

Année universitaire

**2022 /2023**

**Application java pour la gestion d’une messagerie**

**Encadré par :**

**Mme Jihane Lakhrouit**

**Réalisée par :**

**SOUMAIR MOKHTAR**

**LAMDAKKAR OUSSAMA**

**SONDE OUBAIDA**

**BIBANG NANG ARMAND**

**RAPPORT DE PROJET**

**Table de matières**

[Introduction 4](#_Toc127742399)

[I. Contexte théorique 5](#_Toc127742400)

[II. Etude de l’existant 7](#_Toc127742401)

[III. Cahier de charges et modélisation 8](#_Toc127742402)

[1 .Cahier de charges 8](#_Toc127742403)

[2.Modélisations 9](#_Toc127742404)

[IV. Implémentation 11](#_Toc127742405)

[1 .L'authentification : 11](#_Toc127742406)

[2. Erreur de connexion : 12](#_Toc127742407)

[3. L’Inbox : 14](#_Toc127742408)

[4. Sent 14](#_Toc127742409)

[5. L'envoi : 16](#_Toc127742410)

[Conclusion 17](#_Toc127742411)

# Introduction

Une application de messagerie est une application logicielle utilisée pour communiquer avec d'autres utilisateurs sur internet. Son but est d'envoyer et de recevoir des courriers électroniques (courriels) avec des notifications .  
Le courrier électronique repose sur le concept du client/serveur.  
Ce projet est développé en utilisant JAVA et quelques API de Java.  
  
Une API, ou interface de programmation d'application, est un ensemble de définitions et de protocoles qui facilite la création et l'intégration des applications.  
  
L'API JavaMail est un cadre de création d'applications clients Java pour le courrier et la messagerie qui est indépendant de la plate-forme et du protocole. Il propose des classes abstraites comprenant des objets créés dans le système de messagerie en tant qu'interface générique pour les programmes de messagerie.

# Contexte théorique

Dans le contexte de la création d'une application de messagerie développée avec Java, il est important de comprendre les concepts clés liés à la programmation orientée objet (POO), à la gestion de la mémoire et aux protocoles de communication.

* En ce qui concerne la POO, Java est un langage de programmation qui se base sur ce concept. La POO est un paradigme de programmation qui permet de modéliser les objets du monde réel en utilisant des concepts tels que les classes, les objets, les attributs et les méthodes. Dans une application de messagerie, les objets pourraient être des utilisateurs, des messages et des salles de conversation, tandis que les méthodes pourraient inclure la création de nouveaux utilisateurs ou l'envoi de messages.
* La gestion de la mémoire est également un aspect important de la programmation en Java. Étant donné que Java utilise une machine virtuelle (JVM) pour exécuter le code, il est important de comprendre comment la mémoire est gérée par la JVM. La JVM utilise un processus appelé Garbage collection pour libérer la mémoire qui n'est plus utilisée par une application. Il est important de tenir compte de la gestion de la mémoire lors de la création d'une application de messagerie pour s'assurer qu'elle fonctionne efficacement et ne gaspille pas de ressources.
* Enfin, les protocoles de communication sont essentiels pour permettre à l'application de messagerie de communiquer avec d'autres applications ou serveurs. Les protocoles couramment utilisés dans les applications de messagerie incluent TCP/IP, HTTP et WebSocket. Il est important de comprendre comment ces protocoles fonctionnent et comment les utiliser efficacement pour assurer une communication fluide entre les différents composants de l'application de messagerie.

En somme, le contexte théorique pour le développement d'une application de messagerie en Java comprend des concepts clés de la POO, de la gestion de la mémoire et des protocoles de communication.

# Etude de l’existant

Outlook est une application de messagerie électronique développée par Microsoft qui offre des fonctionnalités de messagerie, de calendrier et de gestion des contacts. Voici quelques éléments clés que vous pourriez étudier pour comprendre l'existant de Outlook :

* Interface utilisateur :

Outlook a une interface utilisateur intuitive et conviviale qui permet aux utilisateurs de naviguer facilement dans les différentes fonctionnalités de l'application.

* Intégration avec les autres applications de Microsoft : Outlook est intégré avec d'autres applications de Microsoft, telles que Word, Excel et PowerPoint, ce qui permet aux utilisateurs d'accéder facilement à ces applications à partir de la messagerie électronique.
* Fonctionnalités de calendrier : Outlook offre des fonctionnalités de calendrier avancées qui permettent aux utilisateurs de planifier des réunions, de suivre les rendez-vous et les événements importants, et de partager leur calendrier avec d'autres utilisateurs.
* Gestion des contacts : Outlook offre des fonctionnalités de gestion des contacts qui permettent aux utilisateurs de stocker et de gérer les informations de contact, de créer des listes de distribution, et d'envoyer des e-mails à des groupes de contacts.
* Fonctionnalités de sécurité : Outlook utilise une variété de fonctionnalités de sécurité, telles que la vérification en deux étapes, la protection contre les logiciels malveillants, et la gestion des permissions, pour protéger les utilisateurs contre les menaces de sécurité en ligne.

En étudiant l'existant de Outlook, nous avons identifié les fonctionnalités clés que nous souhaitons inclure dans notre application de messagerie avec Java.

# Cahier de charges et modélisation

# Cahier de charges

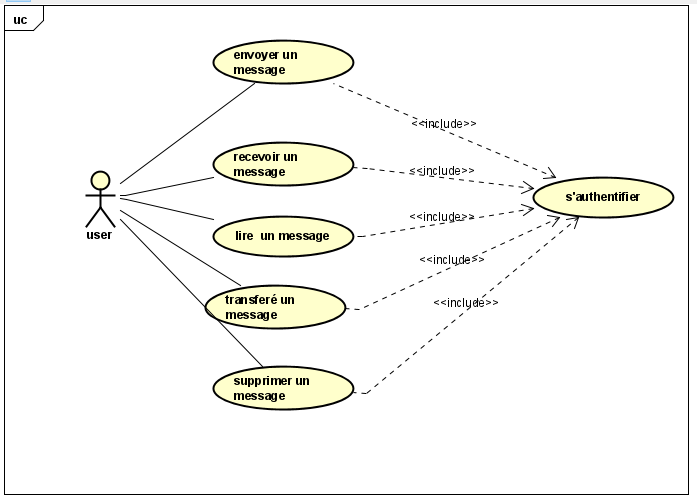
Avant de procéder au développement de l'application de messagerie, il est important de définir les besoins et les fonctionnalités clés qui seront incluses. La première exigence est que l'application doit être capable de prendre en charge les communications entre les employés de l'entreprise, ainsi que les communications externes avec les clients. La messagerie doit permettre la transmission de messages écrits, d'images, de vidéos et de fichiers.

En outre, l'application doit être conçue de manière à garantir la confidentialité des messages échangés.  
Dans cette application l’utilisateur qui est caractérisé par un identifiant, un mot de passe, et un email ; doit s’authentifier avant toute action.  
Après l’authentification l’utilisateur peut recevoir ou envoyer des messages.  
Un message est caractérisé par un identifiant, un objet, corp du message et un statu. Un message est envoyé et reçu à une date précise.

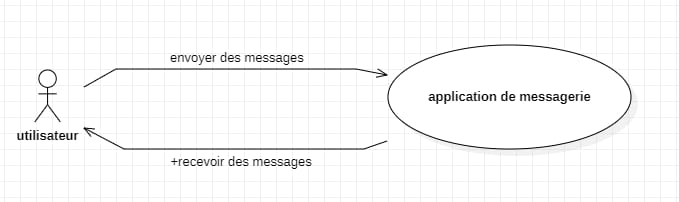
Enfin, l'application doit être simple d'utilisation pour permettre aux utilisateurs de la prendre en main rapidement.

# Modélisations

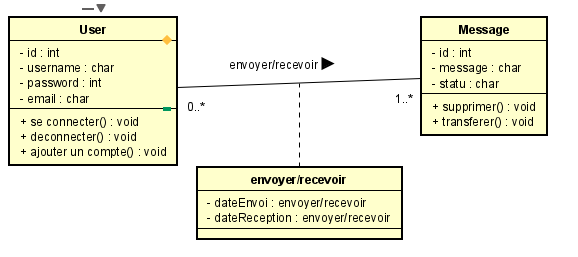
* Diagramme de Use Case



* Diagramme de contexte



* Diagramme de classe

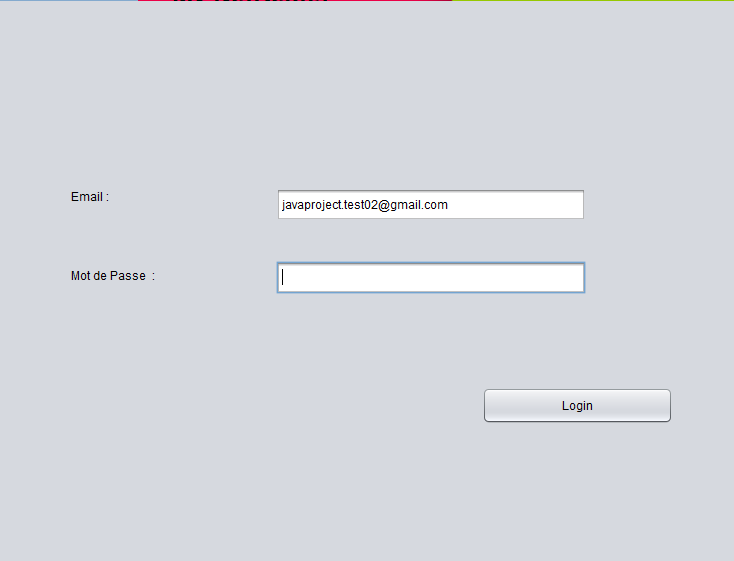


# Implémentation

# L'authentification :

L'authentification est la première étape que les utilisateurs de notre application doivent franchir pour accéder à leur compte de messagerie. Cette étape consiste à vérifier si les informations de connexion (nom d'utilisateur et mot de passe) fournies par l'utilisateur sont correctes. Pour mettre en place l'authentification, nous avons créer une interface utilisateur qui permet aux utilisateurs de se connecter. Cette interface est conviviale et facile à comprendre pour les utilisateurs.

Les informations sont stockées de manière sécurisée dans une base de données. Pour la connexion, l'utilisateur doit fournir son email et son mot de passe. Les informations d'authentification sont vérifiées en les comparant à celles stockées dans la base de données. Si les informations sont correctes, l'utilisateur sera autorisé à accéder à son compte de messagerie.



# Erreur de connexion :

Lorsque les utilisateurs rencontrent des erreurs de connexion, cela peut être frustrant et décourageant. Nous avons donc prévu des mesures pour gérer ces erreurs et aider les utilisateurs à résoudre les problèmes. Les erreurs de connexion courantes que les utilisateurs pourraient rencontrer incluent des erreurs d’email ou de mot de passe incorrect.

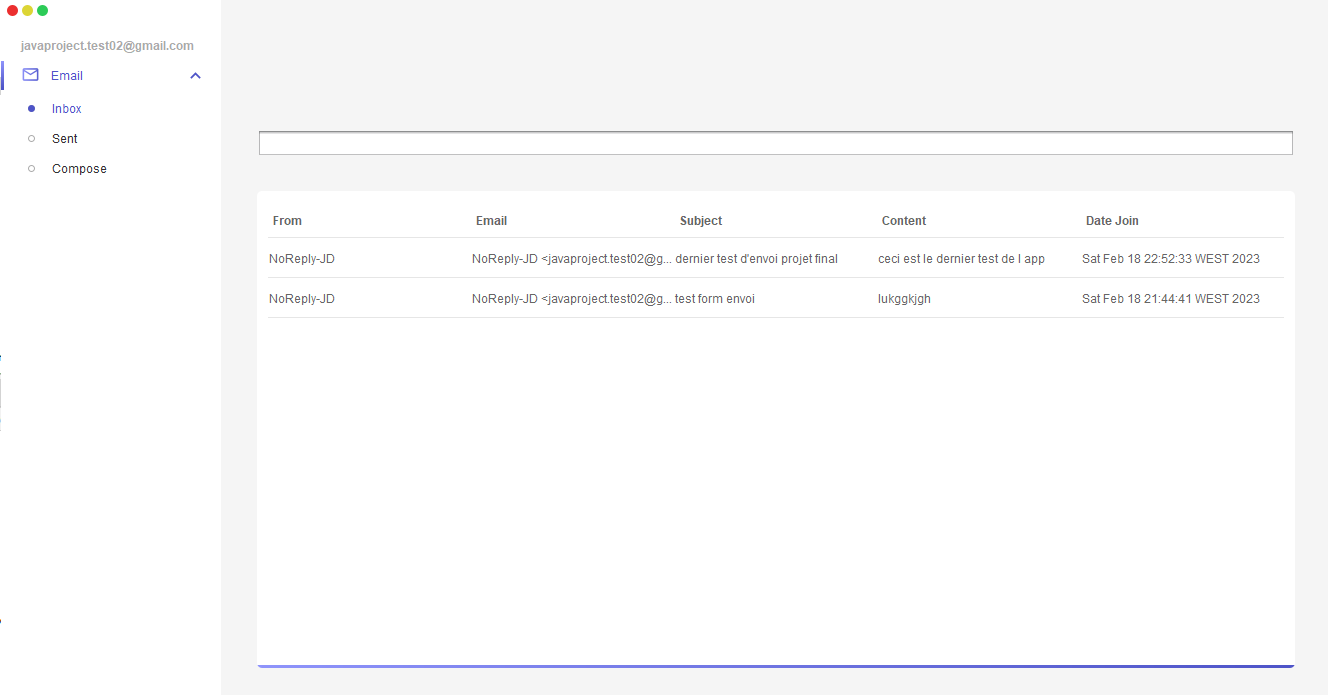
Pour gérer les erreurs de connexion, nous affichons un message d'erreur indiquant que les informations d'identification fournies sont

Incorrectes.



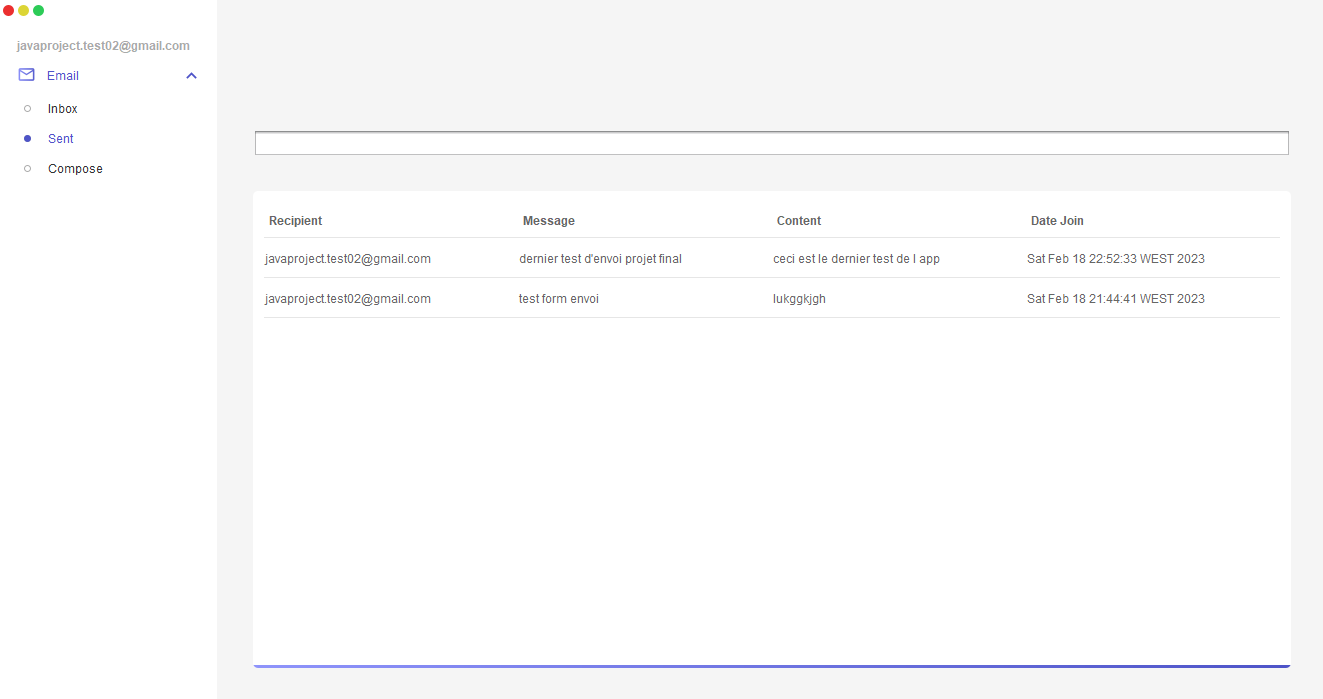
# L’Inbox :

L’Inbox est l'endroit où les utilisateurs peuvent voir tous les e-mails qu'ils ont reçus. Pour afficher les e-mails dans l’Inbox, nous avons stocker les e-mails dans une base de données et les afficher dans l'interface utilisateur de l’Inbox. Les e-mails peuvent être triés par ordre chronologique ou par ordre alphabétique en fonction de différents critères tels que la date, le destinataire, l'expéditeur ou le sujet.



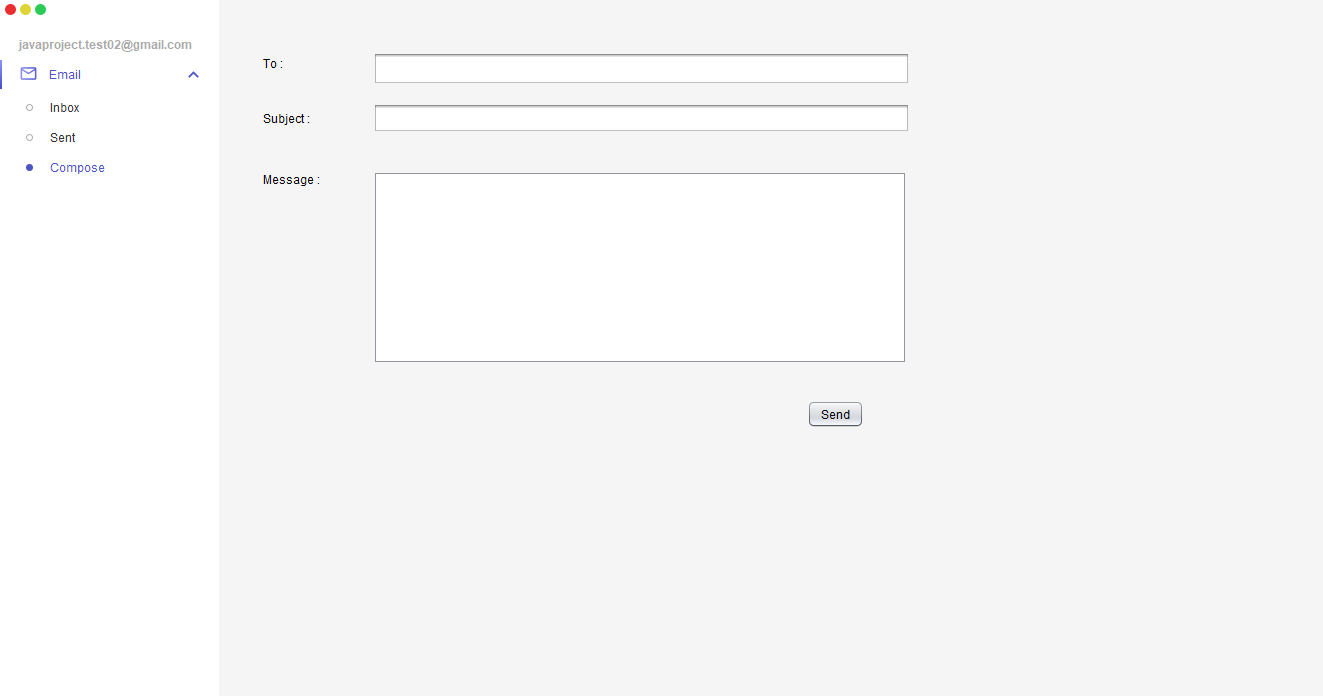
# Sent

Sent est l'endroit où les utilisateurs peuvent voir tous les e-mails qu'ils ont envoyés. Pour afficher les e-mails dans Sent, nous avons stocker les e-mails dans une base de données et les afficher dans l'interface utilisateur de l’Inbox. Les e-mails peuvent être triés par ordre chronologique ou par ordre alphabétique en fonction de différents critères tels que la date, le destinataire.



# L'envoi :

L'envoi est l'étape qui permet aux utilisateurs de composer et d'envoyer des e-mails à d'autres personnes. Pour composer un e-mail, les utilisateurs doivent fournir les informations suivantes : le destinataire, le sujet et le corps de l'e-mail.



# Conclusion

Le développement d'une application de messagerie avec Java a été un projet passionnant et stimulant. Ce projet nous a permis de mettre en pratique nos compétences en programmation Java et de développer des connaissances dans le domaine des applications de messagerie.

Au cours de ce projet, nous avons développé une application de messagerie qui permet aux utilisateurs d'envoyer et de recevoir des messages en temps réel, d'organiser leur boîte de réception, de se connecter et de se déconnecter de manière sécurisée et de gérer leur profil utilisateur. Nous avons également implémenté des fonctionnalités telles que la détection des erreurs de connexion.

Les résultats de ce projet sont significatifs pour les utilisateurs qui souhaitent communiquer de manière efficace et sécurisée. L'application de messagerie que nous avons développée offre une alternative fiable aux applications de messagerie existantes, avec une interface utilisateur intuitive et des fonctionnalités avancées.

Cependant, il y a des limites à ce projet. En particulier, le développement d'une application de messagerie sécurisée et fiable nécessite une expertise supplémentaire en matière de sécurité informatique et de cryptographie. De plus, le projet pourrait être amélioré en incluant des fonctionnalités supplémentaires telles que la récupération de mot de passe, la possibilité d'envoyer des pièces jointes, la recherche de messages, la possibilité de créer des groupes de discussion et la traduction automatique de messages.

Pour l'avenir, il est recommandé de continuer à améliorer l'application de messagerie en intégrant des fonctionnalités supplémentaires et en améliorant la sécurité. Il est également recommandé de poursuivre la recherche dans le domaine des applications de messagerie et de la communication en ligne pour améliorer la compréhension de ces domaines et développer de nouvelles approches.

En fin de compte, ce projet a été une expérience enrichissante et a renforcé nos compétences en programmation Java. J'espère que cette application de messagerie pourra bénéficier aux utilisateurs et contribuer à l'amélioration des pratiques en matière de communication en ligne.