

**Projet de Fin d’Etudes**

**Année Universitaire 2023-2024**

Conception et développement d’**une** **application de gestion des assemblées générales** de la CIMR



**Projet de Fin d’Etudes**

Conception et développement d’**une** **application de gestion des assemblées générales** de la CIMR

***Réalisé par :*** Mohamed IBN KHAYAT

***Membres du jury:***

Mr. BELASLA Mehdi

Mr. BENDRISS El Mehdi

***Président***

Encadrant Interne

# Dédicaces

Je dédie ce modeste travail particulièrement à mes chers parents, qui ont consacré leur existence à bâtir la mienne, pour leur soutien, patience et soucis de tendresse et d’affection, pour tout ce qu’ils ont fait pour que je puisse arriver à ce stade.

A ma mère, qui m’a encouragé et qui sans elle, ma réussite n’aura pas eu lieu. Qu’elle trouve ici mon amour et mon affection.

A mon père, qui est toujours disponible pour nous, et prêt à nous aider, je lui confirme mon attachement et mon profond respect.

A la mémoire de ma grand-mère qui nous a quittés voilà 6 ans.

A toute ma famille, amis de ma promotion et à tous ceux que j’ai connu durant mon cycle d’études.

# Remerciements

Au terme de ce travail, je saisis cette occasion pour exprimer mes remerciements les plus sincères à toute personne ayant contribué, de près ou de loin, à la réalisation de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Je tiens à remercier Mr. Mohammed ERRAIS, qui a accepté de m’encadrer et qui s'est toujours montré à l'écoute tout au long de la réalisation de ce projet, pour ses conseils qui m’ont été bien utiles, et pour la patience dont il a su faire preuve malgré ses charges professionnelles.

Je souhaite exprimer ma plus profonde gratitude à Mr. Youssef BAZ, DSI de la CIMR, pour son soutien indéfectible et la confiance qu'il m'a accordée tout au long de ce projet. Mes remerciements vont également à mon encadrant externe, Mr. Abdallah Qaouch, pour ses orientations pertinentes et son encadrement précieux qui ont été d'une aide inestimable. J'adresse également mes sincères remerciements à l'ensemble du service informatique de la CIMR pour leur aide et leur collaboration tout au long de cette expérience enrichissante.

J’exprime également ma gratitude aux membres du jury qui nous ont honorés par leur présence et leur acceptation d'évaluer ce travail, à tous mes enseignants pour leur dévouement et leur assistance tout au long de cette année, sans oublier les membres du conseil d’administration ; Mr. El Mehdi BENDRISS, Mr. Karim RACHDI et Mr. Driss BENOMAR qui font de leur mieux pour une meilleure gestion administrative de l’Institut Supérieur Vinci.

Je n'oublie surtout pas mes parents, mes frères et sœurs pour leurs contributions, leurs soutiens et leur patience.

Enfin, j’adresse mes plus chaleureux remerciements à tous mes proches et amis, qui m’ont toujours soutenus et encouragés au cours de la réalisation de ce projet.

Merci à toutes et à tous !

# Résumé

Ce document présente le travail réalisé dans le cadre de mon projet de fin d'études, axé sur la gestion des assemblées générales pour la CIMR. L'objectif principal de ce projet est d'automatiser et de moderniser les processus complexes associés à l'organisation des assemblées générales, moments clés pour la gouvernance de la CIMR. Actuellement, ces processus sont souvent gérés de manière manuelle, ce qui peut entraîner des inefficacités et des erreurs. Le projet a été développé sous la forme d'une plateforme web, qui centralise les différentes étapes de la préparation des assemblées générales, depuis la création de l'assemblée jusqu'à l'extraction des listes d'adhérents éligibles. Chaque assemblée est associée à un ensemble de documents, tels que la feuille de route, les convocations, et les feuilles de présence, qui sont gérés de manière automatisée. Une des particularités de ce projet est l'automatisation de la génération de la feuille de route, qui se base sur les données saisies lors de la création de l'assemblée, ainsi que l'extraction automatique des listes d'adhérents en fonction de critères spécifiques. Le projet a été mené en suivant une méthodologie agile, en mettant l'accent sur l'efficacité opérationnelle, la sécurité des données, et la conformité aux réglementations en vigueur. L'ensemble des étapes, depuis l'analyse des besoins jusqu'au développement de la solution, a été soigneusement documenté pour assurer la conformité aux attentes de la CIMR. Les technologies utilisées pour ce projet incluent Spring Boot pour le backend, Angular pour le frontend, et SQL Server pour la gestion des données, avec un focus particulier sur l'intégration de processus automatisés et la gestion des utilisateurs.

.

.

# Abstract

This document outlines the work completed as part of my final year project, focused on the management of general assemblies for the CIMR. The primary objective of this project is to automate and modernize the complex processes associated with organizing general assemblies, which are critical for CIMR's governance. Currently, these processes are often managed manually, leading to inefficiencies and errors. The project was developed as a web platform that centralizes the various stages of preparing general assemblies, from the creation of the assembly to the extraction of eligible member lists. Each assembly is associated with a set of documents, such as the roadmap, invitations, and attendance sheets, which are managed automatically. A key feature of this project is the automation of the roadmap generation, based on the data entered during the assembly creation, and the automatic extraction of member lists based on specific criteria. The project was conducted using an agile methodology, with a focus on operational efficiency, data security, and compliance with current regulations. All stages, from the needs analysis to the development of the solution, were carefully documented to ensure alignment with CIMR's expectations. The technologies used for this project include Spring Boot for the backend, Angular for the frontend, and SQL Server for data management, with particular attention to integrating automated processes and user management.

# Tables des matières

[Dédicaces 4](#_Toc175922509)

[Remerciements 5](#_Toc175922510)

[Résumé 7](#_Toc175922511)

[Abstract 8](#_Toc175922512)

[Tables des matières 9](#_Toc175922513)

[Introduction générale : 12](#_Toc175922514)

[Chapitre 1 : Contexte général 13](#_Toc175922515)

[Introduction 13](#_Toc175922516)

[1. Présentation de l’organisme d’accueil : 14](#_Toc175922517)

[1.1. Présentation de la CIMR : 14](#_Toc175922518)

[1.2. Mission : 15](#_Toc175922519)

[1.3. Valeurs : 15](#_Toc175922520)

[1.4. Métiers 15](#_Toc175922521)

[1.5. Choix stratégiques 16](#_Toc175922522)

[1.6. Fiche technique 16](#_Toc175922523)

[1.7. Organigramme 16](#_Toc175922524)

[1.8. Engagements de service 17](#_Toc175922525)

[1.9. Produits et options de la CIMR 18](#_Toc175922526)

[1.9.1. Les produits 18](#_Toc175922527)

[1.9.2. Les options : 19](#_Toc175922528)

[Conclusion 20](#_Toc175922529)

[Chapitre 2 : Etude du projet 21](#_Toc175922530)

[Introduction 21](#_Toc175922531)

[1. Périmètre du projet 22](#_Toc175922532)

[1.1. Problématique générale 22](#_Toc175922533)

[1.2. Solution : 22](#_Toc175922534)

[1.3. But du projet : 23](#_Toc175922535)

[1.4. Missions du projet : 23](#_Toc175922536)

[1.5. Livrables 24](#_Toc175922537)

[1.6. Risques du projet : 24](#_Toc175922538)

[2. Conduite du projet : 25](#_Toc175922539)

[2.1. Méthodologie de développement 25](#_Toc175922540)

[2.2. Pourquoi SCRUM ? 26](#_Toc175922541)

[2.3. Cycle de vie : 26](#_Toc175922542)

[2.3.1. Phases du cycle retenu 26](#_Toc175922543)

[2.3.2. Description des phases : 27](#_Toc175922544)

[2.4. Planning du projet : 28](#_Toc175922545)

[3. Organisation du projet : 28](#_Toc175922546)

[3.1. Rôles des acteurs du projet 28](#_Toc175922547)

[3.2. Suivi du projet : 29](#_Toc175922548)

[3.2.1. Moyens de communication : 30](#_Toc175922549)

[Conclusion 30](#_Toc175922550)

[Chapitre 3 : Analyse du besoin 31](#_Toc175922551)

[Introduction 31](#_Toc175922552)

[1. Cahier des charges : 32](#_Toc175922553)

[1.1. Problématique 32](#_Toc175922554)

[1.2. Etude de l’existant 32](#_Toc175922555)

[1.2.1. Concept métier : 32](#_Toc175922556)

[1.3. Etude fonctionnelle : 33](#_Toc175922557)

[1.3.1. Besoins fonctionnels : 33](#_Toc175922558)

[Conclusion 37](#_Toc175922559)

[Chapitre 4 : Etude conceptuelle 38](#_Toc175922560)

[Introduction 38](#_Toc175922561)

[1. Langage UML 39](#_Toc175922562)

[1.1. Présentation : 39](#_Toc175922563)

[1.2. UML et POO 39](#_Toc175922564)

[1.3. Les avantages d’UML 39](#_Toc175922565)

[1.4. Utilité de l’UML selon l’OMG : 40](#_Toc175922566)

[1.5. Identification des acteurs : 40](#_Toc175922567)

[1.5.1. Les acteurs de l’application 41](#_Toc175922568)

[1.5.2. Rôles des acteurs : 41](#_Toc175922569)

[1.6. Diagramme de paquetages : 41](#_Toc175922570)

[1.7. Diagramme des cas d’utilisation : 42](#_Toc175922571)

[1.8. Diagramme de séquence : 42](#_Toc175922572)

[1.9. Diagramme des classes : 42](#_Toc175922573)

[Conclusion 43](#_Toc175922574)

[Chapitre 5 : Etude technique 44](#_Toc175922575)

[Introduction 44](#_Toc175922576)

[1. Outils de travail : 45](#_Toc175922577)

[1.1. Logiciels : 45](#_Toc175922578)

[1.1.1. IntelliJ IDEA Ultimate 45](#_Toc175922579)

[1.1.2. Visual Studio Code 45](#_Toc175922580)

[1.1.3. Enterprise Architect 45](#_Toc175922581)

[1.1.4. HeidiSQL 45](#_Toc175922582)

[1.1.5. Insomnia 46](#_Toc175922583)

[1.1.6. Brave 46](#_Toc175922584)

[1.2. Langages et technologies de développement 46](#_Toc175922585)

[1.2.1. Java 17 46](#_Toc175922586)

[1.2.2. TypeScript 46](#_Toc175922587)

[1.3. Framework et bibliothèques utilisés 47](#_Toc175922588)

[1.3.1. Spring Framework 47](#_Toc175922589)

[1.3.2. Angular 47](#_Toc175922590)

[1.3.3. Apache POI 47](#_Toc175922591)

[1.3.4. Tailwind CSS 48](#_Toc175922592)

[1.3.5. Flowbite 48](#_Toc175922593)

[1.4. Base de données : 48](#_Toc175922594)

[1.4.1. Microsoft SQL Server 48](#_Toc175922595)

[2. Architecture de l’application 49](#_Toc175922596)

[2.1. Architecture BO 49](#_Toc175922597)

[2.2. Architecture FO 50](#_Toc175922598)

[3. Structure de l’application 51](#_Toc175922599)

[3.1. Arborescence du BO 51](#_Toc175922600)

[3.2. Arborescence du FO 52](#_Toc175922601)

[Chapitre 6 : Mise en œuvre de l’application 54](#_Toc175922602)

[Introduction 54](#_Toc175922603)

[Liste des abréviations 55](#_Toc175922604)

# Introduction générale :

Dans ce manuscrit, je présente mon projet de fin d'étude réalisé pour la Caisse Interprofessionnelle Marocaine de Retraite (CIMR). Le thème de mon projet, intitulé « Gestion des Assemblées Générales », se concentre sur le développement d'un système automatisé pour la gestion des assemblées générales, un processus crucial pour le fonctionnement efficace de la CIMR. La gestion des assemblées générales inclut la création des assemblées, l'extraction et la gestion des listes de membres, et la préparation des documents nécessaires.

L'idée de ce projet est née de la nécessité d'améliorer l'efficacité et la précision de la gestion des assemblées générales. En automatisant ce processus, nous visons à réduire les erreurs humaines, à gagner du temps et à assurer une meilleure organisation des données. L'automatisation permet également une traçabilité accrue et une meilleure gestion des documents associés aux assemblées.

Le développement de ce système a impliqué une analyse approfondie des besoins, la conception détaillée de la solution, et une implémentation rigoureuse pour garantir sa robustesse et son efficacité.

Le présent rapport qui expose ce travail est constitué de quatre chapitres principaux structurés comme suit :

* Le premier chapitre présente le contexte général de la CIMR, en soulignant son rôle, ses valeurs, et ses choix stratégiques.
* Le deuxième chapitre se concentre sur l'étude du projet, en examinant les besoins qu'il adresse et les objectifs visés, tout en tenant compte des contraintes techniques et organisationnelles.
* Le troisième chapitre est dédié à l'analyse du besoin, visant à identifier les exigences fonctionnelles et non fonctionnelles pour la solution.
* Le quatrième chapitre explore l'étude conceptuelle, en détaillant la conception de la solution à travers divers diagrammes et modèles.
* Enfin, le cinquième et dernier chapitre traite de la réalisation, décrivant la mise en œuvre pratique, les technologies utilisées, et incluant des captures d'écran pour illustrer les résultats obtenus.

# Chapitre 1 : Contexte général

## Introduction

Ce chapitre se concentre sur la présentation globale de la CIMR, en mettant en avant son rôle, ses valeurs fondamentales, et les choix stratégiques qui guident ses actions. Il propose également une vue d'ensemble sur la structure organisationnelle de la CIMR, ses engagements envers ses adhérents, ainsi que les produits et options qu'elle offre.

## Présentation de l’organisme d’accueil :

### Présentation de la CIMR :

Créée en 1949, la CIMR a pour mission de promouvoir un régime de retraite pérenne, équilibré et solidaire, garantissant aux bénéficiaires une pension équitable dans le cadre d’une gestion efficace et de qualité.

La CIMR est gérée par un conseil d’administration élu par l’assemblée générale des adhérents.

La gestion de la CIMR répond aux exigences de performance et de transparence vis-à-vis de ses adhérents et affiliés.

L´instance suprême de décision de la CIMR est l’assemblée générale de ses adhérents.

Le conseil d’administration en constitue l’organe de gestion. Il est composé d’adhérents élus par l’assemblée générale et exerçant à titre bénévole. Quatre comités spécialisés sont issus du conseil d’administration qui leur a confié des missions spécifiques :

* **Le comité de pilotage** : veille sur l´équilibre actuariel du régime
* **Le comité des investissements** : supervise la gestion financière et veille au respect de la charte et du règlement financiers
* **Le comité d’audit** : supervise les missions d´audit interne et externe
* **Le comité des nominations et des rémunérations** : arrête la politique de rémunération et fixe les émoluments des membres de la direction.

A la suite des modifications statutaires introduites par l’assemblée générale extraordinaire du 17 novembre 2016, les comptes de la CIMR sont désormais audités annuellement par deux commissaires aux comptes. Un cabinet d’actuariat indépendant certifie chaque année le bilan actuariel réalisé par la CIMR, sous la houlette du comité de pilotage.

Depuis 2009, la CIMR est certifiée ISO 9001 versions 2015 pour ses deux processus métiers : le processus « Prestations » qui englobe l’ensemble des processus aboutissant à la liquidation des dossiers de pension et au règlement de ces dernières, et le processus « Production » qui couvre les processus d’adhésion des entreprises, l’affiliation et de gestion de la carrière des salariés affiliés.

### Mission :

Administrer un régime de retraite complémentaire pérenne, équilibré et solidaire garantissant aux bénéficiaires une pension équitable reposant sur une gestion efficace, de qualité et une performance financière optimale.

### Valeurs :

* **Honnêteté** : Sens de la justice, de la morale et du devoir ; droiture, intégrité, probité et incorruptibilité, constituent le contenu que la CIMR s´engage à donner à cette valeur dans ses relations externes et internes.
* **Responsabilité** : Assurer ses responsabilités et répondre de ses actes, est pour la CIMR un impératif pour s’acquitter de sa mission. Elle se traduit par un engagement de tous sur les objectifs fixés.
* **Equité** : Donner à chacun ce qui lui est d¹ en toute impartialité, est un principe fondateur pour la CIMR, qui intègre la prise en compte des exigences éthiques les plus élevées de ses métiers.
* **Professionnalisme** : Connaître et maîtriser les règles de ses métiers, les appliquer avec rigueur et exceller dans les domaines techniques et relationnels, constituent pour la CIMR le meilleur moyen pour offrir des prestations de qualité.
* **Serviabilité** : Au cœur de nos actions, l’attention portée à l’individu, qu’il soit adhérent, cotisant, ayant droit, retraité ou citoyen, est fondamentale.
* **Transparence** : La transparence s’applique à la fois aux personnes et aux activités. Elle signifie la mise en pratique du principe d´exemplarité et de respect des règles du jeu.
* **Esprit d'équipe** : Valeur cardinale de la CIMR. Derrière chaque réussite, il y a un groupe, une équipe de femmes et d´hommes.

### Métiers

* Elaborer des produits et prestations adaptés aux besoins évolutifs des affiliés actuels et potentiels
* Développer la base de nos adhérents et de nos affiliés
* Gérer les contributions et assurer leur recouvrement
* Servir les pensions dans une approche client
* Gérer les placements et optimiser leur rendement
* Veiller à l´équilibre du régime

### Choix stratégiques

* L’extension progressive de la couverture de notre régime à l’ensemble des salariés du secteur privé. Cette extension se fera, selon les résultats de la réforme du secteur de la retraite en cours, soit dans le cadre de la généralisation de notre régime, soit à titre facultatif grâce au travail de recrutement que nous entreprenons.
* Le développement et le renforcement des services offerts par la CIMR à ses clients, dans une démarche de progrès continu.
* Le développement de la gestion financière en tant que second métier de la caisse, pour renforcer la part des revenus financiers dans le financement du régime.
* L’amélioration permanente des compétences et du professionnalisme de nos équipes pour qu’elles continent à répondre au mieux aux besoins de nos clients.

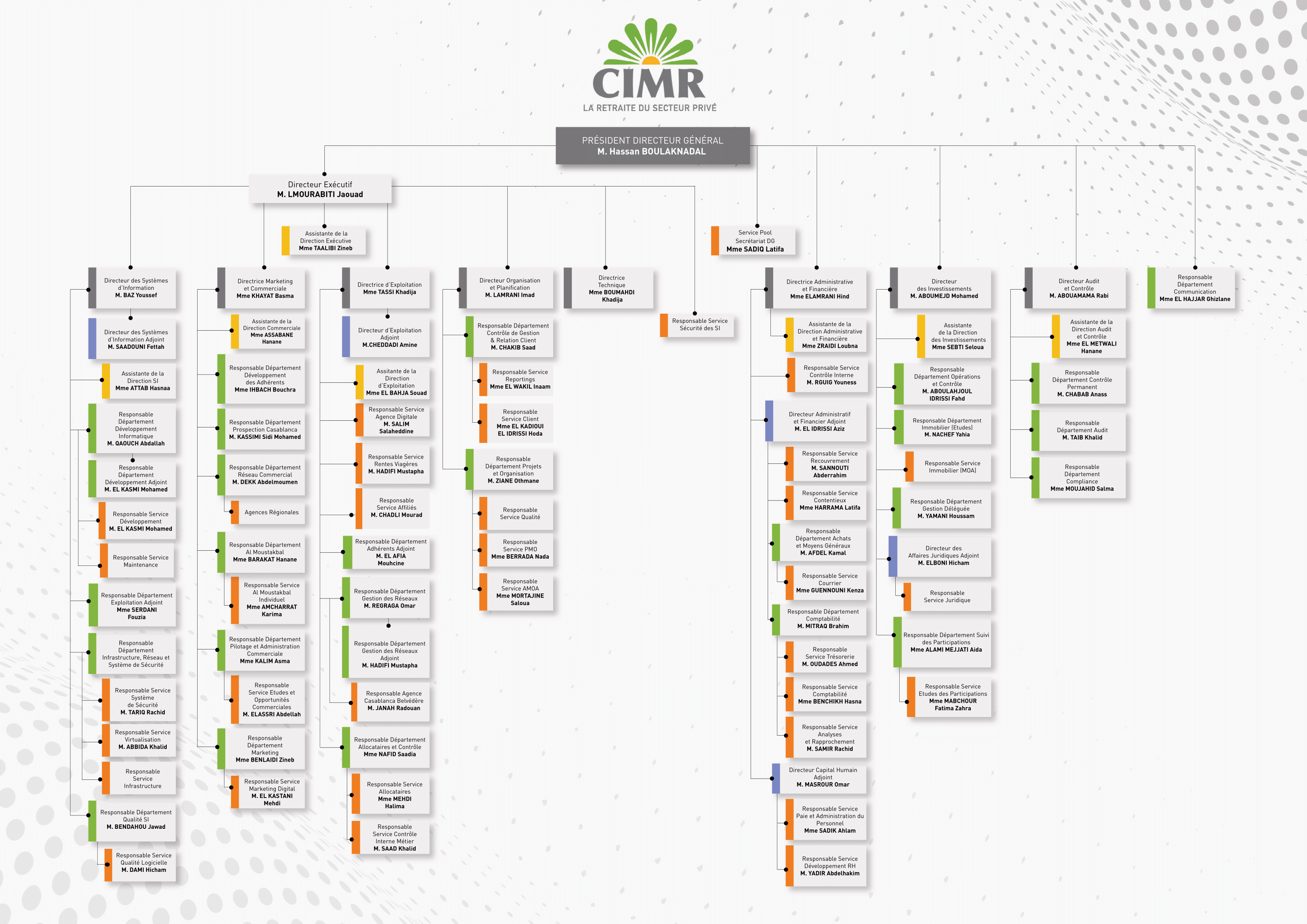
### Fiche technique

|  |  |
| --- | --- |
| **Raison Social** | CIMR |
| **Date de fondation** | 1949 |
| **Forme juridique** | Société Mutuelle de Retraite |
| **Slogan** | Retraite bien préparée, avenir assuré! |
| **Siège sociale** | Casablanca Finance City Casablanca Maroc |
| **Président/Directeur général** | Hassan BOULAKNADAL |
| **Site Web** | [www.cimr.ma/](https://www.cimr.ma/) |
| **Téléphone** | +212 522-424888 |
| **Logo** |  |

### Organigramme

Pour élaborer un projet, l’organigramme permettra de définir de manière brève toutes les fonctions, ce qui aidera dans la répartition efficace des tâches et le suivi des missions. L’organigramme de l’entreprise se présente selon la hiérarchie suivante :

* Dans le sommet de la hiérarchie, on trouve Mr. BOULAKNADAL Hassan qui est le PDG.
* Le corps interne de la CIMR est divisé en sept directions, chacune d’elles se constitue de plusieurs départements.



### Engagements de service

* ***Les engagements pour les allocataires :***
* Transfert des pensions aux banques le premier jour de chaque mois.
* Paiement de la première pension dans un délai de 4 jours ouvrés après activation de la carte RAHATI ou par l’utilisation de l’application CIMR DIALCOM.
* Réalisation d’un changement d’adresse ou de situation familiale dans un délai de 3 jours ouvrés après réception de la demande au niveau du siège ou par l’utilisation de l’application CIMR DIALCOM.
* Mise à disposition des attestations de pension pour usage fiscal au plus tard le 05 Janvier de chaque année téléchargeable depuis de l’application CIMR DIALCOM.
* Analyse et réponse aux réclamations dans un délai de 4 jours ouvrés excepté celles relatives aux pensions virées sur des comptes bancaires hors Maroc.
* ***Les engagements pour les affiliés :***
* Mise à jour des informations dans un délai de 7 jours ouvrés après réception du dossier complet au niveau du siège.
* Traitement des demandes d’abonnement au site dans un délai de 3 jours ouvrés.
* Mise à jour du relevé de points dans un délai de 4 jours ouvrés après confirmation par la banque de l’encaissement couvrant l’échéance concernée.
* Traitement d’une demande d’achat de points dans un délai de 8 jours ouvrés après réception d’un règlement identifié.
* Analyse et réponse aux réclamations dans un délai de 4 jours ouvrés.
* ***Les engagements pour les affiliés :***
* Validation des demandes d’abonnement aux télé services dans un délai de 4 jours ouvrés après réception du dossier complet au niveau du siège.
* Traitement des déclarations de salaires dans un délai de 2 jours ouvrés.
* Traitement des demandes d’adhésion dans un délai de 5 jours ouvrés.
* Traitement d’une demande de changement de taux dans un délai de 5 jours ouvrés.
* Traitement d’une demande de souscription aux options « Moubakkir » et « Moubarak » dans un délai de 5 jour ouvré.
* Analyse et réponse aux réclamations dans un délai de 4 jours ouvrés.
* Réponse aux emails dans un délai de 4 jours ouvrés.

### Produits et options de la CIMR

#### Les produits

La CIMR offre à ses clients le choix entre les produits suivants :

* **« CIMR Al Kamil »** est un régime de retraite ouvert à toutes les entreprises à partir de 3 salariés. Il permet aux salariés dont le revenu est supérieur au salaire plafond de la CNSS ou ceux qui n’y ont pas cotisé suffisamment longtemps de maintenir le même niveau de vie à la retraite.

En optant pour CIMR Al Kamil, l’employeur et ses salariés choisissent ensemble, parmi les taux de contributions proposés, celui qui leur convient le mieux.

* L’offre **« CIMR Al Mounassib »** spécialement dédiée aux petites et moyennes entreprises, a été conçue pour aider le chef d’entreprise à motiver et fidéliser ses collaborateurs en améliorant leur retraite future.

En offrant la possibilité de cotiser sur la tranche de salaire supérieure au plafond de la CNSS, CIMR Al Mounassib améliore le niveau de pension et maintient le niveau de vie à la retraite.

* Avec le produit **« AL Moustakbal Individuel** **»**, toute personne physique pourra prendre la retraite à l’âge qui lui convient à partir de 50 ans. Comme elle aura la possibilité de choisir le montant de contribution mensuelle selon leur objectif en termes de pension, et cette contribution sera prélevée automatiquement à partir de leur compte bancaire.

L’adhésion au produit Al Moustakbal Individuel est possible en toute simplicité et autonomie directement à travers le portail www.cimr.ma.

#### Les options :

A la CIMR, l’âge normal de départ à la retraite est de 60 ans. Cependant, la CIMR permet à ses affiliés de prendre plutôt leur retraite grâce aux options « Mousabbak » et « Moubakkir ».

Ces options ont une double vocation :

1. Améliorer le niveau de la pension CIMR à l’âge de 60ans.
2. Permettre un départ en retraite anticipé avec une pension pleine, sans abattement ou avec un abattement réduit.

En adhérant à l’option **« Mousabbak** **»**, vous payez un taux additionnel fonction de votre taux de contribution à Al Kamil ou Al Mounassib pour bénéficier de points à 55 ans. Vous pouvez alors :

* Partir en retraite à 55 ans avec une pension pleine.
* Décider de continuer à travailler au-delà de cet âge, votre pension est alors bonifiée
* Partir en retraite dès 50 ans avec l’application d’un abattement plus faible

En adhérant à l’option **«** **Moubakkir** **»**, vous payez une surprime additionnelle, dont le taux est déterminé en fonction du taux de contribution à la CIMR, qui vous permet de ne bénéficier de points à 50 ans. Vous pouvez alors partir en retraite à 50 ans avec une pension pleine. Vous pouvez aussi décider de continuer à travailler au-delà de cet âge, votre pension est alors bonifiée.

## Conclusion

En somme, ce chapitre a permis de dresser un portrait global de la CIMR, en soulignant son rôle crucial dans le paysage de la retraite au Maroc. La compréhension de son organisation, de ses valeurs, et de ses engagements est essentielle pour appréhender la portée du projet de gestion des assemblées générales. Les informations présentées ici posent les bases nécessaires pour aborder les aspects plus techniques et spécifiques du projet dans les chapitres suivants.

# Chapitre 2 : Etude du projet

## Introduction

Ce chapitre se propose d'examiner en détail le projet de gestion des assemblées générales, en mettant en lumière les besoins auxquels il répond et les objectifs qu'il vise à atteindre. Nous analyserons les différentes étapes de son développement, de la conception initiale à la mise en œuvre, tout en tenant compte des contraintes techniques et organisationnelles. Cette étude permettra de comprendre les choix méthodologiques et technologiques qui ont guidé la réalisation du projet, ainsi que les défis rencontrés et les solutions apportées.

## Périmètre du projet

### Problématique générale

Les assemblées générales sont au cœur de la gouvernance des organisations, servant de cadre pour prendre des décisions collectives, approuver les rapports financiers, et définir les orientations stratégiques. Pour une organisation comme la CIMR, gérer efficacement ces événements est crucial, mais cela peut rapidement devenir complexe. Chaque étape, depuis la préparation jusqu'au suivi après l'événement, implique une coordination minutieuse, une gestion rigoureuse des documents, et une communication fluide avec les membres.

Malheureusement, les outils et méthodes actuellement en place ne facilitent pas toujours cette tâche. Souvent, les processus sont dispersés entre différents systèmes, ce qui rend la gestion difficile et augmente le risque d'erreurs. Le temps et les efforts nécessaires pour coordonner les acteurs, partager des informations cohérentes, et respecter les délais peuvent devenir accablants. À cela s'ajoute le défi de garantir que toutes les étapes se déroulent conformément aux exigences légales, tout en maintenant un haut niveau de transparence et de traçabilité.

La CIMR se trouve donc à un moment charnière où elle doit repenser sa manière de gérer les assemblées générales. Il devient essentiel de développer une solution numérique qui centralise et automatise ces processus, réduisant ainsi les frictions et les risques associés. Une telle plateforme permettrait non seulement de rendre la gestion plus efficace et plus fiable, mais aussi d'améliorer la communication avec les membres et d'assurer que chaque décision est bien documentée et traçable.

En fin de compte, la véritable question que ce projet cherche à résoudre est la suivante : comment moderniser la gestion des assemblées générales pour qu'elle soit non seulement plus simple et plus efficace, mais aussi plus transparente et alignée avec les attentes des membres et les exigences légales ? La réponse à cette question est au cœur du développement de cette nouvelle solution pour la CIMR.

### Solution :

Pour résoudre les défis posés par la gestion des assemblées générales au sein de la CIMR, il est nécessaire de mettre en place une solution numérique globale capable de centraliser et d'automatiser les processus clés. Cette solution doit offrir une interface intuitive permettant de simplifier la coordination entre les différents acteurs tout en garantissant une gestion rigoureuse et transparente des informations. En centralisant toutes les données pertinentes et en automatisant les tâches répétitives, cette plateforme permettra non seulement de réduire les erreurs humaines, mais aussi d'assurer un suivi précis et efficace des étapes cruciales de l'assemblée générale. De plus, elle devra s'adapter aux exigences légales en constante évolution, tout en offrant la flexibilité nécessaire pour répondre aux besoins spécifiques de la CIMR. Cette transformation digitale vise à améliorer la performance opérationnelle, renforcer la transparence des processus, et garantir une meilleure satisfaction des membres et des parties prenantes.

### But du projet :

Le projet de gestion des assemblées générales pour la CIMR vise à atteindre plusieurs objectifs clés. Tout d'abord, il s'agit de centraliser les informations et les processus liés à la préparation et à la gestion des assemblées générales, afin de faciliter la coordination entre les différentes parties prenantes et de réduire les risques d'erreurs. Ensuite, le projet cherche à automatiser les tâches répétitives et chronophages, telles que l'extraction des listes d'adhérents et la génération de documents, ce qui permettra de gagner en efficacité et en précision. Un autre objectif est d'assurer une conformité stricte aux exigences légales en matière de gestion des assemblées générales, en offrant une traçabilité complète des actions réalisées et en garantissant la sécurité des données. Enfin, le projet a pour but de renforcer l'engagement des membres de la CIMR en améliorant la communication et la transparence tout au long du processus, contribuant ainsi à une meilleure satisfaction et participation des adhérents.

### Missions du projet :

Les missions du projet se déclinent en plusieurs étapes clés :

* **Analyse fonctionnelle** : Comprendre les besoins de la CIMR et identifier les fonctionnalités essentielles et définir les exigences.
* **Étude technique** : Choisir les technologies adaptées, concevoir l'architecture technique et estimer les ressources nécessaires.
* **Analyse et conception** : Modéliser les processus métiers, concevoir les schémas de données et les interfaces utilisateur.
* **Réalisation** : Développer la plateforme tout en veillant à la qualité, à la sécurité, et à la conformité aux normes en vigueur.

### Livrables

|  |  |
| --- | --- |
| Livrable | Description |
| Cahier des charges | Le document qui décrit en détail les objectifs, les exigences et le plan d'action du projet. |
| Étude des besoins | Un rapport détaillé basé sur une analyse approfondie des besoins de l'utilisateur, des exigences du marché et des objectifs de l'entreprise. |
| Code source de l'application | L'ensemble du code écrit pour l'application. Il comprendra le code pour toutes les fonctionnalités décrites dans le cahier des charges et l'étude des besoins. |
| Documentation technique | Un ensemble de documents expliquant comment l'application a été construite et comment elle fonctionne. |
| Application testée et fonctionnelle | Une version de l'application qui a été complètement testée et est prête à être déployée. |
| Rapport de projet | Un document final qui résume le travail accompli pendant le projet. |

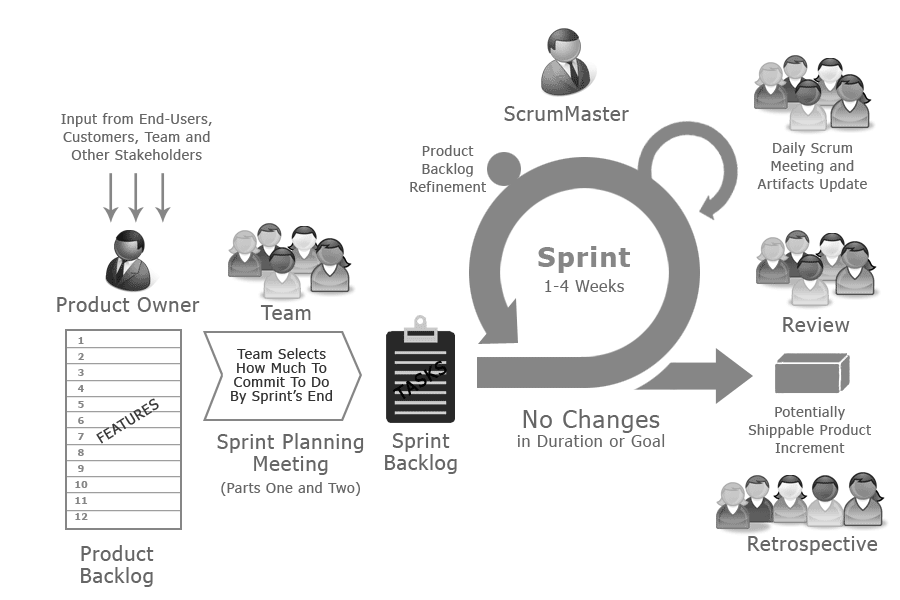
### Risques du projet :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risque | Type | Impact | Probabilité | Actions correctives |
| CDC incomplet | Risque non bloquant | Créé une ambigüité ce qui pourra générer un retard qui peut-être influencera la date de livraison | Moyenne | Prévoir des réunions avec l’encadrant au fur et à mesure de l’avancement du projet. |
| Les pannes du matériel. | Risque bloquant | Ralentissement des travaux | Forte | Utiliser des autres matériaux dispos. |
| Les erreurs et les bugs | Risque non bloquant | Ralentissement des travaux | Forte | Lancer des questions sur les forums et travailler les autres modules restants. |
| Absence ou maladie | Risque non bloquant | Ralentissement des travaux | Faible | Doubler l’effort et travailler un temps extra |

## Conduite du projet :

### Méthodologie de développement

La méthode adoptée dans de ce projet est la méthode agile SCRUM. A l'instar des autres méthodes agiles, SCRUM est une démarche de gestion de projet qui fait du client (ou utilisateur) le principal pilote de l'équipe en charge des développements. Historiquement, elle est principalement mise en œuvre dans le domaine informatique, et dans celui du développement d'applications en particulier. Le principe de base de SCRUM est de focaliser l’équipe de façon itérative sur un ensemble de fonctionnalités à réaliser, dans des itérations de durée fixe de deux à quatre semaines, appelées Sprints. Chaque Sprint possède un but à atteindre, défini par le directeur de produit, à partir duquel sont choisies les fonctionnalités à implémenter dans ce sprint. Un principe fort en SCRUM est la participation active du client pour définir les priorités dans les fonctionnalités du produit, et pour choisir celles qui seront réalisées dans chaque sprint.



Le Backlog du produit définit les modules du projet tout entier qui est ensuite divisé en plusieurs sous modules constituant le Backlog du sprint. Les principales caractéristiques de cette méthodologie sont :

* Les projets SCRUM progressent par une série de sprints
* La durée d’un sprint est de 2 à 4 semaines
* Une durée constante apporte un meilleur rythme
* Le produit (partiel) est conçu, codé et testé pendant le sprint

### Pourquoi SCRUM ?

Nous avons opté pour cette méthode car la mise en œuvre de notre projet nécessite une vision globale de la solution à mettre en place, que ce soit au niveau technique, architectural ou fonctionnel. L’aspect itératif et incrémental de notre projet constituent une autre raison qui est aussi importante que la première sachant qu’il s’agit de prévoir des itérations dont le résultat serait validé. Ces itérations doivent être efficaces avec une durée minimisée et une qualité maximum des livrables. Développement d’une plateforme de mise relation entre investisseurs et porteurs de projets. En SCRUM le changement est accepté, même tardivement dans le développement. Les processus agiles exploitent le changement comme avantage compétitif pour le client.

### Cycle de vie :

#### Phases du cycle retenu

Les phases du cycle de vie du projet sont les suivantes :

* **Eude préalable :** Cette phase consiste à définir les besoins fonctionnels du projet sous la forme d'un cahier des charges. Elle inclut également l’identification de l’architecture technique et la planification globale du projet.
* **Analyse et conception :** Basée sur l’analyse de l’existant, cette phase permet de concevoir l’aspect graphique, fonctionnel et technique du projet. Un prototype technique et une maquette sont réalisés pour valider l’architecture technique et s'assurer qu'elle répond aux besoins des utilisateurs.
* **Réalisation :** Cette phase est dédiée à l’implémentation technique de l’application. Celle-ci est testée, puis déployée sur un environnement de test. La phase de recette est effectuée en deux étapes : une première pour le FO, puis une seconde pour le BO.
* **Déploiement** : L'objectif de cette phase est la mise en production de l’application, accompagnée d'une assistance pour garantir son bon fonctionnement.
* **Documentation et Formation** : Cette phase comprend la rédaction de la documentation technique, la création des guides utilisateurs, ainsi que la formation des utilisateurs finaux.

### Description des phases :

* Phase d’étude préalable

|  |  |
| --- | --- |
| Phase d’étude préalable | |
| Objectifs | * Rédaction et approbation du CDC * Étude technique et ergonomique * Planification |
| Étapes de la phase | * CF Planning |
| Livrables en sortie | * Cahier des charges * Charte graphique * Dossier d’étude technique * PQP |
| Prérequis | * N/A |
| Contrainte | * N/A |
| Dépendance | * N/A |
| Critère de fin de phase | * Cahier des charges approuvé * Étude technique validée avec le MOA (PO) * Charte graphique approuvée |

* Phase d’analyse et conception

|  |  |
| --- | --- |
| Phase d’analyse et conception | |
| Objectifs | * Analyse fonctionnelle générale et détaillée du projet |
| Étapes de la phase | * CF Planning |
| Livrables en sortie | * Dossier des spécifications fonctionnelles du projet * Dossier de conception |
| Prérequis | * Cahier des charges * PQP |
| Contrainte | * Respecter les exigences du PQP |
| Dépendance | * Cette phase dépend de la phase d’étude préalable |
| Critère de fin de phase | * Approbation de la spécification fonctionnelle et la conception par le client (MOA) |

* Phase de réalisation

|  |  |
| --- | --- |
| Phase de réalisation | |
| Objectifs | * Développer l’ensemble des modules de l’application * Test des différents modules |
| Étapes de la phase | * CF Planning |
| Livrables en sortie | * Backlog du projet * Rapport de test : |
| Prérequis | * Dossier de spécifications fonctionnelles * Dossier d’étude technique |
| Contrainte | * Respecter l’architecture technique du projet et du dossier de spécification fonctionnelles |
| Dépendance | * Cette phase dépend de la phase d’analyse et conception |
| Critère de fin de phase | * Démonstration effectuées et validées par le MOA (PO) * Toutes les évolutions demandées effectuées * Dossier de test approuvé |

* Phase de déploiement

|  |  |
| --- | --- |
| Phase de déploiement | |
| Objectifs | * Préparation de l’environnement du déploiement * Déploiement de la solution dans l’environnement du client * Formation |
| Étapes de la phase | * CF Planning |
| Livrables en sortie | * Manuel d’utilisation * Solution déployée |
| Prérequis | * Rapport de test * Étude technique (Architecture de déploiement) |
| Contrainte | * Respecter l’environnement de déploiement du client |
| Dépendance | * Cette phase dépend de la phase de réalisation |
| Critère de fin de phase | * Livraison acceptée par le MOA (PO) * Formation effectuée * Application mise en production avec succès |

### Planning du projet :

L’élaboration du planning s’est basée sur les hypothèses suivantes :

* Le planning peut subir des modifications suivant les impératifs du projet.
* Le planning est revu avec le client à l’occasion de chaque comité de projet ou comité de pilotage.

## Organisation du projet :

### Rôles des acteurs du projet

* **Chef du projet**
  + Est l’interlocuteur principal de la société
  + Est responsable de la conception fonctionnelle et technique de l’application
  + Participe aux comités de projets et pilotage
  + Mène la phase de recette en assistant dans l’appropriation de l’application et la qualification des anomalies/demandes de changement
  + Analyse l’impact des demandes de changements
* **Responsable du projet**
  + Pilote toutes les dimensions techniques et fonctionnelles
  + Valide les spécifications fonctionnelles et technique
  + Participe au pilotage du projet
  + Coordonne les actions de l’ensemble des autres intervenants techniques
  + Responsable de la mise en œuvre des plates-formes de pré production, qualification et production
  + L’interlocuteur principal pour la phase de conception fonctionnelle, participe à toutes les réunions de conception et ateliers de travail
  + Est l’interface avec les utilisateurs finaux de l’application, et assure la transmission des informations entre ceux-ci et le pool d’experts
  + Assume la validation des livrables fonctionnels dans les délais
* **Encadrant**
  + Valide le cahier des charges
  + Valide la conception
  + S’assure de la cohérence entre le Product back et le PQP
  + Valide les livrables documentaires

## Suivi du projet :

Nous avons utilisé un échange de messagerie avec l’encadrant de l’entreprise (PO) et l’encadrant pédagogique, les réunions de suivi sont organisées en 3 catégories :

* **Réunion fonctionnelle** avec l’encadrant de l’entreprise (PO), dont l’objectif est de valider colleter les exigences fonctionnelles et technique du projet, ainsi que les modifications demandées.
* **Réunions de démonstration** *(phase de développement)* : l’objectif de ces réunions est :
  + La démonstration des sprints réalisées
  + La validation des autres livrables du projet
  + La collecte des demandes de modification de la part du PO ;
  + La préparation du déploiement
* **Réunion d’encadrement** *(hebdomadaire)* avec l’encadrant pédagogique : l’objet de cette réunion est consacré à :
  + Le suivi technique et fonctionnel du projet
  + La pré approbation des livrables du projet et les documents pédagogiques
  + La revue du rapport et la validation de la présentation du projet

#### Moyens de communication :

Les moyens utilisés pour le projet sont :

* La messagerie électronique
* Les outils de vidéoconférence : Zoom, Google Meet.

## Conclusion

Ce chapitre a permis de dresser un cadre clair et structuré du projet en définissant son périmètre, ses objectifs, et les défis qu'il doit relever. En identifiant les risques et en choisissant une méthodologie de développement adaptée, notamment SCRUM, nous avons mis en place une approche rigoureuse pour mener à bien le projet. La description des phases du cycle de vie et l'organisation des rôles des acteurs du projet montrent l'importance d'une planification et d'un suivi efficaces pour assurer le succès du projet. L'ensemble de ces éléments pose les bases solides pour la phase de réalisation qui suivra.

# Chapitre 3 : Analyse du besoin

## Introduction

Ce chapitre vise à identifier et formaliser les besoins fonctionnels et non fonctionnels pour la mise en place d'une solution de gestion des assemblées générales au sein de la CIMR. L'objectif est de comprendre en profondeur les attentes de l'organisation, les processus actuels, ainsi que les contraintes spécifiques à son environnement. Cette analyse permet de définir les fondations sur lesquelles sera développée la solution, en s'assurant que celle-ci répondra de manière optimale aux exigences des utilisateurs et aux impératifs de performance, de sécurité, et de conformité.

## Cahier des charges :

### Problématique

L'élaboration du cahier des charges pour le projet de gestion des assemblées générales au sein de la CIMR constitue une étape cruciale, mais complexe. Cette tâche nécessite une compréhension approfondie des besoins et des attentes de l'organisation, ainsi que des contraintes spécifiques à son fonctionnement. La problématique se pose dès lors qu'il s'agit de formaliser ces exigences de manière exhaustive et précise, en tenant compte des diverses parties prenantes, chacune ayant des besoins et des priorités parfois divergentes. Il devient alors indispensable de concilier ces différents intérêts tout en veillant à ce que la solution finale réponde aux objectifs globaux de la CIMR.

Un autre aspect de cette problématique réside dans la traduction de ces exigences fonctionnelles en spécifications techniques claires et réalisables. Cela implique de prendre en compte non seulement les aspects métiers, mais aussi les contraintes techniques, les impératifs de sécurité, et la conformité aux réglementations en vigueur. Le risque de malentendus ou d'interprétations erronées est élevé, ce qui peut conduire à des dérives de projet ou à l'insatisfaction des utilisateurs finaux.

De plus, la complexité du projet, avec ses multiples modules et fonctionnalités, ajoute une couche supplémentaire de difficulté dans l'élaboration d'un cahier des charges complet et cohérent. Il est essentiel de définir des critères de succès clairs et des indicateurs de performance qui permettront de mesurer l'efficacité de la solution mise en place. Comment s'assurer que le cahier des charges capturera fidèlement toutes les exigences nécessaires tout en restant flexible pour s'adapter aux évolutions futures ? Cette question souligne l'importance d'une approche rigoureuse et collaborative dans la phase d'analyse du besoin.

### Etude de l’existant

#### Concept métier :

À la CIMR, la gestion des assemblées générales est un processus central qui implique plusieurs étapes clés : la préparation des documents, la convocation des membres, et la tenue des réunions. Ces assemblées sont essentielles pour la prise de décisions stratégiques et la validation des rapports financiers. Actuellement, ces processus, souvent gérés de manière manuelle, peuvent conduire à des inefficacités et des erreurs. Afin d'améliorer l'efficacité et de répondre aux exigences réglementaires, il est nécessaire d'adopter une solution numérique capable de centraliser et d'automatiser ces tâches.

### Etude fonctionnelle :

#### Besoins fonctionnels :

* Préparation d’une assemblée générale :

|  |  |
| --- | --- |
| **Fonctionnalité** | **Description** |
| **F01 :**  **Préparation d’une assemblée générale** | **Données d’entrée :**   * Accès utilisateur CIMR |
| **Règles de gestion :**   * La préparation d’une assemblée générale est répartie selon les étapes suivantes :  |  |  | | --- | --- | | **Étape** | **Description** | | **Création d’une assemblée générale** | Saisir les informations d’une assemblée générale (Type, Année, Jour, Heure, Adresse, Ville)  Joindre les pièces suivantes :   * Feuille de route FDR * Modèle de la lettre de convocation * Modèle de la feuille de présence FP * Modèle du pouvoir * Fiche de présence   Avec possibilité d’ouvrir la PJ à partir de l’application, de la modifier et de l’enregistrer sans avoir recours à la télécharger de nouveau. | | **Création automatique de la feuille de route** | La feuille de route de l’organisation de l’assemblée générale, comprenant les actions à réaliser par les entités concernées, est créée automatiquement à base de la FDR en annexe insérée dans l’étape de création de la nouvelle assemblée générale. Avec possibilité de joindre au besoin les livrables issus d’une action donnée. | | **Extraction de la liste des adhérents** | L’extraction est exécutée automatiquement par défaut huit (8) semaines avant le jour de l’assemblée générale mentionné dans l’étape de création. Cette extraction comporte les adhérents remplissant les deux conditions suivantes :  **Condition 1** :   * Pour les adhérents de type 1 : A jour le premier jour du mois de la convocation, de leurs Contributions échues au moins six (6) mois avant cette date : * Pour les Adhérents de type 02, 03 et 04 : Avoir effectué des versements de Contributions dans les six (6) mois précédant le premier jour du mois de la date de la convocation, si lesdites contributions sont forfaitaires.   **Condition 2** : Avoir au moins 50 affiliés ;  NB : Le Nombre d’affiliés est calculé selon la règle suivante :   * Pour les adhérents de type 1 : Nombre d’affiliés du DTR ; * Pour les adhérents de type 2 ou 3 : Nombre d’affiliés du DTR.   Ces adhérents sont affichés dans un menu dédié pour génération automatique de la convocation, de la feuille de présence et du pouvoir selon les modèles insérés dans l’étape de création dont des exemples joints en annexe. Ceux disposant de moins de 50 affiliés seront affichés dans un autre menu séparé pour les identifier dans les demandes de regroupement en annexe. | | **Révision de l’extraction** | La liste des adhérents à convoquer comprend les informations suivantes :   * N° Adhérent * Type * Raison sociale * Adresse 1 * Adresse 2 * Ville * Téléphone 1 * Téléphone 2 * Date adhésion * Effectif * Titre * Trimestre DBR * Année DBR * Trimestre DTR * Année DTR | | **Impression des convocations** | Les lettres convocations, feuilles de présence et les pouvoirs seront imprimés par ordre croissant des N° Adhérents. | |
| **Données de sortie :**   * Nouvelle assemblée générale + liste des adhérents à convoquer + génération des lettres de convocation + feuilles de présence + pouvoirs |
| **Acteurs :**   * DPO |

|  |  |  |
| --- | --- | --- |
| **DPO** | **Agent de relance** | **DAMG** |
| Création d’une nouvelle assemblée générale  Joindre les modèles des documents applicables à l’AG  Suivre l’avancement des actions de la FDR créée automatiquement à partir du document inséré  A une date précise  Injection automatique dans l’application de l’extraction des adhérents à convoquer et ceux éligibles par regroupement | Editer les convocations + FP + Pouvoir  Saisir les motifs du retour puis entamer les relances | Déposer les convocations à la poste  Retour courrier  Non  Oui  Fin |

* Administration :

|  |  |
| --- | --- |
| **Fonctionnalité** | **Description** |
| **F02 :**  **Administration** | **Données d’entrée :**   * Accès administrateur CIMR |
| **Règles de gestion :**  Le menu d’administration permet entre autres de :   * Créer et mettre à jour les profils et les habilitations d’accès aux différents menus de l’application de gestion de l’assemblée générale * Paramétrer les différentes cases à cocher |
| **Données de sortie :**   * Création des profils + Gestion des utilisateurs + Paramétrage des différentes listes déroulantes |
| Acteurs :   * DPO |

## Conclusion

L'analyse des besoins a permis de dégager les principales fonctionnalités et exigences indispensables à la réussite du projet de gestion des assemblées générales. Cette étape cruciale a non seulement mis en lumière les processus métiers à automatiser, mais aussi les contraintes techniques et organisationnelles à prendre en compte. Les éléments identifiés dans ce chapitre serviront de guide tout au long des phases de conception, de développement, et de mise en œuvre de la solution, garantissant ainsi qu'elle répondra pleinement aux besoins de la CIMR.

# Chapitre 4 : Etude conceptuelle

## Introduction

Dans ce chapitre, nous allons explorer les différentes étapes de la conception, en mettant l'accent sur la manière dont nous avons structuré les processus clés de l'application. À travers l'utilisation de divers diagrammes et modèles, tels que les diagrammes de cas d'utilisation, de classes, nous allons détailler comment la solution a été conceptualisée pour répondre aux besoins identifiés, tout en garantissant une architecture robuste, extensible, et sécurisée.

## Langage UML

### Présentation :

Le langage UML (ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement.

L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientées objet.

L’UML est un outil très puissant qui permet de :

* Représenter des concepts abstraits (graphiquement par exemple)
* Limiter les ambiguïtés (parler un langage commun, au vocabulaire précis, indépendant des langages orientés objet)

### UML et POO

L'un des principaux paradigmes de la programmation moderne est la programmation orientée objet (POO). Dans ce paradigme, les algorithmes sont formulés en définissant des "objets" et en orchestrant leurs interactions. Les objets représentent généralement des entités du monde réel et peuvent être aussi divers que des bâtiments, des widgets sur un ordinateur de bureau, ou même des êtres humains. UML a une relation directe et profonde avec la POO. En effet, il est né de la fusion de plusieurs notations orientées objet, la conception orientée objet (OOD), la technique de modélisation objet (OMT) et l'ingénierie logicielle orientée objet (OOSE). UML combine les forces de ces trois approches pour présenter une méthodologie plus cohérente, unifiée et facile à utiliser. Il représente en quelque sorte les meilleures pratiques émergentes pour créer et documenter différents aspects de la modélisation des logiciels et des systèmes d'entreprise.

### Les avantages d’UML

Le langage UML se distingue par plusieurs avantages dont on peut citer :

* Il est riche : vu qu'il couvre toutes les phases d'un cycle de développement.
* Il est ouvert : vu qu'il est indépendant du domaine d'application et des langages d’implémentation
* Il s'agit d'un langage formel et normalisé : ce qui favorise le gain de précision et garantie la stabilité

### Utilité de l’UML selon l’OMG :

L'OMG définit les objectifs de l'UML comme suit :

* Fournir aux concepteurs de systèmes, ingénieurs logiciels et développeurs de logiciels des outils pour l'analyse, la conception et la mise en œuvre de systèmes logiciels, ainsi que pour la modélisation de processus métier et d'autres processus similaires.
* Faire progresser l'industrie en permettant l'interopérabilité des outils de modélisation visuelle orientés objet. Toutefois, pour permettre un échange significatif d'informations de modèles entre outils, il est nécessaire de trouver un accord sur la sémantique et la notation.

UML répond aux exigences suivantes :

* Fournir une explication détaillée de la sémantique de chaque concept de modélisation UML. La sémantique définit, d'une façon indépendante de la technologie, comment les concepts UML doivent être mis en œuvre par les ordinateurs.
* Spécifier des éléments de notation lisibles par l'homme pour représenter chaque concept de modélisation UML, ainsi que les règles pour les combiner au sein d'une grande variété de diagrammes correspondant à différents aspects des systèmes modélisés.
* Fixer une définition formelle d'un métamodèle basé sur une norme MOF commune qui spécifie la syntaxe abstraite de l'UML. La syntaxe abstraite définit l'ensemble des concepts de modélisation UML, leurs attributs et leurs relations, ainsi que les règles permettant d'associer ces concepts afin de créer des modèles UML partiels ou complets.
* Définir des moyens grâce auxquels les outils UML peuvent être mis en conformité avec cette spécification. Ceci est pris en charge (dans une spécification distincte) par une spécification XML des formats d'échange de modèles correspondants (XMI) qui doivent être réalisés par des outils conformes.

### Identification des acteurs :

Les acteurs sont les entités qui interagissent directement avec le système que nous développons.

#### Les acteurs de l’application

Dans le cadre de ce projet, l'acteur principal est **User**. Cet acteur est au cœur de toutes les interactions avec la plateforme de gestion des assemblées générales, assurant que les diverses fonctionnalités du système sont utilisées de manière optimale.

#### Rôles des acteurs :

Le rôle de **User** consiste à interagir avec le système pour accomplir les tâches nécessaires au bon déroulement des assemblées générales, depuis leur préparation jusqu'à leur clôture. Cet acteur est crucial pour garantir que le système réponde aux besoins identifiés lors de l'analyse et fonctionne conformément aux attentes de l'organisation.

### Diagramme de paquetages :

**Un diagramme de paquetages** (ou diagramme de packages) est un diagramme structurel utilisé pour représenter l'organisation et la disposition de divers éléments modélisés sous forme de paquetage. Un paquetage est un regroupement d'éléments UML apparentés, tels qu'un diagramme, un document, une classe ou même un autre paquetage. Tous les éléments du diagramme sont imbriqués dans un paquetage, qui est lui-même représenté sous forme de dossier de fichiers et organisé de manière hiérarchique. Un diagramme de paquetage est le plus souvent utilisé pour donner un aperçu visuel de l'architecture en couches d'un classifieur UML, tel qu'un système logiciel. Un diagramme de package bien conçu offre de nombreux avantages à la personne souhaitant créer une représentation graphique de son système ou projet UML.

* Il fournit une perspective précise de la structure hiérarchique des différents éléments UML au sein d'un système donné.
* Ce diagramme permet de simplifier un diagramme de classes complexe sous une forme visuelle bien ordonnée.
* Il offre une vue d'ensemble précieuse pour un projet ou système de grande ampleur.
* Un diagramme de paquetage peut être utilisé pour clarifier visuellement de nombreux types de projets et systèmes.
* Ce visuel peut être facilement mis à jour au fur et à mesure de l'évolution des systèmes et des projets.

### Diagramme des cas d’utilisation :

En langage de modélisation unifié (UML), un diagramme de cas d'utilisation peut servir à résumer les informations des utilisateurs de votre système (également appelés acteurs) et leurs interactions avec ce dernier. La création de ce type de diagramme UML requiert un ensemble de symboles et de connecteurs spécifiques. Lorsqu'ils sont bien conçus, les diagrammes de cas d'utilisation peuvent aider votre équipe à collaborer et représenter :

* Les scénarios dans lesquels votre système ou application interagit avec des personnes, des organisations ou des systèmes externes
* Les objectifs que votre système ou application permet aux entités (appelées acteurs) d'atteindre
* La portée de votre système

### Diagramme de séquence :

Un diagramme de séquence est un diagramme UML qui représente la séquence de messages entre les objets au cours d'une interaction. Un diagramme de séquence comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction. Les diagrammes de séquence représentent la séquence de messages transmis entre des objets. Ils peuvent également représenter les structures de contrôle entre des objets. Par exemple, les lignes de vie dans un diagramme de séquence pour un scénario de banque peuvent représenter un client, un guichetier ou un responsable d'agence. Les communications entre le client, le guichetier et le responsable sont représentés par les messages entre ces derniers. Le diagramme de séquence représente les objets et les messages entre ces objets.

### Diagramme des classes :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations. Ce diagramme fait partie de la partie statique d'UML, ne s'intéressant pas aux aspects temporels et dynamiques. Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe. Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples. Les classes peuvent être reliées grâce au mécanisme d'héritage qui permet de mettre en évidence des relations de parenté. D'autres relations sont possibles entre des classes, représentées par un arc spécifique dans le diagramme de classes. Elles sont finalement instanciées pour créer des objets (une classe est un moule à objet : elle décrit les caractéristiques des objets, les objets contiennent leurs valeurs propres pour chacune de ces caractéristiques lorsqu'ils sont instanciés).

## Conclusion

L'étude conceptuelle a permis de transformer les besoins fonctionnels et techniques identifiés lors de l'analyse en une architecture solide et des modèles clairs, prêts à être implémentés. Les différents diagrammes et conceptions élaborés au cours de cette phase ont posé les bases essentielles pour le développement de la solution, en assurant une cohérence entre les exigences du système et les choix techniques effectués. Cette étape cruciale garantit que le projet avancera sur des fondations bien établies, permettant ainsi de répondre efficacement aux attentes de la CIMR et de fournir une solution robuste, évolutive, et sécurisée.

# Chapitre 5 : Etude technique

## Introduction

Ce chapitre présente les outils, technologies, et architectures utilisées pour développer la solution de gestion des assemblées générales. Il couvre les logiciels et langages de programmation, ainsi que les frameworks et bibliothèques employées. De plus, l'architecture de l'application et sa structure sont décrites pour offrir une vue d'ensemble des choix techniques qui ont guidé le projet.

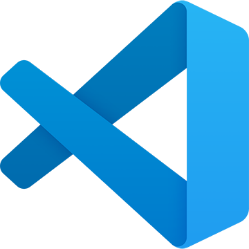
## Outils de travail :

### Logiciels :

#### IntelliJ IDEA Ultimate

IntelliJ IDEA également appelé IntelliJ, est un environnement de développement destiné au développement de logiciels informatiques reposant sur la technologie Java. Il est développé par JetBrains et disponible en deux versions, l'une communautaire, open source, sous licence Apache 2 et l'autre propriétaire, protégée par une licence commerciale. Tous deux supportent les langages de programmation Java, Kotlin, Groovy et Scala.

#### Visual Studio Code

Visual Studio Code est un éditeur de code source et un environnement de développement intégré de Microsoft. Il est open-source et cross-platform, c’est-à-dire qu’il fonctionne sur Windows, Linux et Mac. Il a été conçu pour les développeurs web, mais il prend en charge de nombreux autres langages de programmation tels que C++, C#, Python, Java, etc.

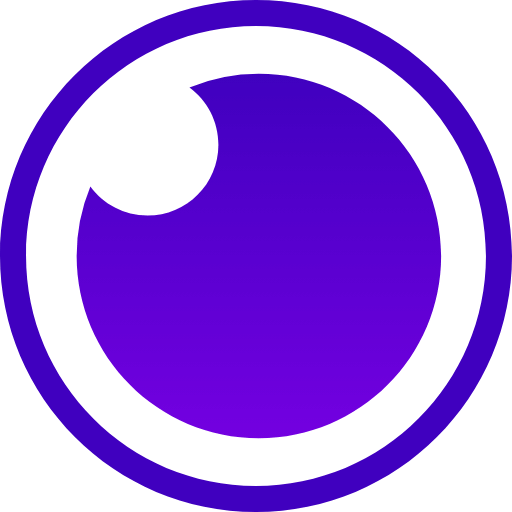
#### Enterprise Architect

Enterprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne Sparx Systems. Couvrant, par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application, il est l'un des logiciels de conception et de modélisation les plus reconnus.

#### HeidiSQL

HeidiSQL est un outil d'administration de base de données possédant un éditeur SQL et un constructeur de requête. Il a été développé et optimisé pour être utilisé avec le SGBD relationnel MySQL disponible commercialement ou gratuitement.

#### Insomnia

Insomnia est un puissant client REST qui vous permet d’envoyer des requêtes HTTP et de visualiser les réponses de REST APIs. Il est conçu pour être facile à utiliser et aide les développeurs à déboguer et à tester leurs APIs de manière plus efficace. Avec Insomnia, vous pouvez créer et enregistrer des requêtes, les organiser dans des espaces de travail et les partager avec d’autres.

#### Brave

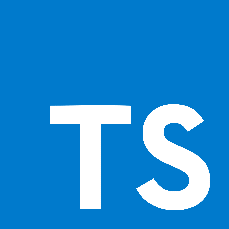
Brave est un navigateur web open source gratuit disponible sur Windows, macOS et Linux ainsi que sur iOS et Android. Il a pour objectif de protéger la vie privée de ses utilisateurs en bloquant par défaut les pisteurs et en permettant la navigation via le réseau Tor. Le logiciel, construit sur Chromium, est développé par l’entreprise Brave Software depuis 2016.

### Langages et technologies de développement

#### Java 17

Java est un langage de programmation et une plate-forme de calcul lancé par Sun Microsystems en 1995. Depuis ses débuts modestes, Java a beaucoup évolué. A l'heure actuelle, une grande partie du monde numérique dépend de Java : de nombreux services et applications reposent sur cette plate-forme fiable. De même, de nouveaux produits et services numériques innovants et tournés vers l'avenir dépendent aussi de Java.

#### TypeScript

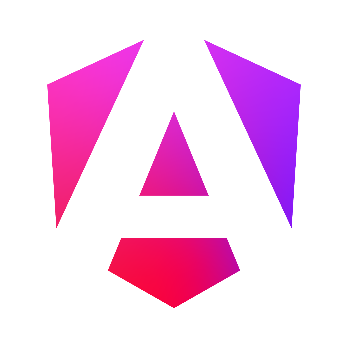
TypeScript est un langage de programmation développé par Microsoft en 2012. Son ambition principale est d’améliorer la productivité de développement d’applications complexes. C’est un langage open source, développé comme un sur-ensemble de Javascript. Ce qu’il faut comprendre, c’est que tout code valide en Javascript l’est également en TypeScript.

### Framework et bibliothèques utilisés

#### Spring Framework

Spring est le framework le plus utilisé pour faciliter le développement et les tests d’applications Java. C’est un framework open source et léger permettant aux développeurs Java et Java EE de concevoir des applications d’entreprise. Il permet de résoudre de nombreux problèmes rencontrés par ces développeurs tels que les problèmes de connectivité ou de sécurités.

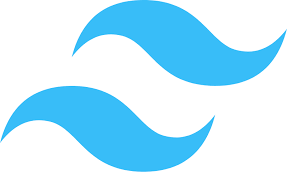
#### Angular

Angular est un framework open source JavaScript développé par Google. Ce framework est utilisé pour développer des applications web et mobile. Avec cette technologie, on réalise des interfaces de type monopage qui fonctionnent sans rechargement de la page web. Angular propose un ensemble de conventions et d’outils pour délimiter les bases d’une solution. Les développements sont ainsi optimisés, s’effectuent plus rapidement et de manière plus sûre. En tant que langage open source, il est régulièrement alimenté et enrichi par une communauté importante de contributeurs. Ces derniers aident à l’amélioration du framework dans le temps grâce des à des mises à jour récurrentes.

#### Apache POI

Apache POI (Poor Obfuscation Implémentation) est une bibliothèque Java open source populaire développée par la Fondation des logiciels Apache. POI signifie « Mauvaise mise en œuvre de l’obscurcissement » faisant référence avec humour aux formats de fichiers binaires propriétaires de Microsoft. L’objectif principal d’Apache POI est de fournir aux développeurs Java un ensemble d’API qui leur permet de lire, d’écrire et de manipuler divers formats de fichiers Microsoft Office, tels que des feuilles de calcul Excel (.xls et .xlsx), Word Documents (.doc et .docx) et PowerPoint Présentations (.ppt et .pptx).

#### Tailwind CSS

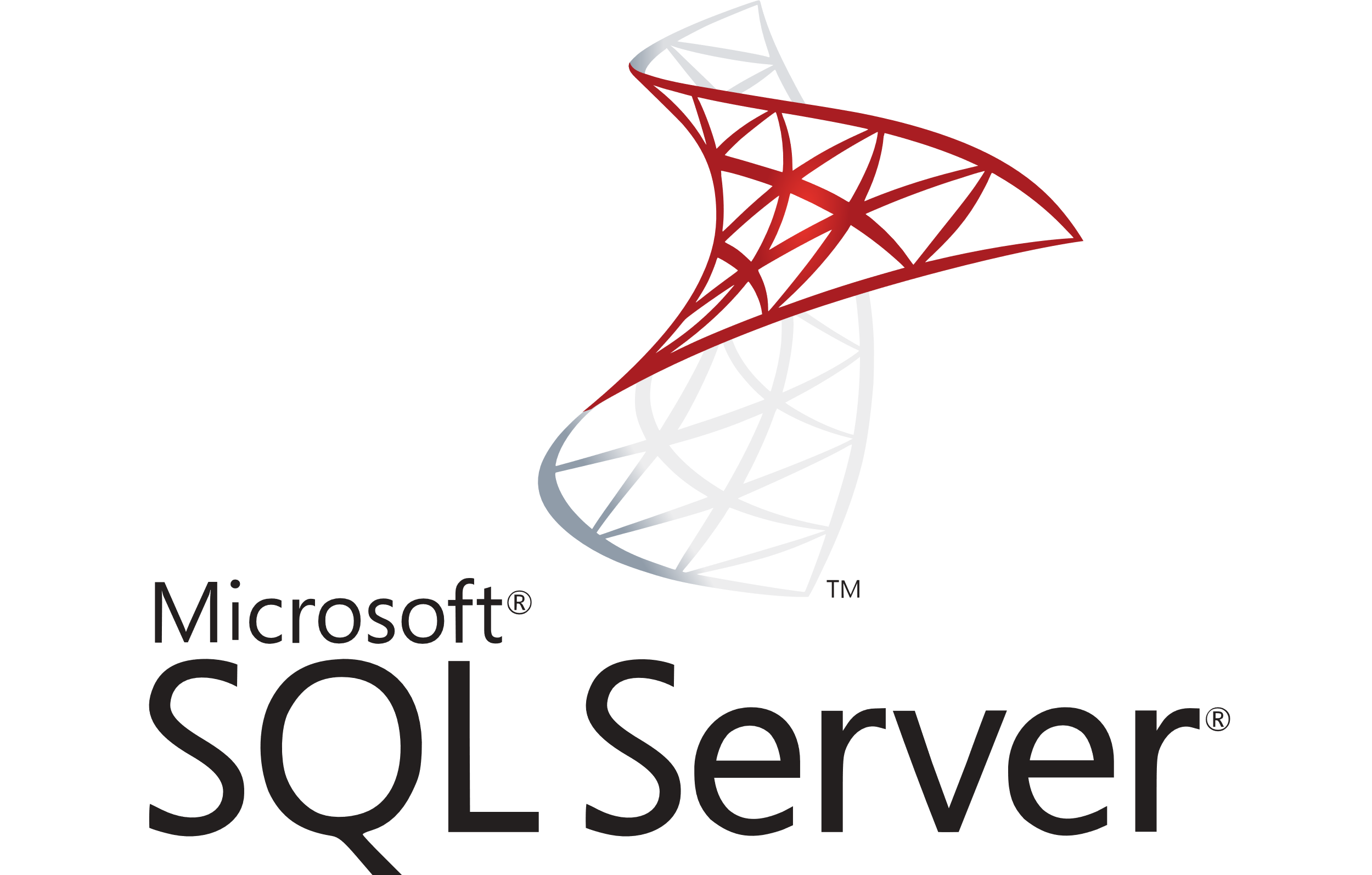
Tailwind CSS est un framework permettant aux développeurs de personnaliser totalement et simplement le design de leur application ou de leur site web. Avec ce framework CSS, il est possible de créer un design d’interface au sein même du fichier HTML. Cette façon de programmer n’interfère pas avec les pratiques recommandées par le W3C comme celle de séparer le HTML des feuilles de style CSS.

#### Flowbite

Flowbite est une bibliothèque open source de composants d'interface utilisateur basée sur le premier framework CSS Tailwind, offrant la prise en charge du mode sombre, un système de conception Figma, des modèles et bien plus encore.

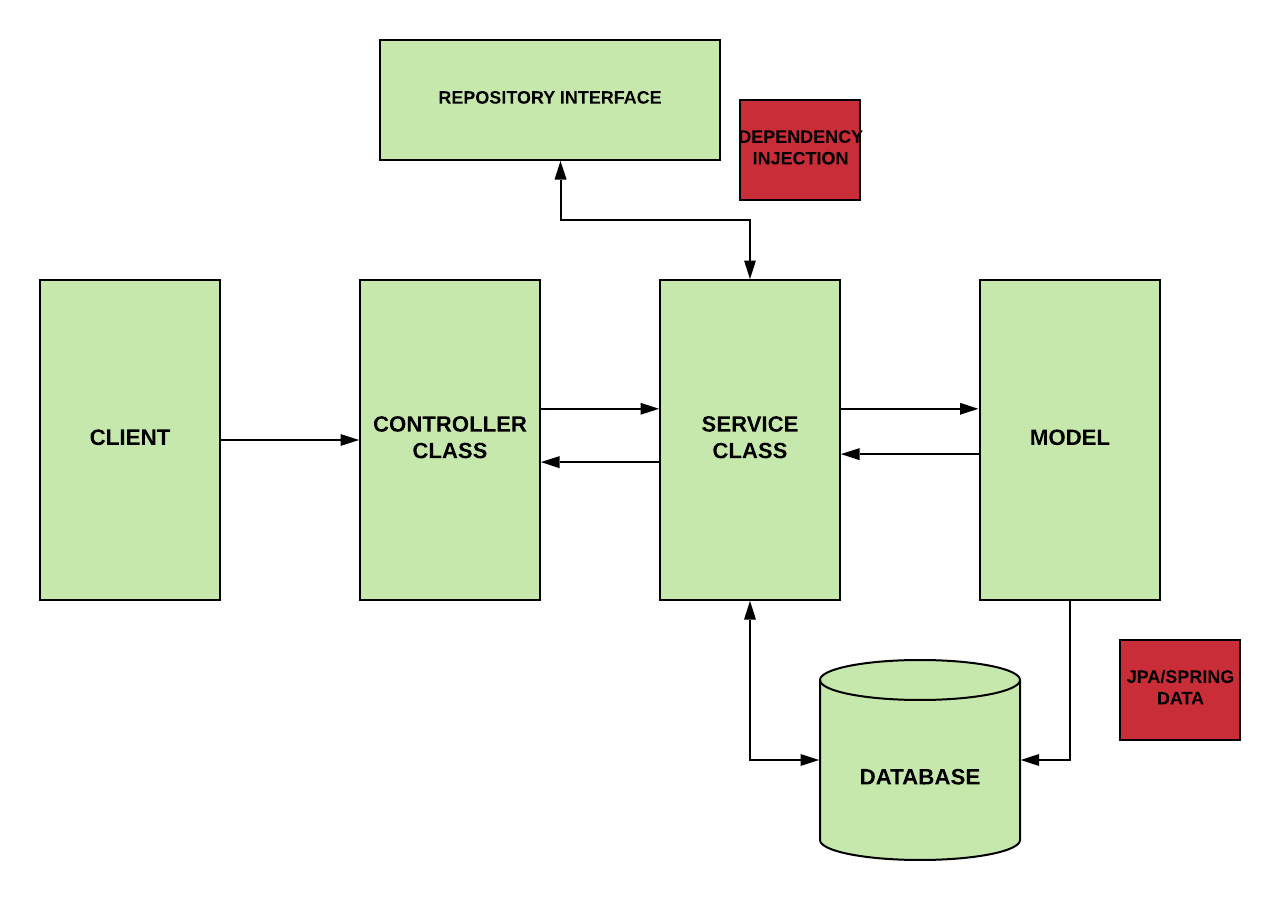
### Base de données :

#### Microsoft SQL Server

Microsoft SQL Server est un système de gestion de base de données (SGBD) en langage SQL incorporant entre autres un SGBDR (SGBD relationnel ») développé et commercialisé par la société Microsoft. Il fonctionne sous les OS Windows et Linux (depuis mars 2016), mais il est possible de le lancer sur macOS via Docker, car il en existe une version en téléchargement sur le site de Microsoft.

## Architecture de l’application

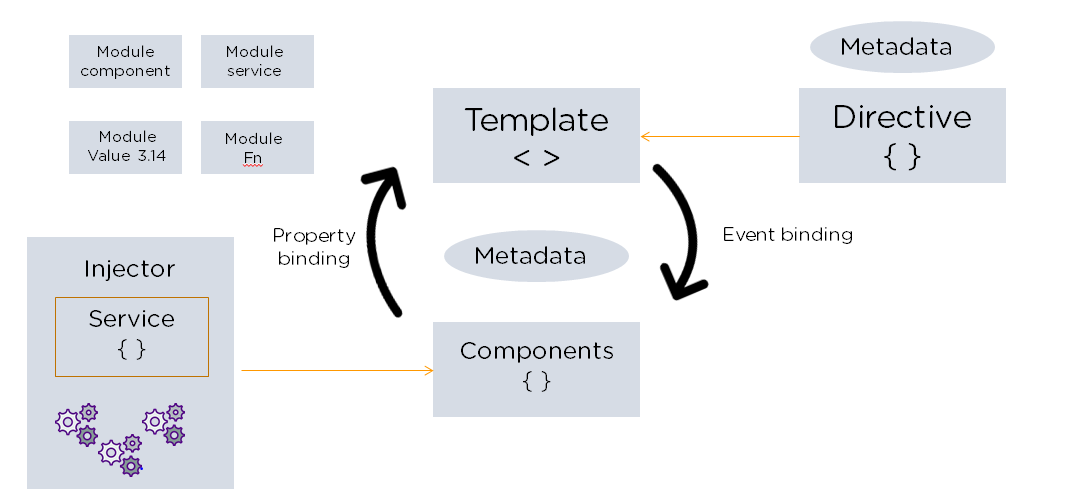
### Architecture BO



* **Couche de Présentation (Controller)** : Cette couche gère les requêtes HTTP entrantes de Postman ou d'autres clients. Elle traduit les paramètres JSON en objets, authentifie les requêtes et les transmet à la couche Service.
* **Couche Service** : Cette couche gère toute la logique métier. Elle utilise les services fournis par la couche Repository pour accéder aux données. Elle effectue également la validation des données et peut implémenter certaines règles métier.
* **Couche de Persistance (Repository)** : Cette couche contient toute la logique de stockage. Elle traduit les objets métier en entrées de base de données et vice versa. Elle effectue les opérations CRUD (Create, Read, Update, Delete) sur la base de données.
* **Couche de Base de Données** : C'est là que vos données sont stockées et récupérées. Les opérations CRUD sont effectuées ici.

En résumé, lorsqu'une requête HTTP est envoyée à votre API Spring Boot, elle est d'abord reçue par la couche Controller. Cette couche dirige la requête vers la méthode de service appropriée. Cette dernière traite les données en utilisant la couche Repository, qui interagit avec la base de données. Enfin, la réponse est renvoyée au client via la couche Controller.

#### Architecture FO



Angular est un cadre puissant qui permet de créer des interfaces utilisateur dynamiques et intuitives. Il utilise une architecture basée sur des composants, ce qui favorise la réutilisabilité, la facilité de maintenance et l'évolutivité.

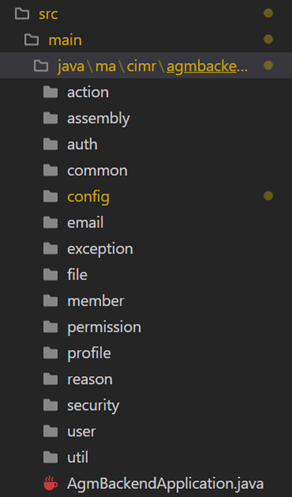
* **Composants (Components)** : Un composant est une unité indépendante qui contrôle une partie de l'interface utilisateur. Chaque composant se compose de trois éléments principaux :
* **Modèle (Template)** : Il s'agit du squelette HTML du composant qui définit comment ildoit être rendu dans l'interface utilisateur.
* **Classe (Component)** : Il s'agit d'une classe TypeScript qui contient les données et la logique du composant. Elle définit comment le composant doit se comporter.
* **Feuille de style (CSS)** : Il s'agit du fichier CSS qui contient le style appliqué au composant.
* **Services** : Les services sont des classes qui fournissent des méthodes spécifiques pour effectuer des tâches telles que la communication avec une API backend. Ils sont injectés dans les composants qui en ont besoin, permettant ainsi une séparation claire entre la logique de l'interface utilisateur et la logique métier.
* **Modules** : Les modules sont des conteneurs qui regroupent des composants, des services et d'autres éléments en blocs de fonctionnalité cohérents.
* **Routage (Routing)** : Le routage permet de naviguer entre les différentes parties de l'application. Il est généralement configuré dans un module de routage séparé.

En résumé, lorsqu'un utilisateur interagit avec l'application, une action est déclenchée. Cette action est gérée par le composant correspondant, qui peut utiliser un service pour communiquer avec une API backend si nécessaire. Les données récupérées sont ensuite traitées et affichées à l'utilisateur.

## Structure de l’application

### Arborescence du BO

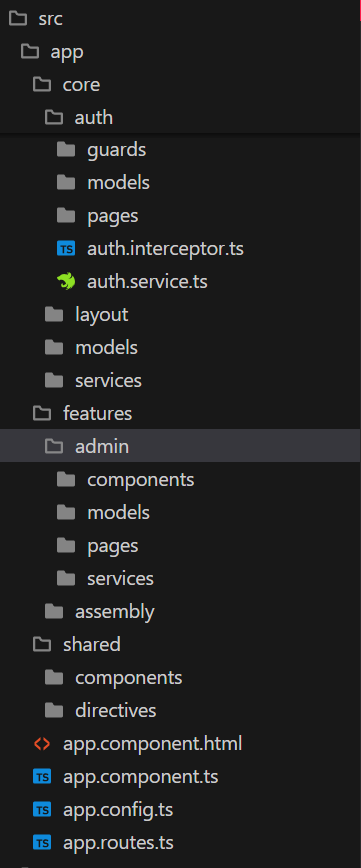
L'architecture du projet Spring est organisée par fonctionnalités, ce qui permet une séparation claire des responsabilités et facilite la gestion du code. Chaque fonctionnalité, comme member, assembly, ou auth, dispose de son propre dossier, où sont centralisés les services, contrôleurs, et autres composants liés. Des dossiers tels que config, security, et exception gèrent respectivement les configurations globales, la sécurité, et la gestion des exceptions. Cette organisation modulaire améliore la maintenabilité et l'évolutivité du projet.



### Arborescence du FO

L'application Angular est organisée de manière modulaire, ce qui permet de gérer efficacement la complexité croissante du code au fur et à mesure que de nouvelles fonctionnalités sont ajoutées. Cette structure modulaire facilite également la maintenabilité et l'évolutivité de l'application, en isolant les différentes responsabilités dans des dossiers spécifiques. Cette approche garantit que chaque partie de l'application peut être développée, testée, et mise à jour indépendamment des autres, tout en assurant une intégration harmonieuse de l'ensemble des fonctionnalités.

* **Core** : Ce dossier contient les fonctionnalités non spécifiques au domaine métier, telles que l'authentification et la mise en page. On y trouve des services, modèles, et gardes qui sont utilisés à travers l'application. Par exemple, le dossier auth comprend les pages de connexion et de récupération de mot de passe, ainsi que des services comme pour gérer l'authentification des utilisateurs. Ce dossier inclut également les éléments partagés au niveau global, tels que les gardes pour sécuriser les routes (auth.guard.ts), les modèles (auth.model.ts), et les services nécessaires pour les fonctionnalités transverses.
* **Features** : Ce dossier regroupe les fonctionnalités spécifiques au domaine métier de l'application. Chaque fonctionnalité est encapsulée dans son propre dossier avec des sous-dossiers pour les pages, services, composants, et autres éléments liés. Par exemple, une fonctionnalité de gestion des utilisateurs ou des produits peut avoir son propre sous-dossier avec des composants Angular dédiés, des services pour la logique métier, et des modèles pour gérer les données. Cette structure permet une gestion claire et isolée de chaque fonctionnalité, facilitant ainsi les mises à jour et l'ajout de nouvelles fonctionnalités.
* **Shared** : Le dossier shared contient le code partagé entre les différentes fonctionnalités, comme les composants réutilisables, les pipes pour la transformation des données, et les utilitaires génériques. Par exemple, un composant de notification ou une pipe de formatage de date pourrait être placé ici, étant utilisé dans plusieurs fonctionnalités de l'application. Ce dossier est essentiel pour éviter la duplication de code et garantir une maintenance plus facile, car il centralise les éléments communs à plusieurs parties de l'application.



# Chapitre 6 : Mise en œuvre de l’application

## Introduction

Ce chapitre se concentre sur la mise en œuvre de l'application de gestion des assemblées générales. À travers des captures d'écran, les principales fonctionnalités de l'application sont illustrées, montrant comment elles ont été intégrées pour répondre aux besoins spécifiques identifiés précédemment. L'objectif est de fournir un aperçu visuel de l'interface utilisateur, de la navigation, et de l'expérience globale offerte par l'application.

# Liste des abréviations

|  |  |
| --- | --- |
| **Abréviation** | **Désignation** |
| CIMR | Caisse interprofessionnelle marocaine de retraite |
| ISO | International Organization for Standardization |
| PDG | Président-directeur général |
| FO | Front office |
| BO | Back Office |
| CDC | Cahier des charges |
| CF | Cash-flow |
| PQP | Plan qualité de projet |
| MOA | Maître d'ouvrage |
| PO | Project Owner |
| FDR | Feuille de route |
| FP | Feuille de présence |
| PJ | Pièce jointe |
| DTR | Dernier trimestre réglé |
| DBR | Dernier bordereau reçu |
| DPO | Département organisation et planification |
| DAMG | Direction administrative des marchés et gestion |
| UML | Unified Modeling Language |
| POO | Programmation orientée objet |
| OOD | Oriented object detection |
| OMT | Object modeling technique |
| OOSE | Object Oriented Software Engineering |
| MOF | Meta-Object Facility |
| XMI | XML Metadata Interchange |