

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
ECOLE SUPERIEURE EN INFORMATIQUE 8 MAI 1945, SIDI BEL-ABBES  
1ERE ANNEE SECOND CYCLE

# Rapport d'architecture

---

une application pour la gestion des œuvres  
sociales de l'ESI-SBA



**2023-2024**

# Table de version

---

Version	Date de sortie	Auteur	Changement
1.0	05-04-2024	Douis Fatima Zohra	Premiere Version

# Table des matieres

---

## **1-Generalite sur rapport d'architecture**

1-1 Definition de rapport d'architecture

1-2 But de rapport d'architecture

## **2-Architecture globale**

2-1 Les acteurs de système.

2-2 Le shema globale.

## **3-Shema detaille de chaque module**

3-1 Gestion des comptes et des roles.

3-2 Gestion des demandes.

3-3 Gestion des annonces.

3-4 Gestion des fonds.

## **4- Architecture utilisee.**

## **5- Diagramme de deploiement.**

## **6- Diagramme de composants.**

## **7- Conclusion.**

# **Chapitre 1**

Generalite sur rapport d'architecture

## **1. Definition de rapport d'architecture**

Un rapport d'architecture est un document qui décrit la structure globale d'un système informatique. Il fournit une vue d'ensemble des composants du système, les fonctionnalités, les interconnexions, les interfaces, les contraintes et les performances. Il est important pour les équipes de développement et d'ingénierie, car il fournit un guide pour la conception, la mise en œuvre et la maintenance du système ou du logiciel.

## **2. But de rapport d'architecture :**

La modélisation de l'architecture système permet de :

- Comprendre les besoins : Examiner ce que veulent les utilisateurs et les parties prenantes pour savoir quoi faire.
- Concevoir le système : Planifier comment les différentes parties du système vont travailler ensemble pour satisfaire les besoins identifiés.
- Communiquer les idées : Partager les plans avec les membres de l'équipe et ceux qui sont concernés, en utilisant des images et des mots faciles à comprendre.

- Évaluer les choix de conception : Tester différentes idées pour voir ce qui marche bien et ce qui ne marche pas aussi bien.

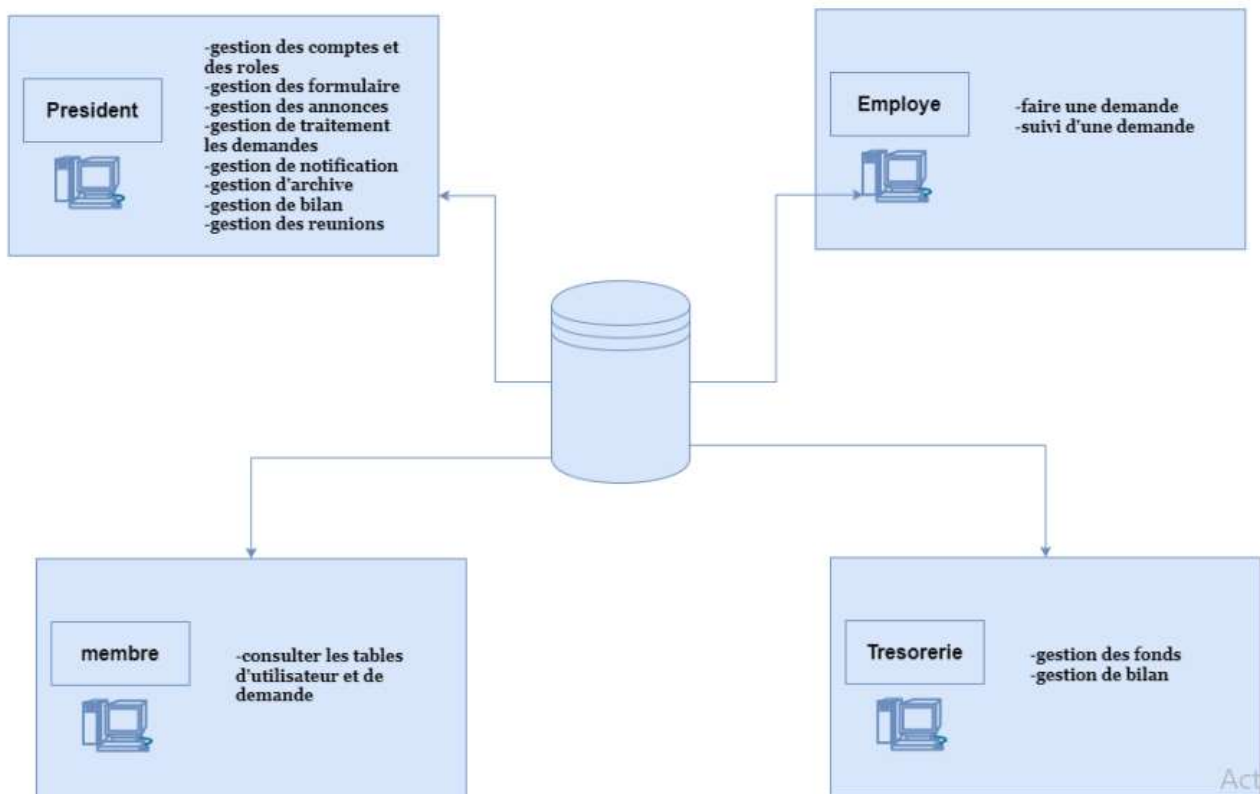
# **Chapitre 2**

Architecture globale

## 1. Les acteurs de système :



## 2. Le shema globale :

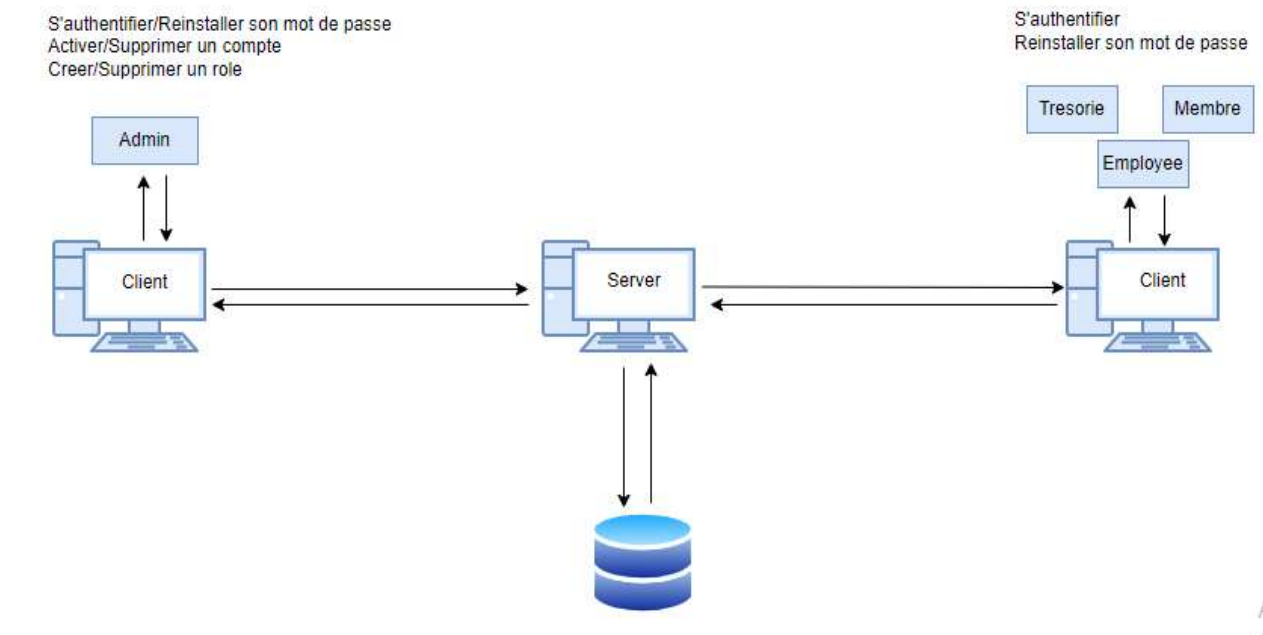




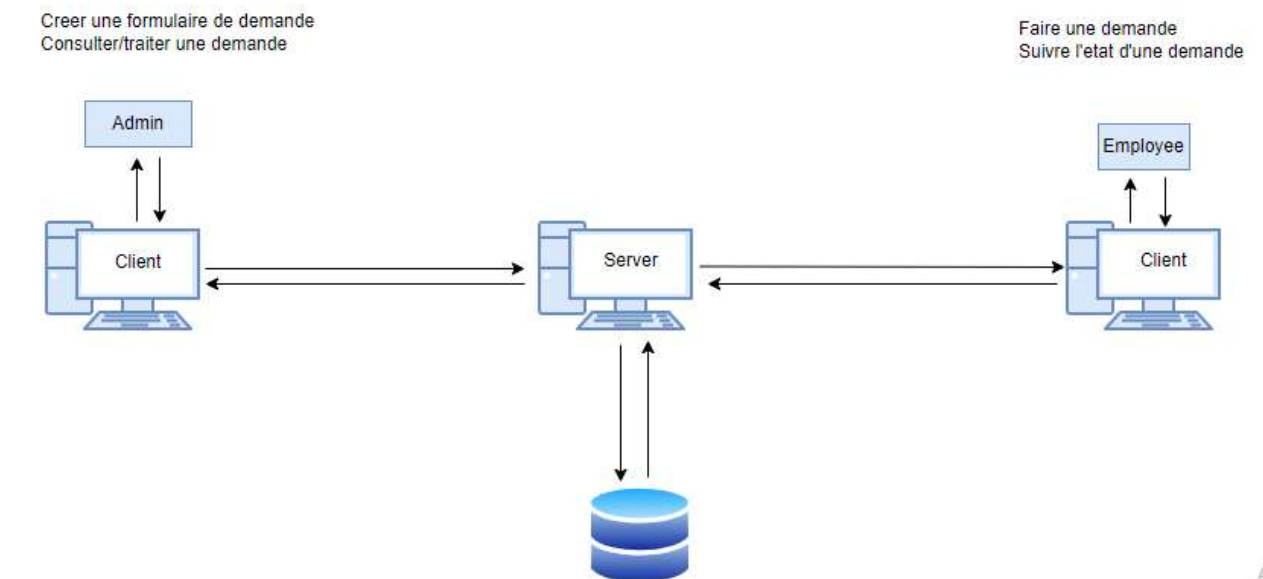
## **Chapitre 3**

Shema detaille de chaque module

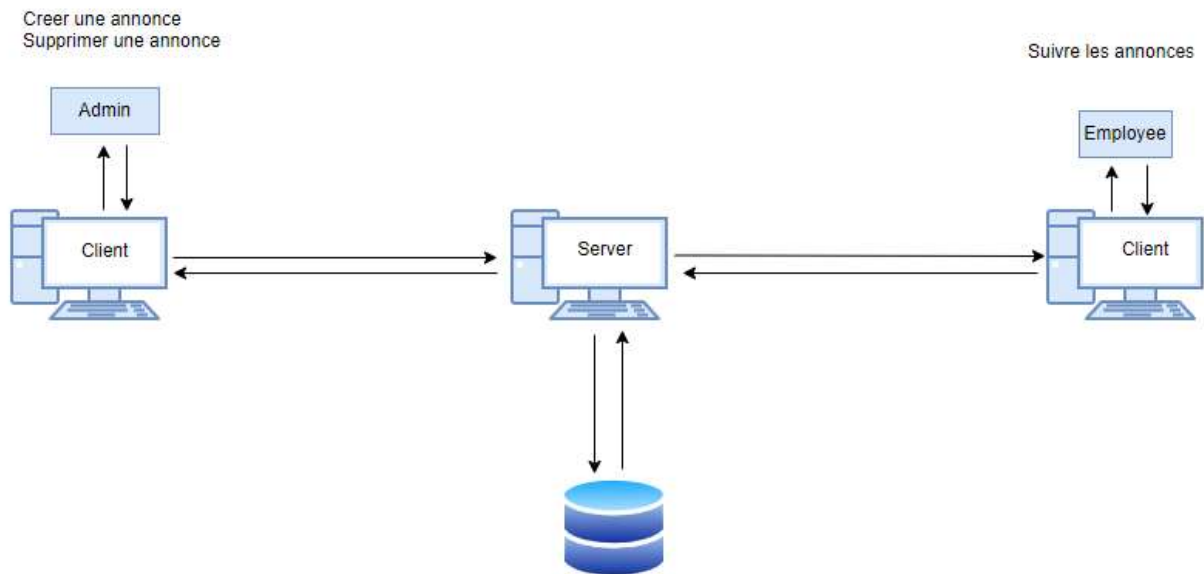
# 1. Gestion des comptes et des roles :



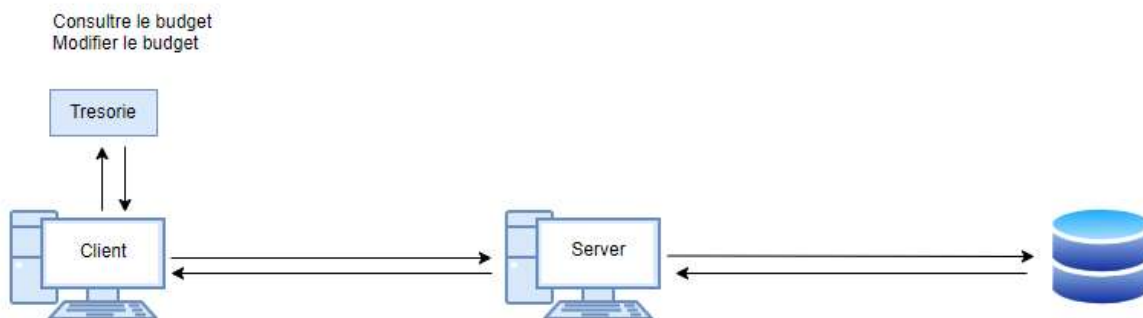
# 2. Gestion des demandes :



### 3. Gestion des annonces :



### 4. Gestion des fonds :



# **Chapitre 4**

Architecture utilise

Dans cette application le modele MVC est utilisee.

L'architecture Modele/View/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste a distinguer trois entites distinctes qui sont, le modele, la vue et le controleur ayant chacun un role prcis dans l'interface.

**Modele** : donnees (acces et mise a jour).

**Vue** : interface utilisateur (entrées et sorties).

**Contrôleur** : gestion des evenements et synchronisation.

L'objectif du MVC est de bien séparer les données, les traitements et la présentation de ces données.

La règle est « diviser pour mieux régner ».

Chaque couche est spécialisée, donc est dédiée aux tâches qu'elles savent faire le mieux.

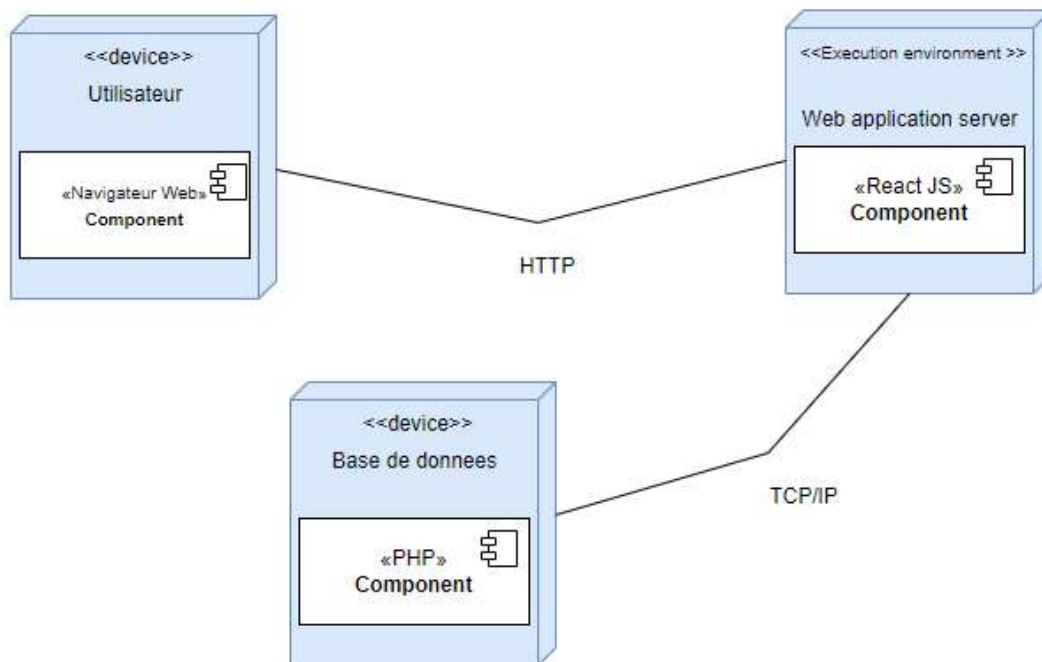
# **Chapitre 5**

Diagramme de déploiement

Un diagramme de déploiement est un type de diagramme UML qui montre l'architecture d'exécution d'un système, y compris les nœuds tels que les environnements d'exécution matériels ou logiciels, et l'intergiciel qui les relie.

Les diagrammes de déploiement sont généralement utilisés pour visualiser le matériel physique et les logiciels d'un système. En l'utilisant, vous pouvez comprendre comment le système sera physiquement déployé sur le matériel.

Les diagrammes de déploiement aident à modéliser la topologie matérielle d'un système par rapport à d'autres types de diagrammes UML qui décrivent principalement les composants logiques d'un système.

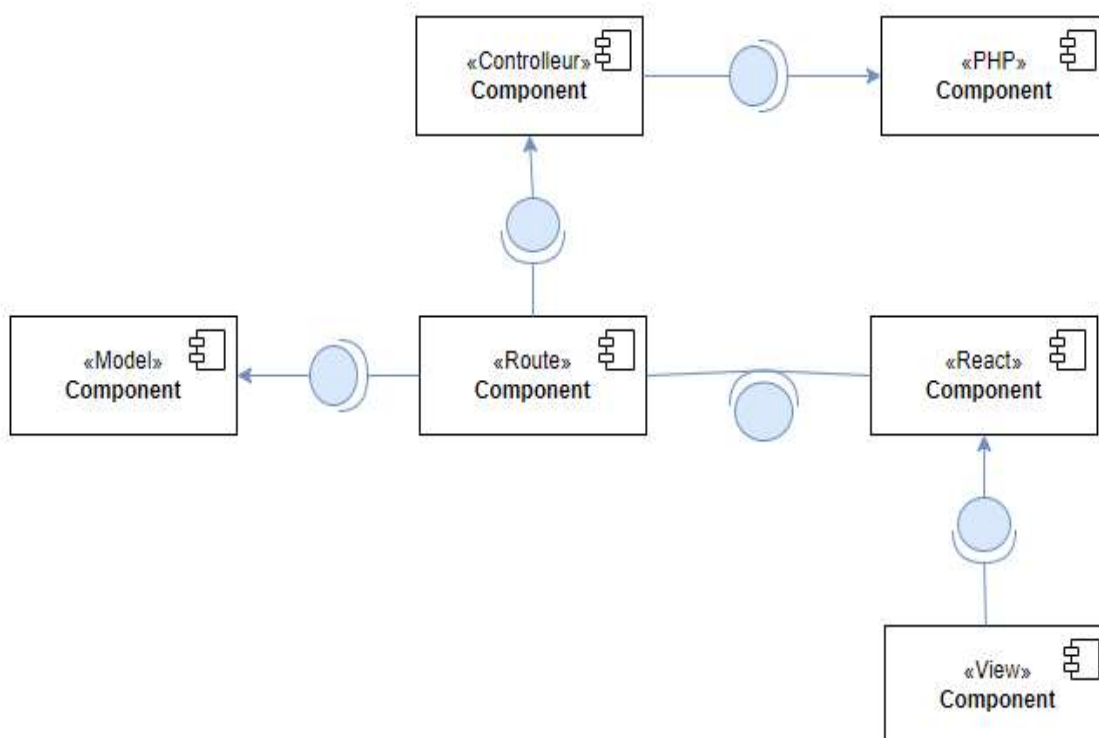


# **Chapitre 6**

Diagramme de composants



Un diagramme de composant a pour objectif d'illustrer la relation entre les différents composants d'un système. Dans le cadre de l'UML 2.0, le terme « composant » fait référence à un module de classes qui représentent des systèmes ou des sous-systèmes indépendants ayant la capacité de s'interfacer avec le reste du système.



# Conclusion

L'architecture MVC offre un cadre normalise pour le developpement, une bonne pratique devenue une philosophie pour bien structurer une application, séparer les problematiques, optimiser la coordination et la communication entre les differents corps de métier (concepteur, developpeur, DBA, integrateur...).

Ainsi, grace au design pattern Model-View-Controller, les developpeurs ne se soucient plus de la technique. Ils ne se concentrent que sur les aspects fonctionnels du metier, du developpement des composants metiers et de la conception de la base de donnees.

Le MVC ne pretend pas etre la solution à tous les problemes, mais montre une voie sur la façon dont ces problemes peuvent etre traites rapidement et avec efficacite.