

25 FEVRIER 2023

FST

# Gestion de Guichets Automatiques Bancaires (GAB)

Master 2 Projet S3

|                                    |        |
|------------------------------------|--------|
| MOHAMED ABDERRAHMANE AHMED EKBAR   | C13514 |
| SIDI ABDELLAH MOHAMED HASSANE EYEL | C13421 |
| KHADIJETOU KEBBADY SIDI MAHMOUD    | C13691 |
| MOURADE MOULAYE AHMED AHMED IDE    | C12220 |

## Table des matières

|   |           |
|---|-----------|
| <b>Introduction.....</b>  | <b>2</b>  |
| <b>1.Cadre général du projet .....</b>                          | <b>3</b>  |
| <b>1-1Les fonctionnalités principales de notre projet .....</b> | <b>3</b>  |
| <b>1-2 Méthode agiles.....</b>                                  | <b>3</b>  |
| <b>1-2-Services web .....</b>                                   | <b>5</b>  |
| <b>1-3design patterns .....</b>                                 | <b>6</b>  |
| <b>2-Etude conceptuelle.....</b>                                | <b>6</b>  |
| <b>3- Environnement de travail .....</b>                        | <b>11</b> |
| <b>3-1Les Outils utilisés : .....</b>                           | <b>11</b> |
| <b>3-2 Les frameworks utilisés : .....</b>                      | <b>14</b> |
| <b>4 Présentation des interfaces du système :.....</b>          | <b>16</b> |
| <b>4-Conclusion .....</b>                                       | <b>18</b> |

## Introduction

Le but de notre travail est de développer un système de distribution d'argent à l'aide d'une carte bancaire valide. Ce système permettra aux utilisateurs d'effectuer des opérations financières telles que le retrait, le dépôt, le virement, l'historique des transactions et la demande de chéquier.

Le système sera basé sur une architecture client-serveur et sera développé en utilisant une méthode agile pour garantir une livraison rapide et efficace du produit final. Le développement de ce projet impliquera la création d'un Back-end en utilisant Spring boot et d'un Front-end en utilisant Angular.

Les utilisateurs du système pourront se lancer l'application, saisir numero du carte bancaire, saisir leur code et choisir l'opération qu'ils souhaitent effectuer. Les opérations seront enregistrées sous forme de triplets (User ; Carte ; Description) contenant le nom de l'utilisateur, le numéro de la carte et le nom de l'opération. Lorsqu'une opération est effectuée, l'utilisateur pourra imprimer les détails de cette opération.

Notre objectif est de créer un système fiable, sécurisé et convivial pour les utilisateurs.

Nous sommes impatients de relever les défis de ce projet et de travailler en étroite collaboration avec l'équipe pour fournir un produit de qualité supérieure.

# 1. Cadre général du projet

## 1-1 Les fonctionnalités principales de notre projet

- **Authentification** : l'utilisateur doit s'authentifier avant de pouvoir réaliser une opération en utilisant son numéro de carte bancaire et un code confidentiel.
- **Retrait** : l'utilisateur peut retirer de l'argent de son compte en spécifiant le montant à retirer. L'application vérifie la disponibilité des fonds avant de valider l'opération.
- **Dépôt** : l'utilisateur peut déposer de l'argent sur son compte en spécifiant le montant à déposer. L'application ajoute le montant déposé au solde du compte de l'utilisateur
- **Virement** : l'utilisateur peut effectuer un virement en spécifiant le montant à transférer et le numéro de compte de destination. L'application vérifie la disponibilité des fonds avant de valider l'opération.
- **Historique** : l'utilisateur peut consulter la dernière opération effectuée sur son compte
- **Sécurité** : toutes les opérations sont sécurisées et les informations de l'utilisateur sont protégées.

Les mots de passe sont stockés de manière cryptée et les communications entre la frontend et l'API back-end sont sécurisées.

## 1-2 Méthode agiles

**Scrum** est une méthode agile de gestion de projet qui permet de gérer de manière itérative et incrémentale le développement de produits complexes. Elle repose sur un cadre de travail collaboratif qui implique le client et l'équipe de développement dans la planification, l'exécution et l'évaluation des itérations de développement appelées **Sprints**.

**Comment Scrum peut-il être appliqué au projet GAB ?** Dans le cadre de ce projet, Scrum est utilisé pour :

- Planifier les itérations de développement (Sprints) en fonction des besoins des utilisateurs.
- Définir et prioriser les fonctionnalités à développer dans le backlog produit.
- Diviser le développement en Sprints courts et réguliers d'environ deux semaines.
- Utiliser des cérémonies Scrum pour suivre l'avancement du projet et ajuster la planification en conséquence.
- Impliquer les parties prenantes tout au long du projet pour obtenir des feedbacks et des ajustements rapides.

**Les rôles Scrum dans le projet GAB** Les rôles Scrum dans le projet GAB sont les suivants :

**Le Product Owner** : qui est responsable de la définition du backlog produit et de la priorisation des fonctionnalités en fonction des besoins des utilisateurs.

**L'équipe de développement** : qui est responsable de la mise en œuvre des fonctionnalités du backlog produit et de l'ajout de tests pour garantir la qualité.

**Le Scrum Master** : qui est responsable de faciliter le processus Scrum, d'aider l'équipe à résoudre les problèmes et d'assurer le respect des règles Scrum.

**Les cérémonies Scrum dans notre projet GAB sont les suivantes**

**La réunion de planification de Sprint** : qui permet à l'équipe de développement de planifier le travail pour le prochain Sprint en fonction du backlog produit et des objectifs de l'itération.

**Les daily stand-up meetings** : qui permettent à l'équipe de développement de se synchroniser quotidiennement sur les tâches en cours et les éventuels problèmes rencontrés.

**La revue de Sprint** : qui permet à l'équipe de développement de présenter les fonctionnalités terminées lors du Sprint et de recueillir des feedbacks.

**La rétrospective de Sprint** : qui permet à l'équipe de développement d'évaluer le Sprint précédent et de proposer des améliorations pour les Sprints suivants

## 1-2-Services web

**L'architecture REST** (Representational State Transfer) est un style d'architecture logicielle qui permet de créer des services web simples, évolutifs et faciles à maintenir. Cette architecture repose sur un ensemble de contraintes qui permettent de garantir l'interopérabilité des systèmes, la performance et la scalabilité des services web.

Dans le cadre de notre projet GAB, l'utilisation de l'architecture REST permettra de fournir des services web accessibles à partir de n'importe quelle application ou dispositif connecté à Internet. L'architecture REST est basée sur les principes suivants :

- **Utilisation de ressources** : l'ensemble des services web est organisé en ressources identifiées par des URI (Uniform Resource Identifier).
- **Manipulation des ressources à travers les méthodes HTTP** : l'ensemble des opérations CRUD (Create, Read, Update, Delete) sont représentées par les méthodes HTTP (GET, POST, PUT, DELETE).
- **Passage d'informations entre client et serveur** : les données sont transférées entre client et serveur sous forme de messages en utilisant des formats de données standards tels que JSON ou XML.

- **Stateless** : chaque requête du client contient toutes les informations nécessaires pour que le serveur puisse comprendre la requête et y répondre, sans avoir besoin de stocker l'état de la session du client.

## 1-3 design patterns

Dans notre projet de distribution d'argent à l'aide d'une carte bancaire, nous avons appliqué plusieurs design patterns pour améliorer la qualité, la fiabilité et la maintenabilité du code.

- **MVC (Model-View-Controller)** : Nous avons utilisé le modèle MVC pour séparer la logique de présentation, la logique métier et les données de notre application. Le modèle représente les données et la logique métier, la vue représente la présentation et le contrôleur gère la communication entre la vue et le modèle.
- **Singleton** : Nous avons utilisé le pattern Singleton pour nous assurer qu'il n'y ait qu'une seule instance d'une classe donnée dans notre application. Nous avons utilisé cette approche pour garantir que certaines classes importantes ne soient pas instanciées plusieurs fois, ce qui pourrait causer des problèmes de cohérence.
- **Factory** : Nous avons utilisé le pattern Factory pour créer des objets sans avoir à connaître leur classe spécifique. Nous avons utilisé cette approche pour créer des objets de manière dynamique en fonction des besoins de l'application.
- **Observer** : Nous avons utilisé le pattern Observer pour permettre à des objets de surveiller les changements d'état d'autres objets. Nous avons utilisé cette approche pour permettre à certaines parties de notre application de s'abonner aux événements déclenchés par d'autres parties de l'application.
- **Strategy** : Nous avons utilisé le pattern Strategy pour permettre à notre application de choisir dynamiquement l'algorithme à utiliser pour résoudre un problème donné.  
Nous avons utilisé cette approche pour rendre notre application plus flexible et adaptable à différents contextes.

## 2-Etude conceptuelle

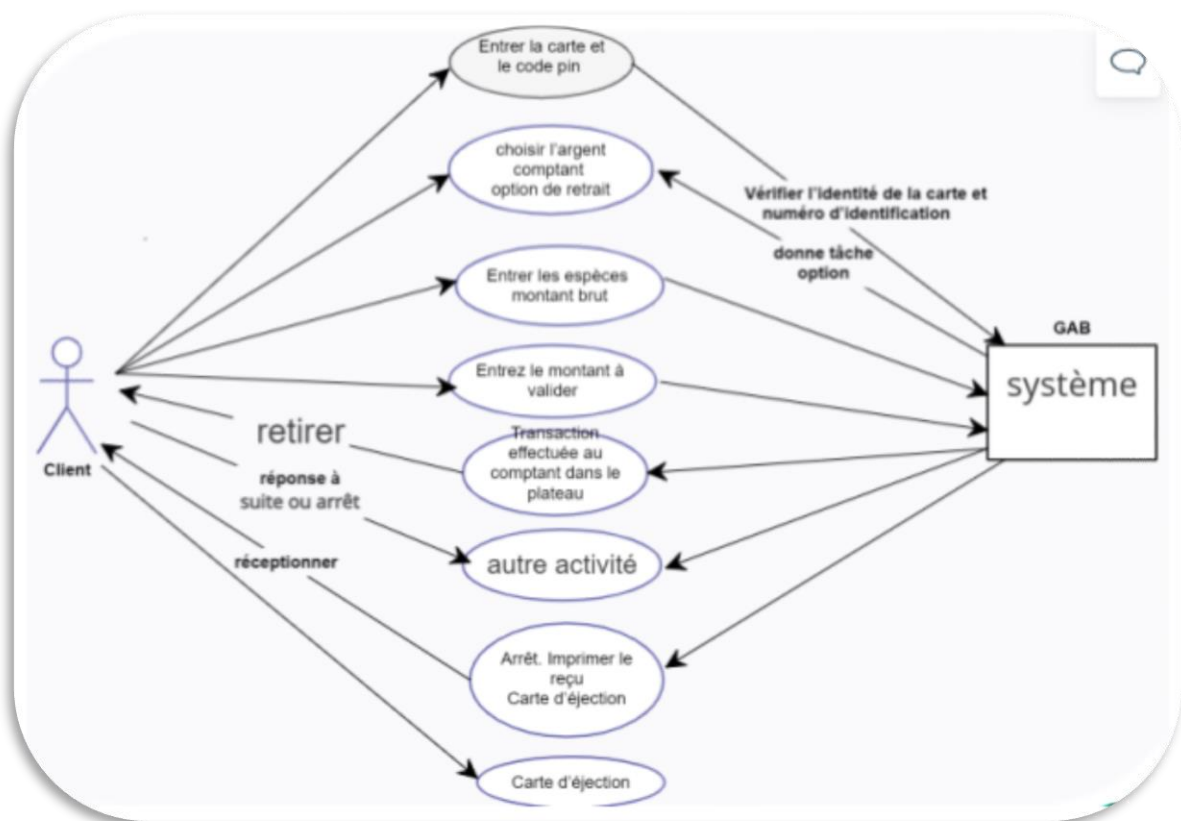
Le diagramme de cas d'utilisation est un outil important pour décrire les fonctionnalités d'un système et les utilisateurs qui interagissent avec ce système. Ce diagramme contient les éléments suivants :

- **Les acteurs :** les acteurs représentent les utilisateurs du système, qu'ils soient des utilisateurs humains ou des systèmes informatiques externes. Les acteurs sont représentés par des symboles spécifiques dans le diagramme et sont connectés aux cas d'utilisation correspondants.
- **Les cas d'utilisation :** les cas d'utilisation représentent les fonctionnalités ou les services fournis par le système aux utilisateurs. Chaque cas d'utilisation est représenté par un ovale dans le diagramme et est associé à un acteur ou à plusieurs acteurs.

Le diagramme de cas d'utilisation est un outil utile pour visualiser les interactions entre les utilisateurs et le système. Il peut être utilisé pour clarifier les exigences du système, pour définir les fonctionnalités à implémenter et pour faciliter la communication entre les différents membres de l'équipe de développement.

En utilisant ce diagramme, il est possible de mieux comprendre les besoins des utilisateurs et de concevoir un système qui répond à ces besoins de manière efficace et efficiente.

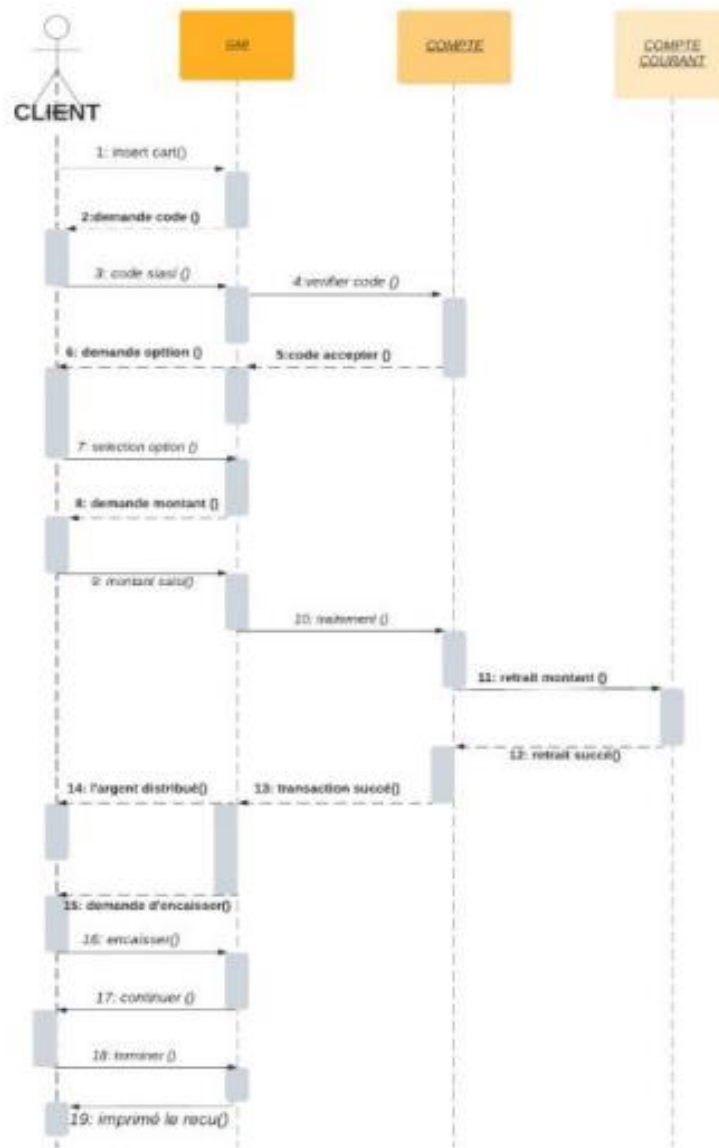




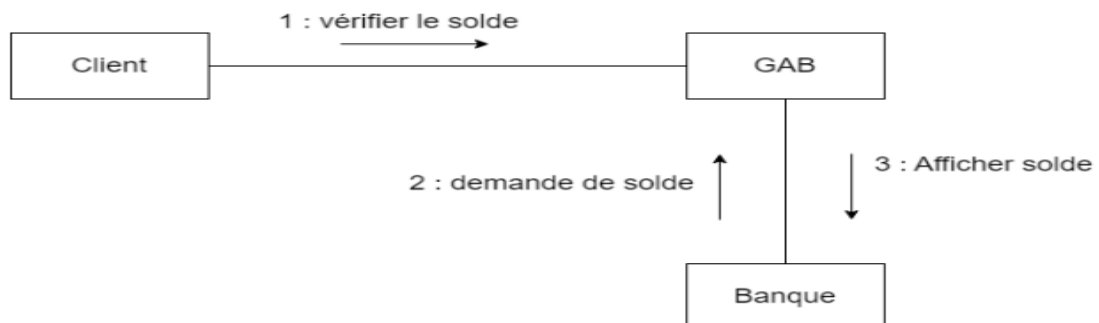
Les diagrammes de séquences représentent le flux des interactions entre les acteurs et les objets impliqués dans un cas d'utilisation. Ils permettent de visualiser la manière dont les fonctionnalités sont exécutées dans un système.

Un diagramme de séquence peut inclure :

- Les acteurs impliqués dans la fonctionnalité, c'est-à-dire les personnes ou les systèmes externes qui interagissent avec le système étudié.
- Les objets qui sont nécessaires au fonctionnement du système et qui interagissent entre eux pour fournir la fonctionnalité attendue.
- Les messages échangés entre les différents objets, qui représentent la communication nécessaire à la réalisation de la fonctionnalité.



Voici un exemple de diagramme de communication de vérifier le solde :



Voici un exemple de diagramme de communication de solde de dépôt :

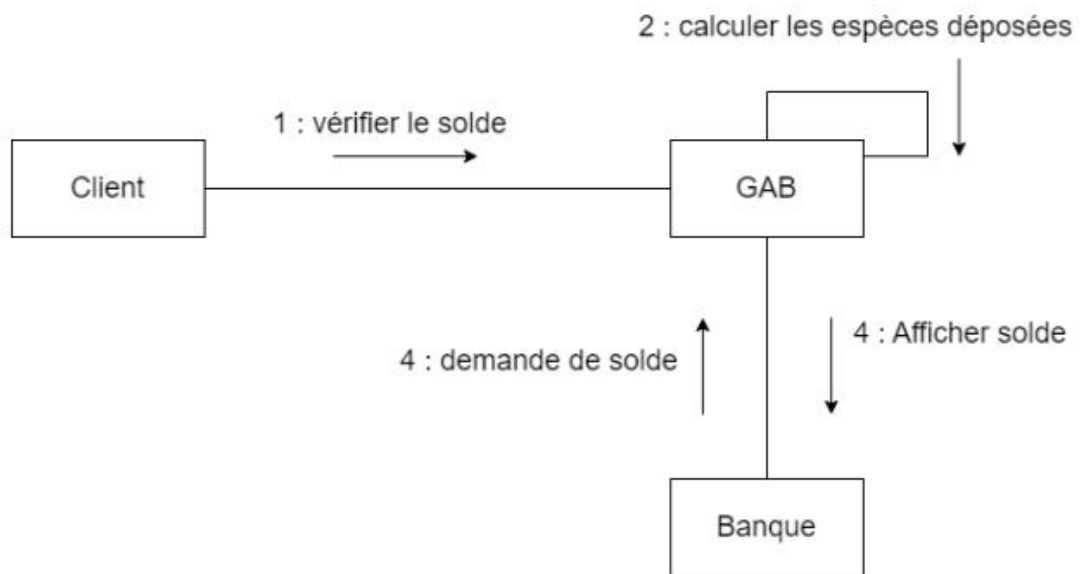
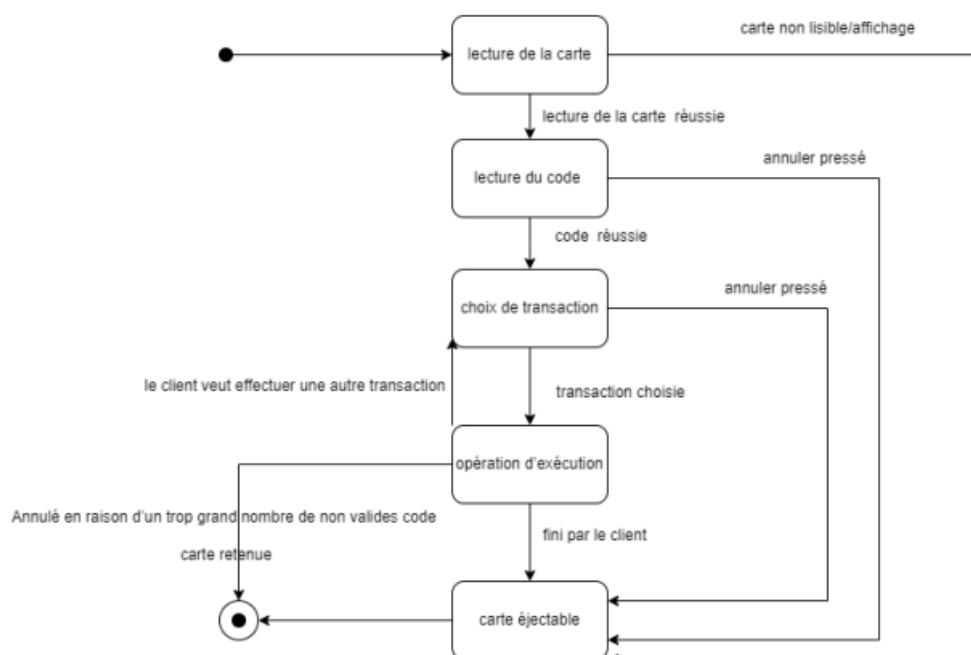
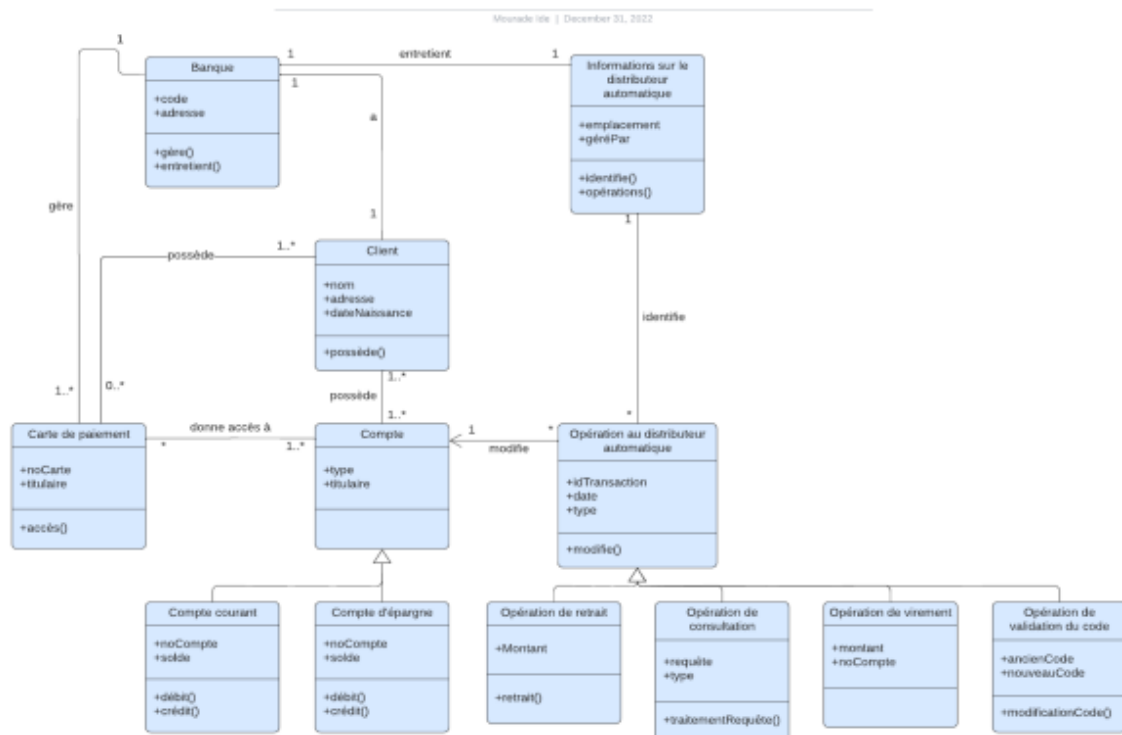


Diagramme d'état pour la session de GAB :



**Les diagrammes de classes** sont un outil de modélisation qui permettent de décrire la structure statique d'un système, c'est-à-dire comment il est organisé plutôt que comment il fonctionne.

Ces diagrammes représentent les classes, qui sont des entités ayant des caractéristiques communes. Les classes peuvent inclure des attributs (qui décrivent les propriétés de l'objet), des opérations (qui décrivent les actions que l'objet peut effectuer) et des associations (qui décrivent les relations entre les classes).



### 3- Environnement de travail

#### 3-1 Les Outils utilisés :

##### ➤ UML :

Uml le langage de modélisation unifié, c'est un langage graphique standard qui permet la spécification, la visualisation, la construction et la documentation des artefacts des systèmes logiciels. Il utilise principalement des notations graphiques pour exprimer l'analyse OO et

la conception de projets logiciels. Uml simplifie le processus complexe de conceptions de logiciels.



### ➤ **IntelliJ IDEA :**

IntelliJ IDEA développé par JetBrains est un environnement de développement intégré de technologie java utilisé pour le développement des logiciels informatiques. Il est disponible sous les éditions suivantes :

- ❖ Une édition communautaire gratuite et open-source qui fournit les fonctionnalités de base pour le développement JVM et Android.
  - ❖ Une édition commerciale (IntelliJ IDEA Ultimate), Elle contient des outils et fonctionnalités supplémentaires pour le développement web et Mobile..etc
- ✚ Durant notre travail on a utilisé une édition gratuite d'IntelliJ IDEA spécifique aux étudiants et professeurs.
  - ✚ Pour l'obtenir, vous devez visiter le site officiel du IntelliJ IDEA <https://www.jetbrains.com> et remplir le formulaire approprié pour cela, qui contient essentiellement des champs pour définir l'étudiant, sa carte étudiante, son email, le site de son établissement et autres ... La licence sera envoyée sur email en cas d'acceptation, elle prend environ une semaine (dans notre cas). Utilisable pour 1 ou 2 ordinateurs seulement.



➤ **Apache Tomcat :**

Tomcat est un serveur Web open source et un conteneur de servlets développé par apache software foundation.

Tomcat fournit un serveur Web open source et un conteneur de servlets et comprend des outils de configuration et de gestion.



➤ **Jwt :**

JSON Web Token (JWT) est un standard ouvert qui permet l'échange de jetons (tokens) entre plusieurs parties en toute sécurité. A travers l'algorithme HMAC ou RSA s'effectue l'échange sécurisé et le contrôle d'intégrité de données par une signature numérique.



➤ **Maven :**

- ❖ Maven est un outil de construction qui aide à la gestion de projet. Cet outil aide à construire et documenter le projet.
- ❖ Maven est écrit en java et C#, il est basé sur le modèle objet du projet POM, l'outil est utilisé pour créer et gérer n'importe quel projet basé sur Java. Il simplifie le travail quotidien des développeurs java et les aide dans leurs projets.



➤ **Laragon :**

Laragon est un logiciel gratuit, open source et léger utilisé pour configurer et gérer des environnements de développement locaux pour les applications web sur les systèmes d'exploitation Windows. Il est conçu pour simplifier le processus d'installation, de configuration et de basculement entre différentes versions de serveurs web, de bases de données et de langages de programmation tels que PHP, Apache, Nginx, MySQL, MariaDB, Node.js et Python. Laragon fournit également une interface conviviale pour gérer les hôtes virtuels, les certificats SSL et d'autres outils de développement web. Il est largement utilisé par les développeurs web pour créer et tester des applications web sur leurs machines locales avant de les déployer sur des serveurs de production.



### 3-2 Les frameworks utilisés :

➤ **Spring Boot :**

Spring boot est un Framework de développement applicative java open source qui permet de réaliser des micro-services, on l'utilise fréquemment pour la création d'api web.



## ➤ Angular

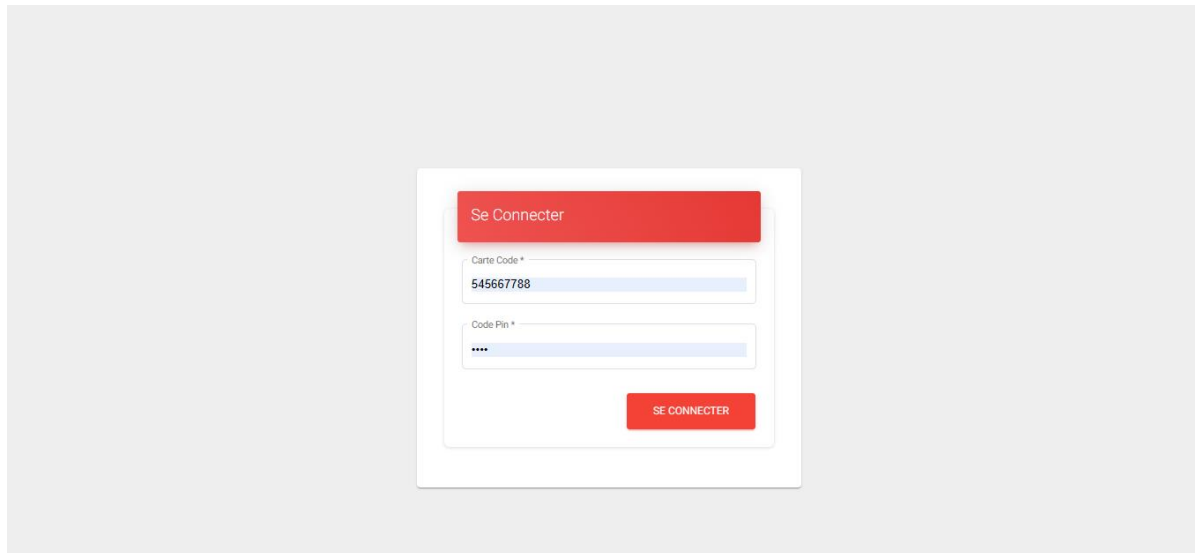
Angular est un framework de développement d'applications web open source créé par Google. Il permet aux développeurs de créer des applications web dynamiques et interactives en utilisant des langages tels que HTML, CSS et TypeScript. Angular utilise une architecture basée sur les composants pour faciliter la création de pages web réutilisables et modulaires. Il propose également des fonctionnalités telles que la liaison de données bidirectionnelle, les services, l'injection de dépendances, le routage et la gestion des formulaires. Angular est utilisé pour développer des applications web pour diverses plateformes, notamment les ordinateurs de bureau, les smartphones et les tablettes. Il est considéré comme l'un des frameworks les plus populaires pour le développement d'applications web côté client.





## 4 Présentation des interfaces du système :

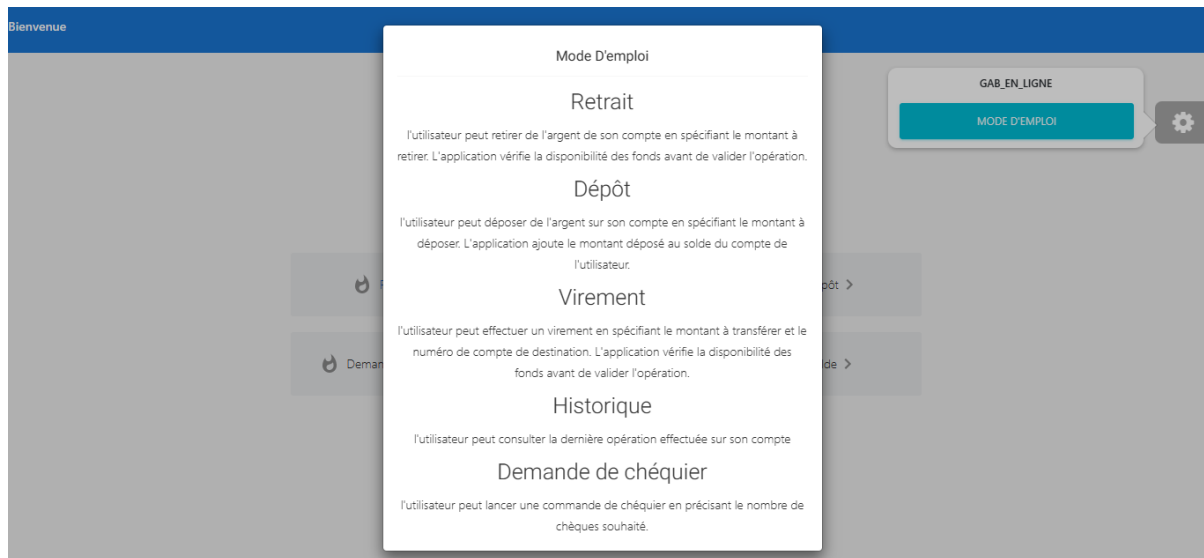
### ➤ La page d'authentification :



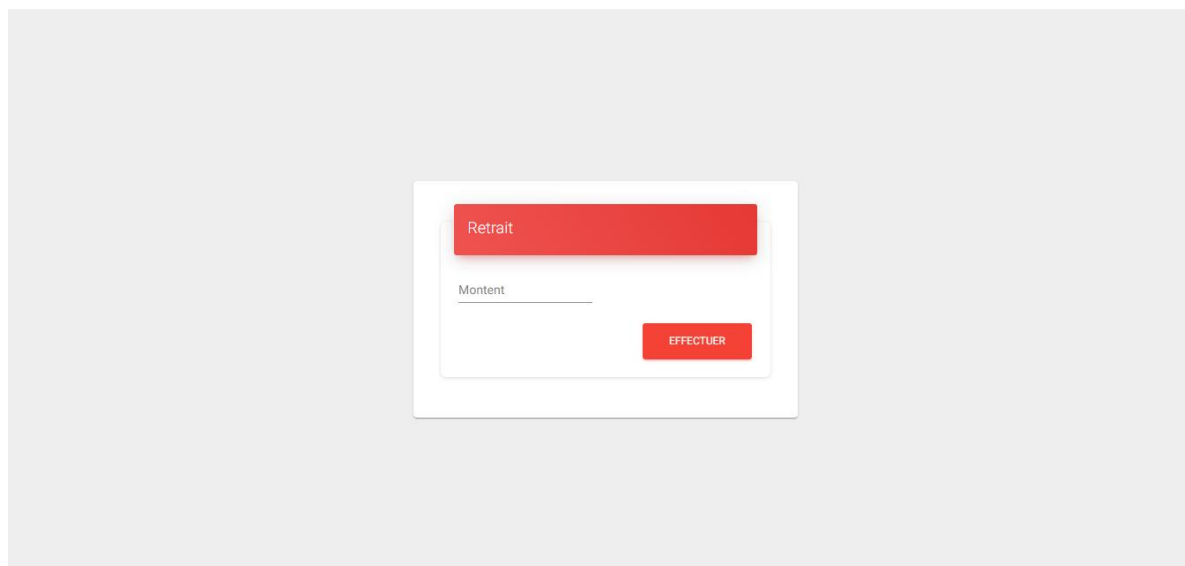
### ➤ Page home



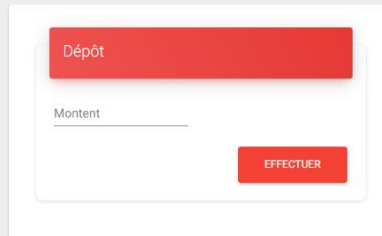
## ➤ Mode emploi



## ➤ Retrait

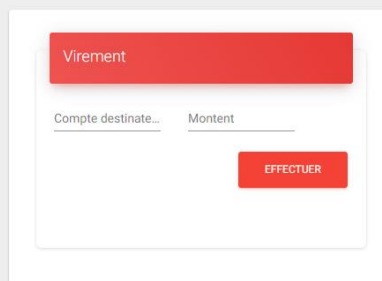


## ➤ Dépôt



A UI mockup for a deposit form. It features a red header bar with the text 'Dépôt'. Below the header is a text input field labeled 'Montant'. At the bottom right of the form is a red button labeled 'EFFECTUER'.

## ➤ Virement



A UI mockup for a transfer form. It features a red header bar with the text 'Virement'. Below the header are two text input fields: 'Compte destinate...' and 'Montant'. At the bottom right of the form is a red button labeled 'EFFECTUER'.

## 4-Conclusion

- ❖ Ce projet nous a aidé à :
- ✓ Utilisation des technologie web
- ✓ Expérience concrète de la réalité du métier
- ✓ Acquisition d'une méthode de travail en équipe

✓ **Manipulation des outils multiples**