

Cycle de Formation des Ingénieurs en Informatique

# Cahier de charge

## Elaboré Par

Moetez Aissaoui  
Mariem Neit Addi  
Nesrine Bradai  
Syrine Ghazghazi

## Table des matières

1. Introduction .....	2
2. Étude .....	2
2.1 Définition du besoin .....	2
2.2 Définition des attendus .....	3
3. Organisation.....	4
3.1 L'équipe.....	4
3.2 Définition des tâches .....	4
4. Conception.....	7
4.1 Modélisation UML.....	7
4.2 Architecture physique de l'application .....	9
4.3 Développement des fonctionnalités.....	10
4.4 Écriture de la documentation technique .....	11
4.5 Tests .....	11
5. Livraison .....	12
5.1 Déploiement de l'application.....	12
6. Conclusion .....	13

## 1. Introduction

Dans le cadre de notre projet de deuxième semestre en Génie Logiciel et système d'information, il nous a été demandé de concevoir et développer une application web dans les langages et technologies étudiées, en équipe de quatre personnes .

Ce dossier a pour objectif de présenter et justifier de :

- L'étude préalable et le cadrage du projet → définition des attendus client.
- L'organisation → planification de l'utilisation des ressources.
- La conception → modélisation graphique et fonctionnelle de l'application.
- La réalisation → développement des fonctionnalités et rédaction des documentations.
- La livraison → mise à disposition de l'application.

## 2. Étude

### 2.1 Définition du besoin

Nous développerons une application qui permettra non seulement de fiabiliser l'acquisition des temps de présence, mais également d'automatiser le circuit de vérification de chaque participant selon le cycle RH.

Ce module vise à :

A. Faciliter la gestion des congés et de la planification du travail en permettant à l'employeur de :

- Soumettre les demandes de congé de manière rapide et efficace.
- Enregistrer les heures de travail des employés.
- Élaborer et modifier les plans de travail selon les besoins opérationnels.

B. Optimiser le processus d'approbation des congés en :

- Assurant la traçabilité et la gestion des demandes de congé.
- Permettant à l'employeur d'approuver les demandes de congé de manière efficace et transparente.

➔ Ces fonctionnalités reposent sur le développement d'une application RH dédiée, visant à simplifier la gestion des congés, la planification du travail, le suivi du temps de travail et l'approbation des demandes de congé par l'employeur.

**Il existe en outre trois rôles différents permettant des actions distinctes :**

Employé :

- S'authentifier pour accéder au système.
- Consulter ses congés, son planning de travail et son historique de pointage.
- Enregistrer son pointage.

Manager :

- Effectuer les mêmes actions que les employés.
- Gérer les demandes de congé de son équipe.
- Consulter l'historique des congés et du pointage de son équipe.
- Gérer les compétences de son équipe.

## Gestionnaire RH :

- S'authentifier pour accéder au système.
- Gérer le catalogue des employés.
- Gérer les budgets de congé et de pointage.
- Valider les demandes de congé une fois traitées.

➔ Le processus métier se déroule comme suit :

1. Le gestionnaire RH alloue un budget de congé et de pointage.
2. L'employé consulte son planning de travail, enregistre son pointage et peut soumettre une demande de congé.
3. La demande de congé est transférée à son manager :
  - Si le manager approuve la demande, l'employé est notifié et peut prendre son congé.
  - Si la demande est refusée, l'employé reçoit une notification expliquant le motif du refus.
4. Chaque utilisateur doit se connecter au système pour effectuer des actions relatives aux congés et au pointage.

## 2.2 Définition des attendus

En plus des fonctionnalités citées ci-dessus, des exigences clients sont à prendre en compte.

### Exigences technologiques

Outils de développement	
GitHub	Service web d'hébergement et de gestion de développement logiciel utilisant l'outil de gestion de versions Git.
React JS	Framework front-office.
WebService REST	Style d'architecture logicielle permettant l'interopérabilité entre ordinateurs sur internet.
JPA	Permet la persistance des données.
Spring Boot	Micro-framework permettant de générer l'architecture d'une application Java.

### Exigences de qualité

► Ergonomie ➔ l'ergonomie représente la disposition et l'accessibilité de chacune des fonctionnalités listées ci-dessus. Afin de répondre du mieux possible aux besoins de nos employeurs, la navigation doit être la plus simple et intuitive possible. Nous allons donc présenter par la suite des maquettes de l'interface générale de notre application, point de départ pour le développement global de l'application.

- **Fiabilité** → la fiabilité d'une application se caractérise par sa capacité à fonctionner comme prévu, sans bug ou ralentissement altérant l'expérience utilisateur. C'est au travers d'un développement méthodologique et testé à chaque phase que nous évaluerons la fiabilité. Enfin, lors de la mise en production nous analyserons et comblerons les dernières anomalies éventuellement détectées.
- **Maintenabilité** → il s'agit de toujours penser aux possibles améliorations qui pourraient être implémentées par la suite, que ce soit des nouvelles fonctionnalités ou une modification d'interface. Cela implique donc une culture de codage où chaque fonctionnalité peut fonctionner indépendamment l'une de l'autre.

### 3. Organisation

#### 3.1 L'équipe

L'équipe est constituée de quatre personnes issues de cursus scolaires similaires mais de métiers différents. Cette mixité des profils admet non seulement un échange des connaissances acquises dans les domaines de chacun, mais aussi un partage des points de vue et des expériences personnelles sur les sujets appréhendés.

#### 3.2 Définition des tâches

Les phases du projet ont été retranscrites dans ce tableau :

Phase	Explication	Nb jours / homme
Définition du projet	Découverte et réflexion sur le projet.	1
Pilotage	Action menées tout au long du projet afin de le cadrer.	1
Modélisation	Les interactions entre les acteurs et le système seront modélisées via le langage UML.	4
Préparation des environnements	Afin de maximiser la productivité de chacun des membres de l'équipes les environnements seront mis en place très tôt dans le projet.	0,5
Réalisation	Développement front et back.	20
Tests	Réalisation des tests pour notre application.	2
Documentation	Réalisation de la documentation de notre application.	0,5

Déploiement	Déploiement de notre application sur git.	1
Total		30

Selon nos estimations, ce projet nécessite 30 jours homme. Possédant 20 jours au maximum, nous pouvons penser que le projet ne sera pas entièrement finalisé lors de sa livraison. De ce fait, nous allons approfondir le déroulement et le contenu de chacune de ses phases.

## Planification des tâches

### Définition du projet (1 jour / homme)

Tâches	Priorité	Ressources
Prise de connaissance du projet	/	Equipe
Définition du paramètre	/	Equipe
Analyse métier	/	Equipe
Analyse fonctionnelle	/	Equipe

### Pilotage (1 jour / homme)

Tâches	Priorité	Ressources
Définition des tâches	/	Equipe
Estimation du temps par tâches	/	Equipe
Répartition des tâches	/	Equipe
Suivi	/	Equipe

### Modélisation (4 jours / homme)

Tâches	Priorité	Ressources
Diagrammes cas d'utilisation	/	Equipe
Diagramme de classes	/	Equipe
Diagramme de séquence	/	Equipe

Préparation environnement (0,5 jour / homme)

Tâches	Priorité	Ressources
Préparation environnement	/	Equipe

Documentation (0,5 jour / homme)

Tâches	Priorité	Ressources
JavaDoc	/	Moetez
ReadMe	/	Nesrine

Réalisation (20 jours / homme)

Tâches	Priorité	Ressources
Front		
Page Login	/	Moetez
Dashboard	/	Syrine
Page Demande de congé	/	Nesrine
Page Historique Pointage	/	Mariem
Page Historique Congé	/	Nesrine
Page Gestion des Tickets	/	Mariem
Page Pointage	/	Mariem
Page validation des congés	/	syrine
Page Liste des employeurs	/	Nesrine
Back		
S'authentifier	1	Moetez
Gérer taches	1	Mariem
Valider congés	4	Nesrine
Gérer Congés	4	Nesrine
Gérer pointage	4	Moetez
Voir liste employeurs	1	Nesrine
Demander Congés	2	Syrine

Tâches	Priorité	Ressources
Voir historique Pointage	3	Syrine
Gérer employé	2	Moetez
/	/	Equipe

Test (2 jour / homme)

Tâches	Priorité	Ressources
Test unitaire	/	Equipe
Test utilisateur	/	Equipe

Déploiement (1 jour / homme)

Tâches	Priorité	Ressources
Déploiement	/	Moetez

## 4. Conception

### 4.1 Modélisation UML

Le langage UML (*Unified Modeling Language*) est un standard de modélisation graphique. Il permet la visualisation de la conception d'un système. Nous avons utilisé ce langage afin modéliser notre application et produire les diagrammes suivants :

- Diagramme de cas d'utilisation
- Diagramme de classe



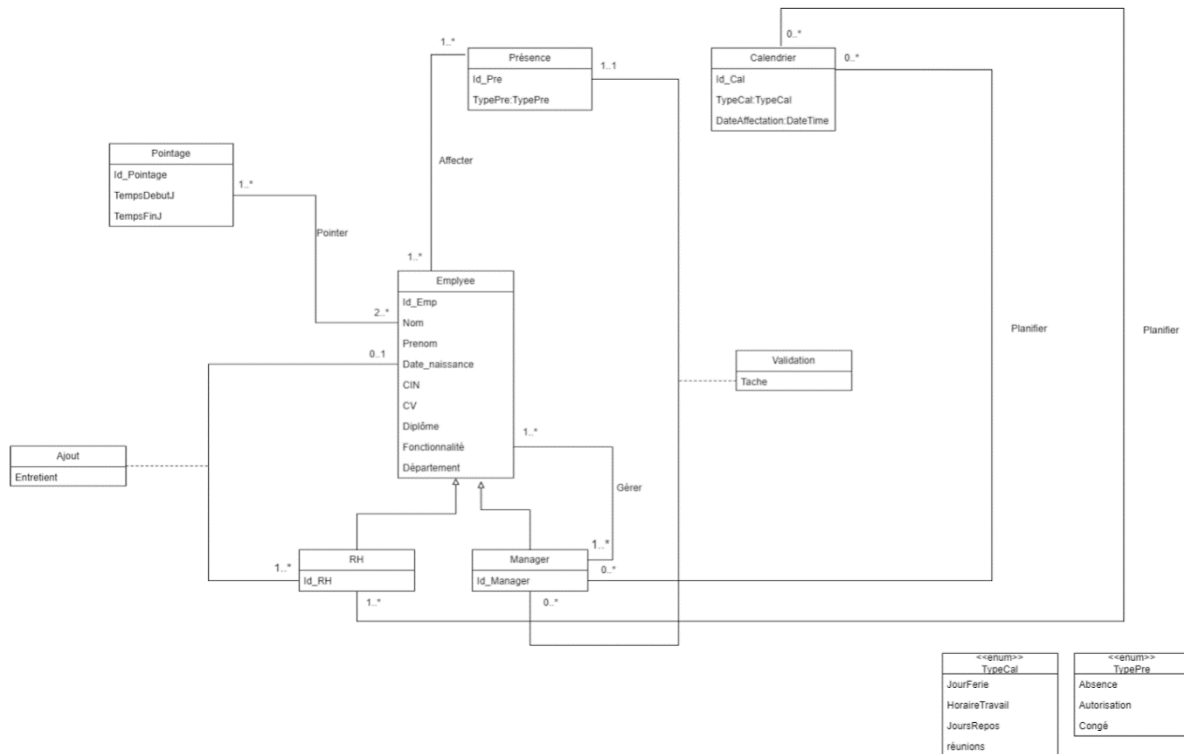
### Diagramme de cas d'utilisation

Ce diagramme permet de modéliser les principales interactions entre les utilisateurs et le système.



### Diagramme de classe

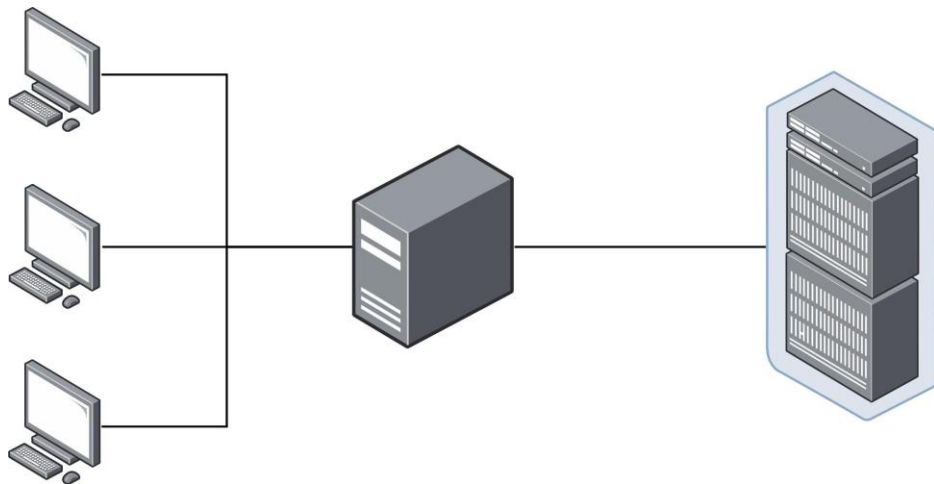
Le diagramme de classes est un schéma utilisé pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations qu'elles entretiennent.



## 4.2 Architecture physique de l'application

L'architecture définie est connue sous le nom de « trois-tiers ». Elle est constituée de trois couches communiquant entre elles : la couche client, la couche serveur d'application et la couche serveur de base de données et est reconnue dans le monde professionnel par sa bonne fiabilité, sécurité, disponibilité et évolutivité.

Pour l'hébergement de notre base de donnée, nous avons choisi la solution gratuite proposée par springBoot qui répond entièrement aux besoins de l'application.



*Architecture 3 tiers.*

*De gauche à droite : postes client, serveur d'application et base de données.*

L'application en elle-même n'est déployée que sur la partie serveur : le client n'a besoin que d'un navigateur web compatible. Cette facilité de déploiement aura pour conséquence non seulement de réduire le coût de déploiement mais aussi de permettre une évolution régulière du système car elle nécessitera que la mise à jour de l'application sur le serveur applicatif.

Réalisation

### **4.3 Développement des fonctionnalités**

#### Architecture logicielle

Pour répondre aux besoins de l'architecture trois tiers, nous utiliserons le modèle MVC ;  
modèle – vue – contrôleur.

- ▶ **Modèle** : contient les données et la logique de traitement (validation, lecture et enregistrement) via les classes métier. Il vient suppléer le contrôleur et est complété par le DAO (ou objet d'accès aux données) qui est un patron de conception gérant l'accès aux données dans notre projet. Il correspond ici à la partie domaine.
- ▶ **Vue** : interface graphique qui sert à la présentation et la modification des données du modèle.
- ▶ **Contrôleur** : gère la partie logique de l'interaction homme machine.

## 4.4 Écriture de la documentation technique

### JavaDoc

La JavaDoc, outil développé par Oracle donne la possibilité de créer une documentation d'API au format HTML. Cette documentation se réalise à partir des commentaires présents dans le code source de l'application, qui sont écrits tout au long du développement.

### ReadMe

Le fichier ReadMe est un fichier dans lequel apparaît des informations relatives aux autres fichiers constitutif du projet.

## 4.5 Tests

Tests réalisés	
Tests unitaires.	À l'apparition d'une nouvelle fonctionnalité, celle-ci est testé afin de voir si l'action réalisée est conforme à l'attendu.
Tests de bout-en-bout.	Consiste à tester un process précis. Par exemple la demande et la validation de la demande de formation: faire la demande de congé par l'employé, gestion de cette demande par le manager, réponse visible par l'employé.

## 5. Livraison


### 5.1 Déploiement de l'application

L'un des prérequis projet initiaux concernait la mise à disposition de l'application sur une adresse internet accessible depuis l'extérieur. Il a donc fallu sélectionner un hébergeur pour la base de données et l'application.

#### Application web


Nous avons utilisé l'offre « Elastic Beanstalk » d'Amazon Web Services, un des leaders du domaine du cloud computing à la demande, qui permet de déployer des applications et services web développés avec différents langages populaires comme Java, .NET, Node.js...

Il s'agit de venir déposer un fichier .zip ou .war du projet, le déploiement et la mise en ligne étant totalement automatique et optimisée.



**Santé**  
**OK**  
[Causes](#)

**Version en cours d'exécution**  
assist-rh-source-2  
[Charger et déployer](#)

  
**Configuration**  
Tomcat 8.5 with Java 8 running  
on 64bit Amazon Linux/3.1.6  
[Modification](#)

**Événements récents**  
[Afficher tout](#)

Horaires	Type	Détails
05-07-2019 11:47:28 UTC+0200	INFO	Environment health has transitioned from Info to Ok. Application update completed 32 seconds ago and took 18 seconds.
05-07-2019 11:46:28 UTC+0200	INFO	Environment health has transitioned from Ok to Info. Application update in progress on 1 instance. 0 out of 1 instance completed (running for 10 seconds).
05-07-2019 11:46:04 UTC+0200	INFO	Environment update completed successfully.
05-07-2019 11:46:04 UTC+0200	INFO	New application version was deployed to running EC2 instances.
05-07-2019 11:45:46 UTC+0200	INFO	Deploying new version to instance(s).

## 6. Conclusion

En conclusion, le système développé offre une solution complète pour la gestion des ressources humaines, en mettant à disposition des employés, des managers et des gestionnaires RH des fonctionnalités adaptées à leurs besoins spécifiques.

Les employés peuvent facilement consulter leur planning de travail, enregistrer leur pointage et gérer leurs demandes de congé, tandis que les managers ont accès à des outils de gestion des congés et du pointage de leur équipe. De plus, les gestionnaires RH bénéficient d'un ensemble d'outils pour gérer efficacement les budgets de congé et de pointage, ainsi que pour valider les demandes de congé.

Ce système permet une gestion fluide et transparente des ressources humaines, facilitant ainsi le suivi des activités et la prise de décisions éclairées au sein de l'organisation. En centralisant toutes ces fonctionnalités au sein d'une seule plateforme, il favorise également la cohérence et la collaboration entre les différents acteurs impliqués dans la gestion du personnel.