

# **Système Intelligent de Gestion des Stagiaires**

## **Centre National de l'Informatique (CNI) – Tunisie**

### **1. Introduction générale**

Dans le cadre de la transformation numérique de l'administration publique tunisienne, le **Centre National de l'Informatique (CNI)** joue un rôle stratégique dans la modernisation des systèmes d'information et l'optimisation des processus administratifs.

La gestion des stages constitue un processus essentiel pour le CNI, tant pour la formation des étudiants que pour la valorisation des compétences nationales. Cependant, ce processus demeure en grande partie **manuel**, ce qui engendre des lourdeurs administratives, un suivi limité et une exploitation insuffisante des données liées aux stagiaires.

Ce projet vise la **conception et le développement d'un Système Intelligent de Gestion des Stagiaires**, permettant de digitaliser l'ensemble du cycle de vie du stage, depuis la soumission de la demande jusqu'à l'évaluation finale, tout en intégrant des mécanismes d'automatisation et d'aide à la décision.

### **2. Contexte et problématique**

#### **2.1 Contexte**

Chaque année, le CNI accueille un nombre important de stagiaires issus de diverses spécialités telles que :

- Informatique
- Réseaux
- Développement logiciel
- Systèmes d'information

Le processus actuel de gestion des stages repose principalement sur :

- Des demandes effectuées par courrier électronique ou sur support papier
- Une affectation manuelle des stagiaires aux encadrants
- Un suivi non structuré des activités réalisées
- Une génération manuelle des attestations et documents administratifs

Cette approche limite l'efficacité globale du processus et augmente le risque d'erreurs.

## 2.2 Problématique

Les principales difficultés identifiées sont les suivantes :

- Manque de traçabilité des demandes de stage
- Affectation non optimisée des stagiaires aux encadrants
- Absence d'indicateurs de performance et de suivi
- Perte de temps administratif significative
- Difficulté à valoriser les compétences acquises par les stagiaires

👉 **Problématique centrale :**

Comment mettre en place un système numérique intelligent permettant une gestion efficace, équitable, automatisée et traçable des stages au sein du CNI ?

## 3. Objectifs du système

### 3.1 Objectif général

Développer une **plateforme web intelligente** permettant la gestion complète et centralisée des stages au sein du Centre National de l'Informatique.

### 3.2 Objectifs spécifiques

- Digitaliser le processus de dépôt des demandes de stage
- Automatiser l'affectation des stagiaires selon des critères définis
- Assurer un suivi structuré et continu des stages

- Mettre en place une évaluation objective et normalisée des stagiaires
- Générer automatiquement les documents officiels (attestations, conventions, etc.)
- Constituer une base de données de talents exploitable par le CNI

## 4. Analyse fonctionnelle

### 4.1 Acteurs du système

#### ◇ *Stagiaire*

Étudiant ou diplômé souhaitant effectuer un stage au sein du CNI.

#### ◇ *Administrateur*

Responsable administratif chargé de la gestion, du traitement et de la validation des demandes.

#### ◇ *Encadrant*

Responsable du suivi pédagogique et de l'évaluation du stagiaire.

### 4.2 Besoins fonctionnels

#### *Gestion des comptes*

- Création de comptes utilisateurs
- Authentification (connexion / déconnexion)
- Réinitialisation du mot de passe
- Activation et désactivation des comptes
- Gestion des rôles et des autorisations

#### *Fonctionnalités du stagiaire*

- Créer et soumettre une demande de stage

- Choisir le type de stage (Initiation, Perfectionnement, PFE, etc.)
- Renseigner les informations personnelles
- Saisir les informations académiques
- Télécharger les documents requis (CV, lettre de motivation, convention...)
- Consulter l'état d'avancement de la demande
- Recevoir des notifications automatiques
- Télécharger les documents validés
- Consulter le résultat de l'évaluation finale

### **Fonctionnalités de l'administrateur**

- Consulter la liste des demandes de stage
- Filtrer les demandes (par date, état, type...)
- Accéder aux détails d'une demande
- Accepter ou refuser une demande
- Affecter un encadrant à un stagiaire
- Modifier l'état d'une demande
- Gérer les documents administratifs
- Gérer les utilisateurs du système
- Générer des rapports et statistiques

### **Fonctionnalités de l'encadrant**

- Consulter la liste des stagiaires affectés
- Accéder au dossier complet d'un stagiaire
- Suivre la période de stage
- Ajouter des observations et commentaires
- Réaliser l'évaluation finale
- Valider l'évaluation

## **5. Besoins non fonctionnels**

Les besoins non fonctionnels définissent les **contraintes de qualité, de performance, de sécurité et d'exploitation** du système.

## 5.1 Performance

- Le système doit être capable de gérer un nombre élevé d'utilisateurs simultanés
- Le temps de réponse des requêtes principales ne doit pas dépasser **2 secondes**
- Les opérations critiques (soumission de demande, validation, évaluation) doivent être optimisées

## 5.2 Sécurité

- Les données sensibles doivent être protégées par des mécanismes d'authentification sécurisés (JWT)
- Les mots de passe doivent être stockés de manière chiffrée (hashing)
- Le système doit respecter la gestion des rôles et des autorisations
- L'accès aux fonctionnalités doit être restreint selon le profil utilisateur

## 5.3 Fiabilité et disponibilité

- Le système doit être disponible **24h/24 et 7j/7**
- Les données doivent être sauvegardées régulièrement
- En cas d'erreur, le système doit afficher des messages clairs et explicites

## 5.4 Ergonomie et utilisabilité

- L'interface utilisateur doit être intuitive et facile à prendre en main
- Le système doit être accessible aux utilisateurs non techniques
- La navigation entre les différentes fonctionnalités doit être fluide

## 5.5 Compatibilité et portabilité

- L'application doit être compatible avec les principaux navigateurs web
- L'interface doit être responsive (PC, tablette, mobile)

- Le système doit être déployable sur différents environnements (local, serveur, cloud)

## 5.6 Maintenabilité et évolutivité

- L'architecture doit être modulaire et facilement extensible
- Le code doit être structuré, documenté et maintenable
- Le système doit permettre l'ajout futur de nouvelles fonctionnalités (IA, statistiques avancées, matching automatique, etc.)

## 6. Cas d'utilisation (Use Cases)

### UC1 – Déposer une demande de stage

- **Acteur** : Stagiaire
- **Description** : Le stagiaire remplit le formulaire et soumet la demande
- **Résultat** : La demande est enregistrée avec l'état *En attente*

### UC2 – Traiter une demande

- **Acteur** : Administrateur
- **Description** : Étude et validation du dossier
- **Résultat** : Demande *Acceptée* ou *Refusée*

### UC3 – Affecter un encadrant

- **Acteur** : Administrateur
- **Résultat** : Le stagiaire est associé à un encadrant

### UC4 – Évaluer un stagiaire

- **Acteur** : Encadrant
- **Résultat** : Évaluation finale enregistrée et validée

## 7. Règles de gestion

- Un stagiaire ne peut soumettre qu'une seule demande active à la fois
- Une demande ne peut être acceptée sans documents complets
- L'évaluation finale ne peut être effectuée qu'après la fin du stage
- Toute modification de l'état d'une demande génère une notification
- Une demande acceptée ne peut plus être supprimée

## 8. États d'une demande de stage

1. Brouillon
2. Soumise
3. En attente
4. Acceptée
5. Refusée
6. En cours de stage
7. Terminée
8. Évaluée

## 9. Données manipulées

### Utilisateur

- Identifiant
- Nom
- Email
- Rôle
- Statut

### Demande de stage

- Identifiant
- Type de stage
- Date de début et de fin
- État
- Stagiaire associé

- Encadrant associé

### **Document**

- Type
- Fichier
- Date de dépôt

### **Évaluation**

- Note
- Appréciation
- Date

## **10. Notifications (transversales)**

- Création d'une demande de stage
- Décision administrative
- Affectation d'un encadrant
- Validation de l'évaluation finale
- Mise à disposition des documents

## **11. Contraintes fonctionnelles**

- Interface simple, ergonomique et intuitive
- Sécurité et confidentialité des données
- Gestion multi-rôles
- Historique des actions
- Application responsive (PC, tablette, mobile)



## 12. Priorisation des fonctionnalités

### ● Fonctionnalités obligatoires

- Authentification
- Gestion des demandes
- Traitement administratif
- Évaluation finale

### ● Fonctionnalités importantes

- Notifications
- Filtres et recherche
- Historique

### ● Fonctionnalités optionnelles

- Tableaux de bord
- Statistiques avancées
- Export PDF / Excel

## Conclusion

Cette analyse fonctionnelle constitue la **base officielle du projet “Système Intelligent de Gestion des Stagiaires”**.

Elle servira de référence pour :

- La conception technique
- Le développement backend et frontend
- La gestion du projet (Trello, Jira)
- Les phases de test et de validation