilisateur graphique ait été notée, notre travail sur le backend a été rigoureux et complet. Les APIs développées ont été testées avec succès, et notre environnement de développement a été soigneusement configuré pour assurer une productivité optimale.

En fin de compte, ce projet représente une étape importante dans la transformation numérique de BMG, fournissant une solution moderne et efficace pour la gestion des calendriers. Notre engagement envers la qualité et l'innovation nous a permis de relever les défis techniques et de livrer une application qui répond aux attentes de l'entreprise et des utilisateurs.

# Nétographie

[1] [Node.js (nodejs.org)](https://nodejs.org/en) – 18/01/2024

[2] [Node.js Tutorial (w3schools.com)](https://www.w3schools.com/nodejs/) – 18/01/2024

[3] [PostgreSQL: The world's most advanced open source database](https://www.postgresql.org/) – 23/01/2024

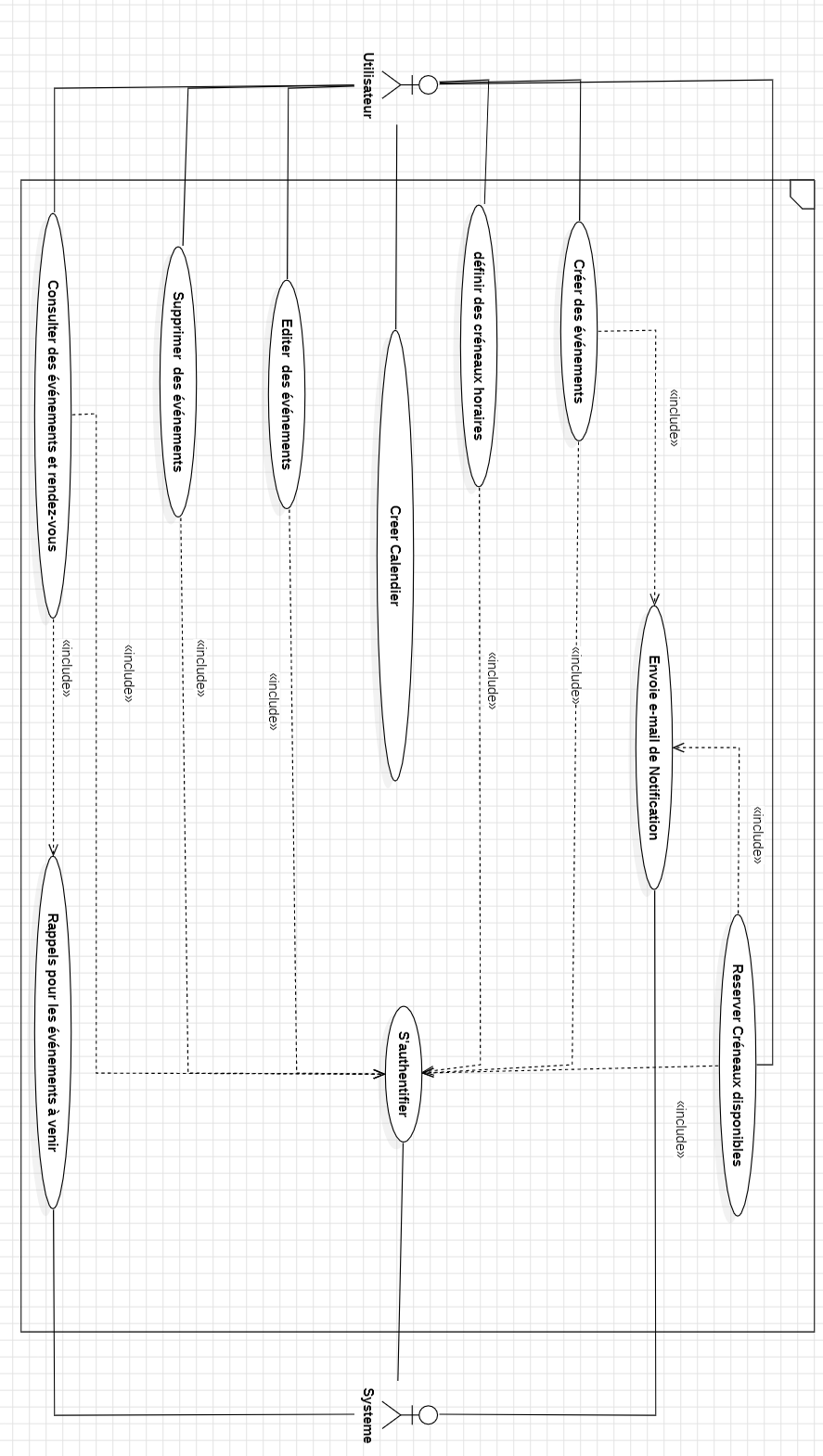
[4] [PostgreSQL: Tutorials & Other Resources](https://www.postgresql.org/docs/online-resources/) – 23/01/2024

[5] [Password hashing in Node.js with bcrypt - LogRocket Blog](https://blog.logrocket.com/password-hashing-node-js-bcrypt/) – 25/01/2024

[6] [Understanding JWT Authentication With Node.js [Updated] (simplilearn.com)](https://www.simplilearn.com/tutorials/nodejs-tutorial/jwt-authentication) – 26/01/2024

[7] [Postman API Platform | Sign Up for Free](https://www.postman.com/) – 26/01/2024

# Annexes

* Diagramme de cas d’utilisation