Table of Contents

[I. Étape 1 : Cahier de charge 2](#_Toc183008339)

[II. Étape 2 : Planification du projet et Organisation 2](#_Toc183008340)

[A. Équipe projet 2](#_Toc183008341)

[1. L’équipe est constituée de : 2](#_Toc183008342)

[2. Compétences techniques requises : 2](#_Toc183008343)

[3. Attribution et répartition de rôles 2](#_Toc183008344)

[B. Méthodologie de développement 3](#_Toc183008345)

[C. Élaboration d'un plan de projet détaillé 3](#_Toc183008346)

[III. Étape 3 : Conception du Système 5](#_Toc183008347)

[A. Architecture du système 5](#_Toc183008348)

[1. Design de l'architecture globale 5](#_Toc183008349)

[B. Spécifications techniques 6](#_Toc183008350)

[1. Diagramme de Cas d'Utilisation 6](#_Toc183008351)

[2. Diagramme de Classes 7](#_Toc183008352)

[3. Diagramme de Séquence 8](#_Toc183008353)

[4. Diagramme d'Activité 8](#_Toc183008354)

[5. Modélisation de la Base de Données 8](#_Toc183008355)

# Étape 1 : Cahier de charge

# Étape 2 : Planification du projet et Organisation

## Équipe projet

### L’équipe est constituée de :

* **Kangni Komlan Antoine SODO** 
  + Téléphone : +228 91 93 94 13
  + Email: [kangni.komlan.sodo@etud.u-picardie.fr](mailto:kangni.komlan.sodo@etud.u-picardie.fr)
* **Aristide Sthephane OUEDRAOGO**
  + Téléphone : +974 77358387
  + Email: [aristide.ouedraogo@etud.u-picardie.fr](mailto:aristide.ouedraogo@etud.u-picardie.fr)

### Compétences techniques requises :

* Développement Backend
* Développement Frontend
* Design
* Test
* Administration Système
* Gestion de projet

### Attribution et répartition de rôles

#### Détails des rôles

* Développement Backend :
  + Conception et développement de l'architecture serveur.
  + Gestion de la base de données.
  + Implémentation des API et de la logique métier (au besoin)
* Développement Frontend :
  + Conception de l'interface utilisateur.
  + Développement des composants frontend.
  + Assurer la réactivité et l'ergonomie du site.
* Design :
  + Création des maquettes et wireframes.
  + Tests d'ergonomie et d'expérience utilisateur.
* Test et Validation :
  + Conduire les tests unitaires et fonctionnels.
  + Effectuer les tests d'intégration.
* Administration Système :
  + Configuration des serveurs, déploiement.
  + Monitioring, maintenance.
* Gestion de Projet :
  + Planification, suivi des tâches, coordination.
  + Communication avec les parties prenantes, gestion des risques…

#### Attribution des rôles

## Méthodologie de développement

**Méthodologie adoptée : Scrum (Agile)**

* **Adaptabilité :** Scrum permet de s'adapter aux changements et aux imprévus, ce qui est idéal pour des novices.
* **Apprentissage continu :** Il favorise l'amélioration constante et l'intégration des retours.
* **Engagement :** Les réunions régulières maintiennent la motivation et la communication.

**Mise en place de Scrum :**

## Élaboration d'un plan de projet détaillé

**Définition des jalons et des livrables**

**(à mettre à jour selon le calendrier définit dans le cahier de charge)**

**Préparation**

* **Objectifs :**
  + Mise en place de l'environnement de développement.
  + Installation des outils nécessaires.
  + Lecture approfondie du cahier des charges.
* **Livrables** :
  + Plan de projet détaillé.
  + Liste des outils et technologies choisis.

**Analyse et Conception**

* **Objectifs :**
  + Analyse détaillée des besoins.
  + Conception de l'architecture système.
  + Modélisation de la base de données.
* **Livrables :**
  + Diagrammes UML.
  + Schéma de la base de données.

**Développement Backend - Partie 1**

* **Objectifs :**
  + Mise en place du framework backend.
  + Développement des fonctionnalités d'authentification et de gestion des utilisateurs.
* **Livrables :**
  + Modules d'authentification fonctionnels.
  + Documentation du code.

**Développement Frontend - Partie 1**

* **Objectifs :**
  + Mise en place du framework frontend.
  + Développement de la page de connexion et du tableau de bord.
* **Livrables :**
  + Interfaces utilisateur initiales.
  + Maquettes design.

**Gestion des Données de Prospection**

* **Objectifs :**
  + Développement des fonctionnalités de création, modification et suppression des données.
  + Intégration frontend-backend pour ces fonctionnalités.
* **Livrables :**
  + Module de gestion des contacts opérationnel.
  + Tests unitaires.

**Génération de Fiches Préremplies**

* **Objectifs :**
  + Implémentation de la génération de fiches préremplies.
  + Exportation des fiches en PDF.
* **Livrables :**
  + Fonctionnalité de génération de fiches.
  + Documentation utilisateur pour cette fonctionnalité.

**Historique des Modifications et Traçabilité**

* **Objectifs :**
  + Mise en place de l'historique des modifications.
  + Gestion des versions des fiches entreprises.
* **Livrables :**
  + Module d'historique fonctionnel.
  + Interface utilisateur pour visualiser l'historique.

**Sécurité et Droits d'Accès**

* **Objectifs :**
  + Implémentation du contrôle d'accès basé sur les rôles.
* **Livrables :**
  + Système de gestion des rôles et permissions.
  + Tests de sécurité.

**Tests Finaux et Documentation**

* **Objectifs :**
  + Tests de performance, charge, et sécurité.
  + Rédaction de la documentation technique et des guides utilisateurs.
* **Livrables :**
  + Documentation complète.

# Étape 3 : Conception du Système

## Architecture du système

### Design de l'architecture globale

Pour ce projet, nous adoptons une architecture en couches afin de séparer les responsabilités et faciliter la maintenance et l'évolution du système. Les trois couches principales sont :

1. **Couche de Présentation (Frontend)**
   * **Responsabilité :** Interface utilisateur avec laquelle les utilisateurs interagissent.
   * **Technologies :** HTML5, CSS3, JavaScript, framework front-end (React.js ou Angular). (**je n’ai aucune exepérience entre React.js ou Angular au cas tu en as une**)
   * **Fonctionnalités :**
     + Formulaires pour la saisie et la mise à jour des données.
     + Affichage des fiches entreprises et des listings.
     + Tableau de bord interactif.
     + Navigation intuitive et ergonomique.
2. **Couche Logique Métier (Backend)**
   * **Responsabilité :** Traitement des données, application des règles métier, gestion de la sécurité.
   * **Technologies :** PHP avec le framework Symfony.
   * **Fonctionnalités:**
     + Gestion des utilisateurs et des rôles.
     + Traitement des demandes du frontend.
     + Gestion de l'historique des modifications.
     + Génération des fiches préremplies et des listings personnalisés.
3. **Couche d'Accès aux Données (Base de Données)**
   * **Responsabilité :** Stockage et récupération des données.
   * **Technologies :** MySQL.
   * **Fonctionnalités :**
     + Stockage des informations de contact, historiques des interactions, droits des utilisateurs, etc.
     + Exécution des requêtes SQL optimisées.
     + Assurer l'intégrité et la sécurité des données.

**MVC (Modèle-Vue-Contrôleur)**

Le modèle MVC est choisi pour structurer l'application en séparant les données (Modèle), l'interface utilisateur (Vue), et la logique de contrôle (Contrôleur).

* **Modèle** : Représente les données et la logique métier. Il gère l'accès aux données, les règles de validation, et l'état de l'application.
* **Vue** : Gère l'affichage des données fournies par le modèle. Elle est responsable de la présentation et de l'interaction avec l'utilisateur.
* **Contrôleur** : Intercepte les actions de l'utilisateur (via la Vue), traite les événements, et interagit avec le Modèle pour mettre à jour les données ou modifier l'état de l'application.

## Spécifications techniques

**Diagrammes UML**

### Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation représente les interactions entre les acteurs (utilisateurs) et le système. Voici les principaux cas d'utilisation :

* **Gérer les fiches entreprises**
  + Créer une nouvelle fiche.
  + Consulter une fiche existante.
  + Modifier une fiche.
  + Supprimer une fiche.
* **Générer une fiche préremplie**
  + Sélectionner une entreprise.
  + Générer la fiche en format PDF.
* **Mettre à jour les fiches après visite**
  + Ajouter des informations recueillies.
  + Enregistrer l'historique des modifications.
* **Générer des listings personnalisés**
  + Définir les critères de recherche.
  + Générer et exporter le listing.
* **Gérer les utilisateurs et les droits d'accès**
  + Authentification.
  + Gestion des rôles et permissions.

**Acteurs :**

* **Service de prospection**
* **Responsable de formation**
* **Direction**
* **Service secrétariat**
* **Service d’orientation**
* **Service informatique**
* **Enseignants/Formateurs**

### Diagramme de Classes

Les principales classes identifiées sont :

* **Utilisateur**
  + Attributs : id, nom, prénom, email, motDePasse, rôle.
  + Méthodes : s'authentifier(), gérerProfil().
* **Entreprise**
  + Attributs : id, raisonSociale, coordonnées, secteurActivité, tailleEntreprise, informationsComplementaires.
  + Méthodes : créer(), modifier(), supprimer(), consulter().
* **FicheEntreprise**
  + Attributs : id, entrepriseId, dateVisite, commentaires, crééPar, modifiéPar, dateModification.
  + Méthodes : générerPDF(), mettreAJour().
* **HistoriqueModification**
  + Attributs : id, ficheEntrepriseId, utilisateurId, dateModification, modifications.
  + Méthodes : enregistrerModification(), consulterHistorique().
* **Listing**
  + Attributs : critèresRecherche, résultats.
  + Méthodes : générer(), exporterPDF(), exporterExcel().
* **Authentification**
  + Méthodes : seConnecter(), seDéconnecter(), gérerSessions().
* **DroitsAccès**
  + Attributs : rôle, permissions.
  + Méthodes : vérifierPermission().

### Diagramme de Séquence

### Diagramme d'Activité

### Modélisation de la Base de Données

**Schéma Entité-Association (E/A)**

**Principales entités :**

1. **Utilisateur**
   * id (PK)
   * nom
   * prénom
   * email
   * mot\_de\_passe
   * rôle
2. **Entreprise**
   * id (PK)
   * raison\_sociale
   * adresse
   * téléphone
   * email
   * secteur\_activité
   * taille\_entreprise
   * site\_web
3. **FicheEntreprise**
   * id (PK)
   * entreprise\_id (FK vers Entreprise)
   * date\_visite
   * commentaires
   * créé\_par (FK vers Utilisateur)
   * date\_création
   * modifié\_par (FK vers Utilisateur)
   * date\_modification
4. **HistoriqueModification**
   * id (PK)
   * fiche\_entreprise\_id (FK vers FicheEntreprise)
   * utilisateur\_id (FK vers Utilisateur)
   * date\_modification
   * détails\_modification
5. **Rôle**
   * id (PK)
   * nom\_rôle
   * description
6. **Permission**
   * id (PK)
   * nom\_permission
   * description
7. **Rôle\_Permission** (Table associative)
   * rôle\_id (FK vers Rôle)
   * permission\_id (FK vers Permission)
8. **Session**
   * id (PK)
   * utilisateur\_id (FK vers Utilisateur)
   * token\_session
   * date\_dernière\_activité

**Normalisation des Données**

* **Forme Normale 1 (1NF) :** Toutes les valeurs des attributs sont atomiques.
* **Forme Normale 2 (2NF) :** Toutes les colonnes dépendent entièrement de la clé primaire.
* **Forme Normale 3 (3NF) :** Aucune dépendance transitive.

**Exemple :**

* Les informations de l'entreprise sont stockées dans la table **Entreprise**, évitant ainsi la redondance des données dans la table **FicheEntreprise**.
* Les rôles et permissions sont séparés pour faciliter la gestion des droits d'accès.