



Gestion de Guichets Automatiques Bancaires (GAB)

Plan

1 Introduction

2 Cadre général du projet

3 Etude conceptuelle

4 Réalisation

5 Démonstration

6 Conclusion

Introduction

1



Introduction

Buts du projet:

- Le projet de Gestion de Guichets Automatiques Bancaires (GAB) consiste en la mise en place d'un système permettant la distribution d'argent à l'aide d'une carte bancaire valide. Les utilisateurs peuvent effectuer différentes opérations telles que le retrait, le dépôt, le virement,
- Ce projet vise à offrir un service de banque en libre-service pour les utilisateurs tout en garantissant la sécurité et la confidentialité de leurs informations personnelles et financières..

Cadre général du projet

2



Cadre général du projet

1 Les fonctionnalités principales de notre projet

2 Méthodes agiles

3 Services web

4 Design patterns

Les fonctionnalités principales de notre projet sont les suivantes :

- **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.

Les fonctionnalités principales de notre projet sont les suivantes :

- **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.

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**.

Méthode agiles

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.

Méthode agiles

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.

Méthode agiles

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.

Méthode agiles

Les cérémonies Scrum dans le projet GAB sont les suivantes

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.

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.

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 :

Services web

- **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.

Les Patterns

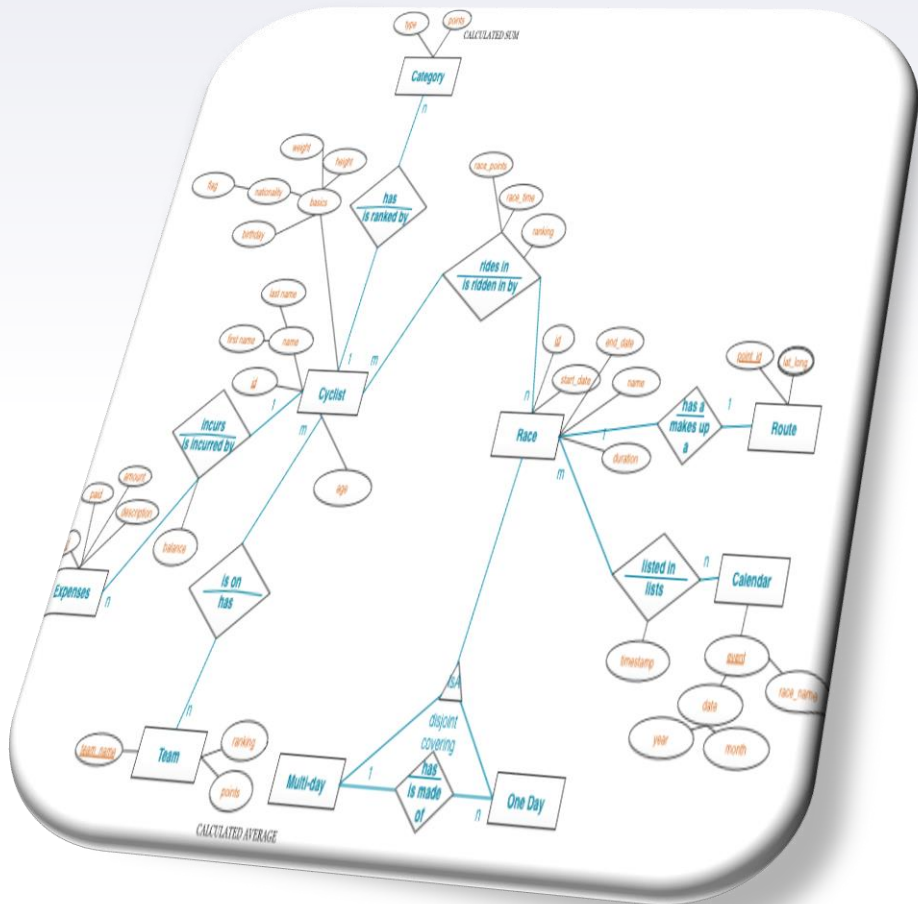
- **Dans notre projet de distribution d'argent à l'aide d'une carte bancaire, nous avons appliqué plusieurs design patterns pour améliorer la qualité, la fiabilité et la maintenabilité du code**

Les Patterns

1. **MVC (Model-View-Controller)**
2. **Singleton**
3. **Factory**
4. **Observer**
5. **Strategy**

Etude conceptuelle

3



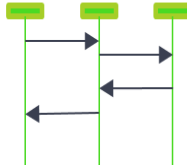
Etude conceptuelle



Cas d'utilisation



Diagramme de séquence



Modèle physique
de données



Diagramme de cas d'utilisation

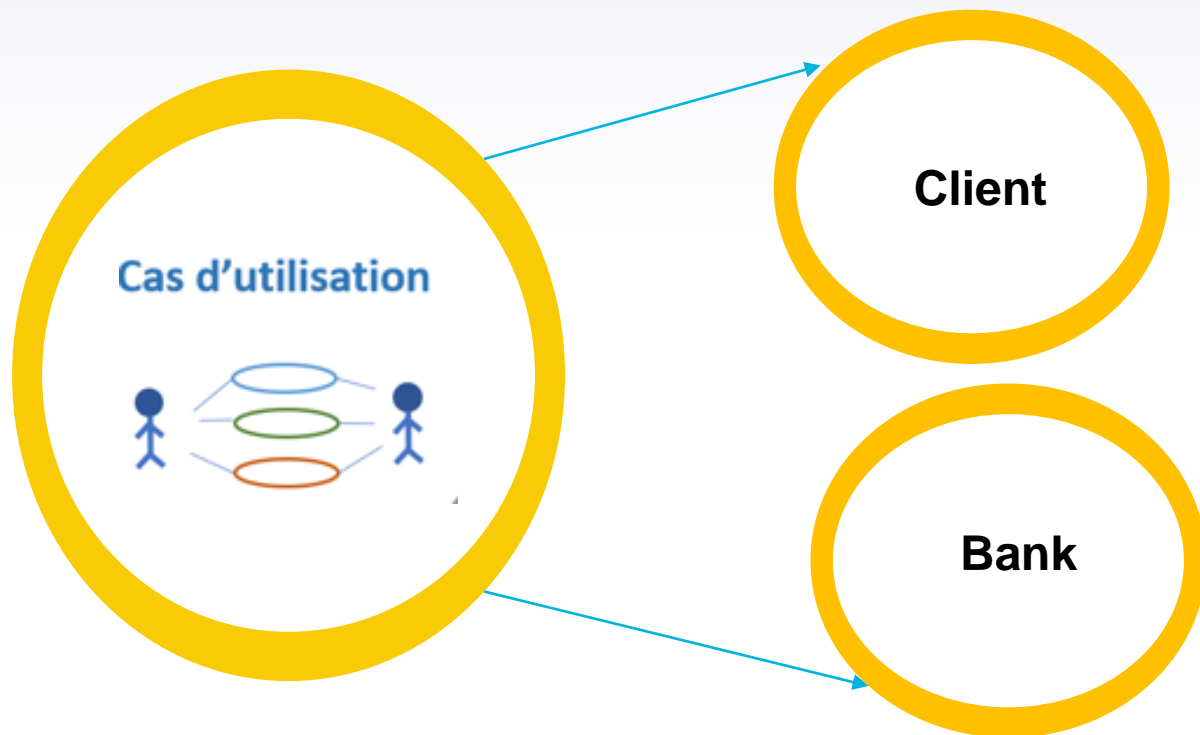


Diagramme de cas d'utilisation

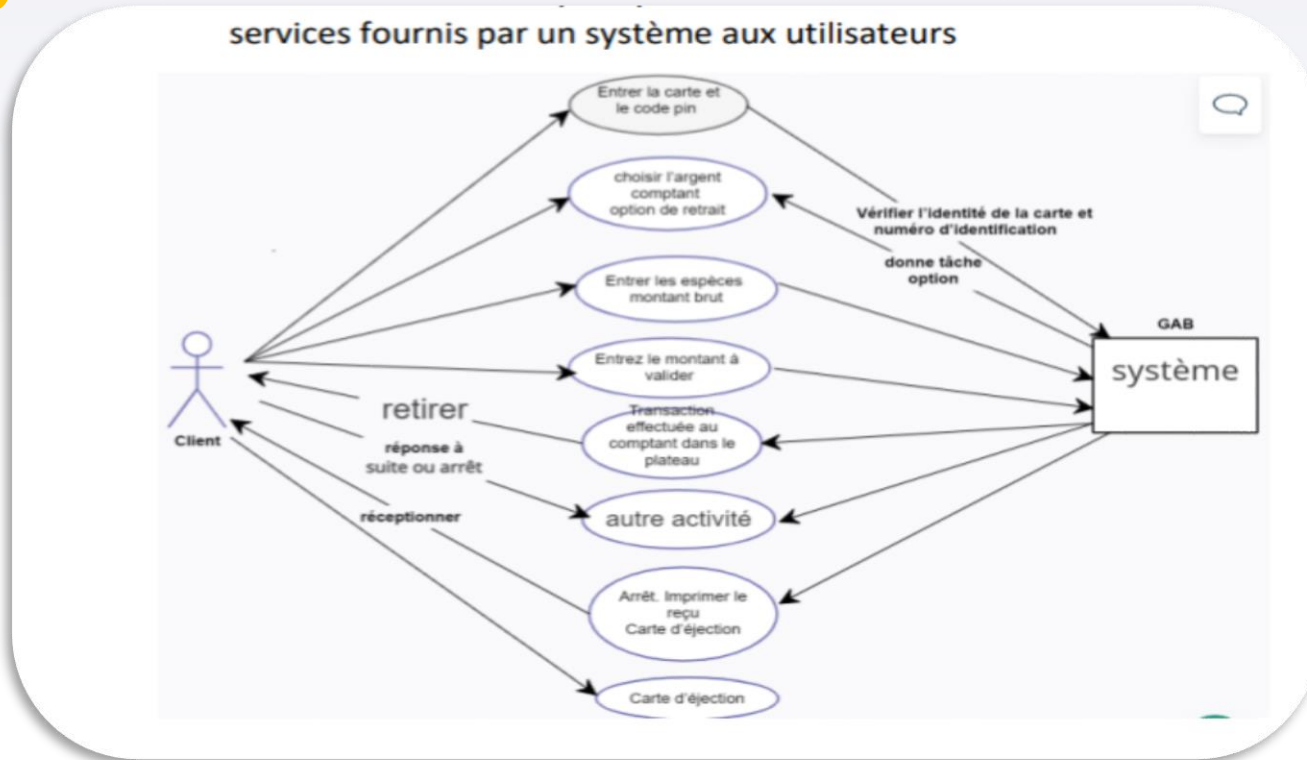
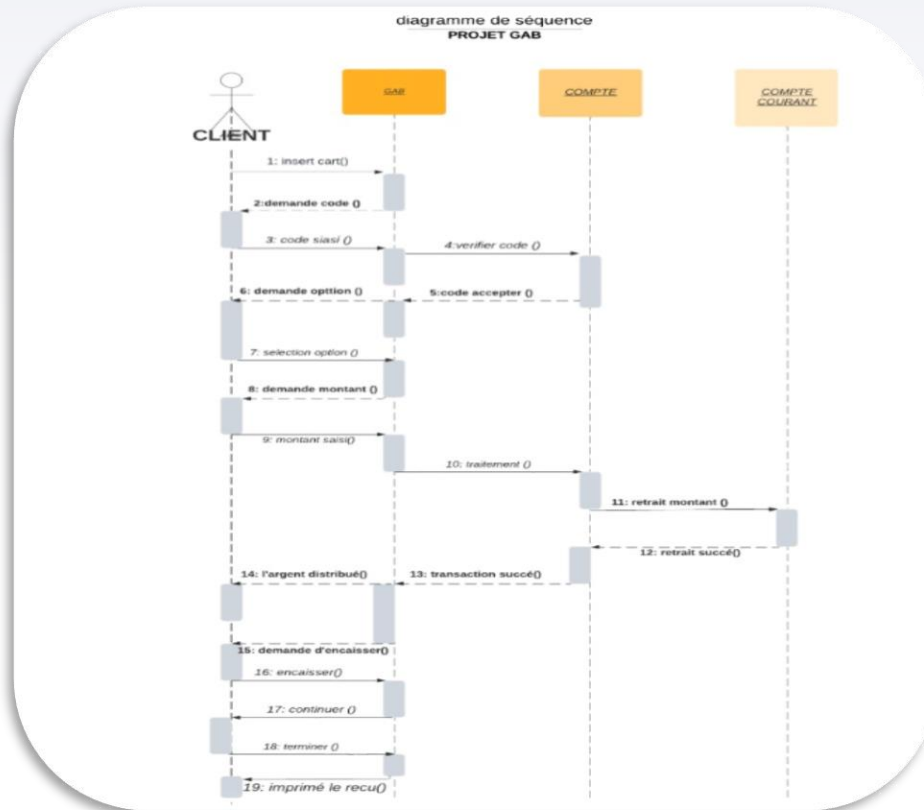
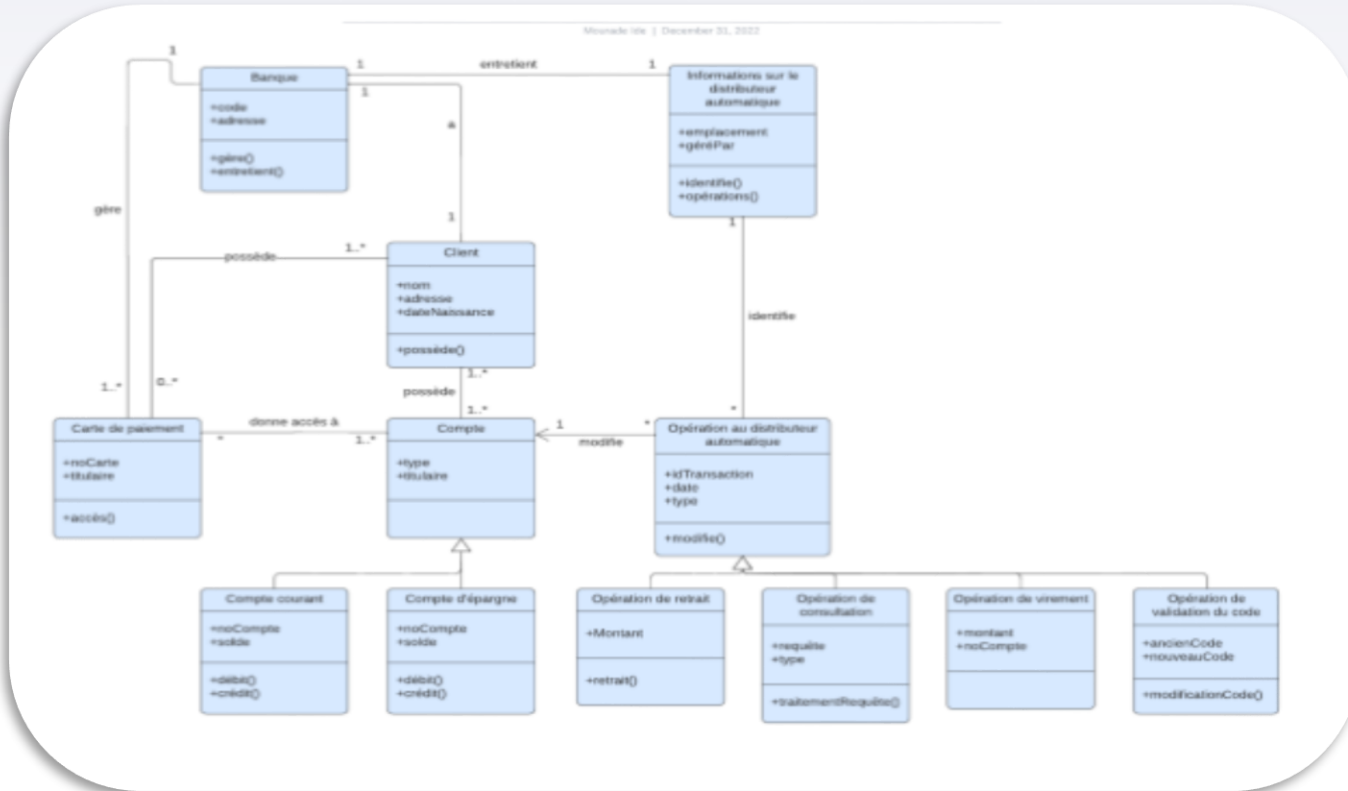


Diagramme de cas d'utilisation « Bank et Client »

Diagramme de séquence



Modèle physique de données



► Réalisation

4



Réalisation

Les outils



Les Frameworks



L'architecture
MVC



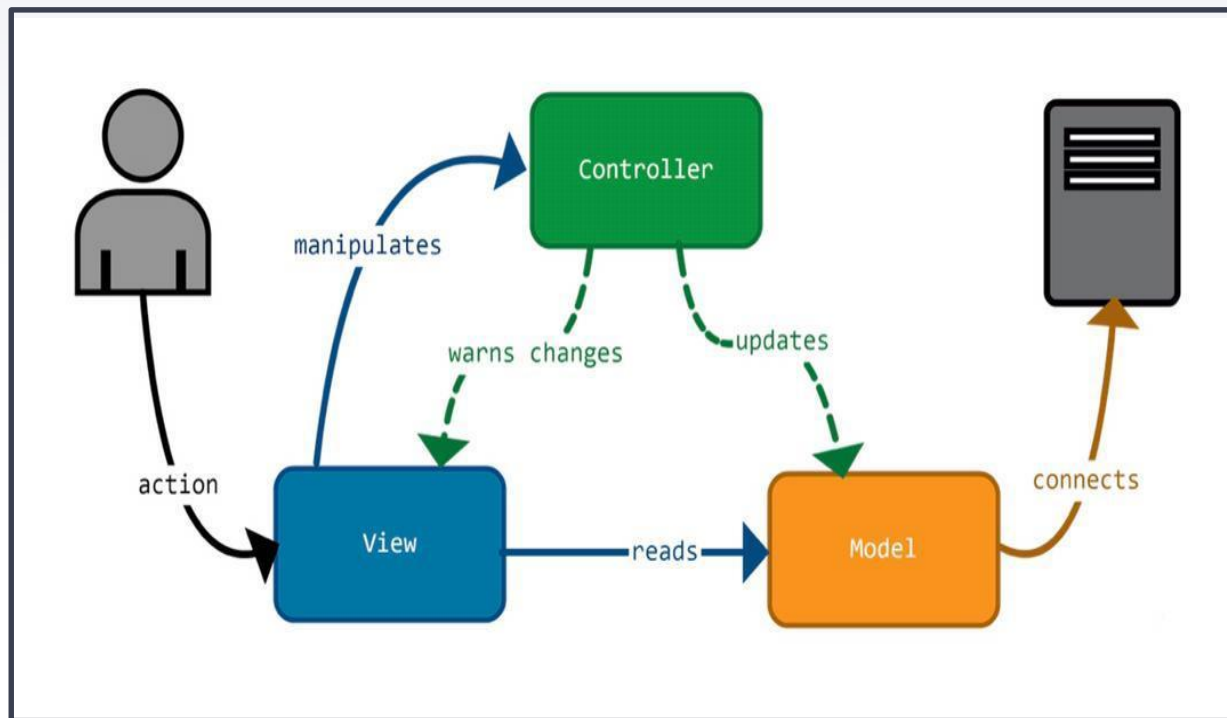
Les outils utilisés



Les outils



L' Architecture MVC



Les frameworks utilisés



► **Spring Boot**



► **Angular**

► Démonstration



Solution proposée



Conclusion

6



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

❖ Perspectives :

- ✓ Ameliorer les fonctions
- ✓ Verision Mobile Avec Flutter

MERCI!

