**Rapport de stage**

**4ème année**

**Ingénierie Informatique et Réseaux**

**Sous le thème**

|  |
| --- |
| **Application de gestion bancaire** |

***Effectué à :***

|  |  |
| --- | --- |
| **ENTREPRISE :** |  |

|  |  |  |
| --- | --- | --- |
| Réalisé par :    **EMHAMED Yassine**  **MEHDAOUI Mouad** |  | Encadré à l’entreprise par :  **Mme. Khadija FIKRINE** |

**Dédicaces**

*Nous dédions ce modeste travail, comme preuve de respect, de gratitude, et de reconnaissance à : Nos chers parents : Qui n’ont jamais cessé de nous*

*encourager et nous conseiller. Ils nous ont beaucoup aidés tout en long de notre chemin, grâce à leur*

*amour, leur dénouement, leur compréhension et leur patience sans jamais nous quitter des yeux ni*

*Baisser les bras.*

*À nos professeurs Pour leurs efforts qu’ils n’ont cessé de fournir, Pour le savoir qu’ils ont partagé avec nous et pour leurs conseils précieux.*

*À nos amis qui ont été toujours avec nous lors du*

*Besoin.*

*À tous ceux qui ont contribué à ce travail*

*Merci à vous tous*

**Remerciements**

*Nous remercions ALLAH le tout puissant de nous avoir donné le courage, la volonté et la patience de mener à terme le présent travail.*

*Nous remercions I-ETERIA et l'administration locale de l'entreprise qui nous permettons d’effectuer notre stage.*

*Nous adressons aussi nos vifs remerciements aux membres des jurys Pour avoir bien voulu examiner et juger ce travail.*

*Nous ne laisserons pas cette occasion passer, sans remercier tous les enseignants et le personnel de EMSI RABAT, tout particulièrement ceux de la section informatique pour leur aide et leurs précieux conseils et pour l’intérêt qu’ils portent à notre formation. Enfin, nous adressons nos sincères remerciements à tous ceux qui ont contribué de près ou de loin au bon déroulement de ce projet*

**Résumé**

Notre travail consiste à développer une application web pour la gestion bancaire. Cette idée de projet a émergé après avoir remarqué que beaucoup de banques souffraient de beaucoup d'alertes

Grâce à notre application, nous pouvons prévisualiser les alertes qui nous parviennent des clients.

Le travail est présenté à travers trois chapitres :

Le premier sera consacré à la présentation des besoins fonctionnels et non fonctionnels.

Dans le second chapitre, nous nous intéresserons à l’étude conceptuelle et nous détaillerons les différents modèles adoptés ainsi la structure de la base de données.

Le dernier chapitre sera consacré à la présentation et la description des différentes interfaces de l’application.

**Table des matières**

Chapitre 1 10

**1. Contexte**  11

**2. Présentation de l’organisme d’accueil** 11

**2.1 La société I-ETERIA** 11

**2.2** **Les champs d’expertise et activités** 12

**3** **Contexte du projet** 12

**3.1** **Le système de gestion bancaire** 12

**3.2** **Exemple des sites de gestion bancaire** 13

* 1. **Problématique générale du projet** 13

**3.4 Objectif du Projet** 14

**4 Méthodologie de travail** 14

**4.1 Méthode Agile** 14

**Conclusion** 16

Chapitre 2 17

**1.** **Etude de l’existant** 18

**2. Spécification des besoins fonctionnels** 18

**2.1** **Identification Des acteurs** 18

**2.2** **Les besoins fonctionnels** 18

**2.2**.**1** **Les besoins fonctionnels liés à l’administrateur** 19

**3. Besoins non fonctionnels** 19

**4.** **Diagramme de Cas d’utilisation** 20

**4.1** **Diagramme de cas d’utilisation de l’administrateur** 20

**5.** **Description textuelle du cas d’utilisation** 21

**5.1** **Description textuelle du cas d’utilisation « s’authentifier »**  21

**5.2** **Description textuelle du cas d’utilisation « Gérer les alertes »** 22

**5.3** **Description textuelle du cas d’utilisation « Demande d’information »** 23

**5.4 Description textuelle du cas d’utilisation « Récapitulatif des décision »**  24

**5.5** **Description textuelle du cas d’utilisation « Gérer l’historique »**  25

**5.6** **Description textuelle du cas d’utilisation « Intégration des fichiers »**  26

**5.7** **Description textuelle du cas d’utilisation « Gérer les mots de passe »**  27

**5.8** **Description textuelle du cas d’utilisation « Gérer les utilisateurs »**  28

**5.9** **Description textuelle du cas d’utilisation « Lister les utilisateurs »**  29

**5.10** **Description textuelle du cas d’utilisation « Modifier l’utilisateur »**  30

**5.11** **Description textuelle du cas d’utilisation « Supprimer l’utilisateur »**  31

**Conclusion** 32

Chapitre 3 33

**1.** **Outils de travail :** 34

**2.** **Les langages et bibliothèques utilisés** 34

**Conclusion** 36

Chapitre 4 37

**Introduction** 38

**I.** **Partie Administrative** 38

**1.** **Mise en œuvre de l'authentification** 38

**2.** **Page d’accueil** 39

**3.** **Liste des utilisateurs** 40

**3.1** **Ajouter un utilisateur** 40

**3.2 Modifier un utilisateur** 41

**4. Les alertes non traiter** 41

**5**. **Les demandes d’informations**  42

**6.** **Intégration des fichiers** 42

**7. Changement de mot de passe** 43

**Conclusion** 43

**Conclusion générale** 44

**Webographie** 45

**Table des figures**

Figure 1 : Les champs d’expertise et activités ........................................................................................6

Figure 2 : La page d’accueil du site megasol.se .……………………..............................................................7

Figure 3 : Méthode Agile..........................................................................................................................9

Figure 4 : Diagramme de cas d'utilisation de l’administrateur …………………………………………………………………...10

Figure 5 : Page de login…………………………………............................................................................................11

Figure 6 : username ou password ……………………........................................................................................12

Figure 7 : le tableau de bord de l’Admin.....................................................................................................21

Figure 8 : la liste des utilisateurs ………………………………………………………………………………………………………………..22

Figure 9 : formulaire d’ajouter un utilisateur …………………………………………………..……………………………………29

Figure 10 : formulaire de modifier une voiture…………………………............................................................... 30

Figure 11 : formulaire des alertes non traiter……………………………..............................................................30

Figure 12 : formulaire de modèle de demande d’information………….........................................................31

Figure 13 : formulaire d’intégration des fichiers……………...........................................................................32

Figure 14 : formulaire de changer le mot de passe……...............................................................................31

**Liste des abréviations**

|  |  |
| --- | --- |
| **Abréviation** | **Signification** |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| TS | Typescript |
| JS | JavaScript |
| UML | Unified Modeling Language |
| MUI | Material UI |

Tableau 1 : Liste des abréviations

**Introduction générale**

Le marché des technologies a connu une énorme évolution ces dernières années et avec l'intégration de l'internet il y a eu des changements sur la façon et la rapidité d'accès aux informations. Plusieurs secteurs se sont adaptés à ce changement et encore plus ont vu le jour pour satisfaire ces nouveaux clients.

Les banques représentent run des domaines qui connaissent un nombre énorme de client et qui soufflent pour les satisfaire. Ils doivent alors proposer leurs services d'une façon plus rapide et sécurisé mais aussi à distance afin de pouvoir éviter la congestion dans les agences.

Notre projet consiste à développer une application web de gestion bancaire qui propose de nombreuses fonctionnalités à l’utilisateur ainsi que sa partie administrative qui consiste à gérer les services clientèle.

Ce rapport décrit les étapes de développement de notre projet. II contient 5 Chapitres.

Le premier chapitre « Présentation Générale » présente le contexte du travail.

Le deuxième chapitre « Etude théorique et choix techniques » traie l'étude de quelque solution déjà présente dans le marché, voir ce qu'ils apportent et déterminer ce qui manque pour proposer la meilleure solution.

Le troisième chapitre intitulé « Analyse des besoins et spécification »

Ensuite le quatrième chapitre « Conception » détaille l'étude conceptuelle du projet.

Le dernier chapitre « Réalisation » présente les démarches techniques pour la réalisation de l'application ainsi que l'environnement et les outils utilisés.

Finalement on termine avec une conclusion générale

CHAPITRE 1

CONTEXTE GENERALE DU PROJET

1. **Contexte**

Dans le cadre de notre école marocaine des sciences de l'ingénieur et afin de valider nos études nous sommes invités à passer un stage PFA de deux mois dans l’entreprise I-ETERIA. Grâce à ce stage on va découvrir et maîtriser des nouvelles technologies, et être capable à s’intégrer au marché du travail facilement.

1. **Présentation de l’organisme d’accueil**

**2.1 À propos d’I-ETERIA**

**I-ETERIA** Excelle aujourd’hui dans la transformation digitale dans tous ces aspects , nous maîtrisons parfaitement les infrastructures (Administration system et réseaux (Linux REDHAT – AIX – Solaris – HUPUX ) – Base de données ORACLE – administration des serveurs d’applications WEBSPHERE, JBOSS , ordonnancement , Scripting , monitoring et Hypervision , nous faisons aussi la recette et tests fonctionnels et techniques (IST , UAT .. En Méthode Agile avec une intégration continue.

**Administration de système**

Les OS de prédilection sur lesquels intervient i-eteria sont les machine Linux REDHAT Entreprise v6 v7 , une large expertise en terme de tunning , d’analyse de performances de dimensionnement …

**Installation réseaux et infrastructures**

Depuis 2016 , la programmation réseaux et installation fait partie intégrante des activités couvertes par i-eteria , nos ingénieurs disposant d’un compétence confirmée…

**Ingénierie et développement**

I-etrria par sa nature ingénieuse, gère des sujets de développement en utilisant les derniers langages de programmations a la pointe , tel que AJAX , JavaScript , python , bootstrap …

**Objets connectés & IOT**

Internet or Things , le future de la prochaine révolution informatique , avec l’installation des réseaux 5G , i-eteria se voit pionnière dans la communication des objets à travers les ESB d’entreprise…

**2.2 Les champs d’expertise et activités**

Activités

Recherche et technologie et ingénieurie

Design et artwork

Services de consulting

Intelligence artificielle

Champs d’expertise

RH, droit de la propriété intellectuelle et collecte de fond

Electromécanique et technologie

Information et communication technologie

Marché

Biotechnologie

Loisirs et jeux

Figure 1 : Les champs d’expertise et activités

1. **Contexte du projet**

Cette partie du rapport donne une vision globale du projet, son problématique ainsi que son cadre général

* 1. **Le système de gestion bancaire**

L’organisation du système bancaire s’est accomplie en Europe depuis le 19e siècle. A chaque stade de l’évolution de l’économie, le financement de la production a été réalisé par des institutions et mécanismes mettant en œuvre des techniques financières élaborées.

Le système bancaire peut être défini comme étant un ensemble d’institutions et de mécanismes permettant le **fonctionnement des banques**. Ces banques créent la monnaie en consentant des crédits.Elles procurent donc à l’**économie des instruments de paiement**, en mettant à la disposition des clients leur propre monnaie ou la monnaie émise par la banque d’émission.

* 1. **Exemple des sites de gestion bancaire :**

[**https://www.megasol.se/**](https://www.megasol.se/)

Application conçue pour la gestion des banques privées, des banques offshore et d'autres institutions financières impliquées dans les paiements et les prêts, Corniche offre des fonctions de comptabilité multidevises avec une interface d'arrière-guichet uniforme, des services pour les commerçants (cartes), des interfaces de paiement et de puissants services bancaires en ligne. Vous pouvez gérer plusieurs entités avec autant de devises que nécessaire. La conception de Corniche en fait un outil idéal pour la gestion des banques et des succursales de petite et de moyenne taille

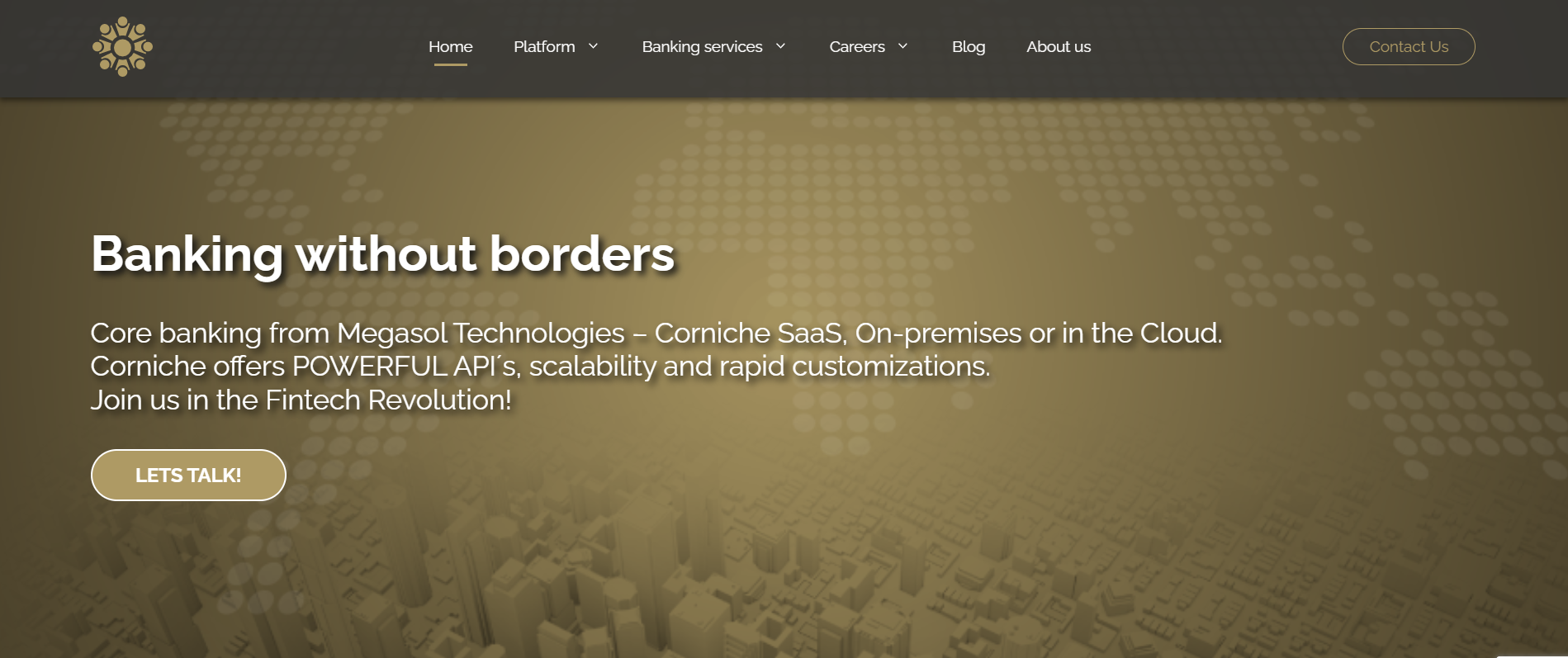
****

Figure 2 : La page d’accueil du site megasol.se

* 1. **Problématique générale du projet**

L'entreprise est un système complexe dans lequel transitent de très nombreux flux d'information.

Sans un dispositif de ce flux l'entreprise peut très vite être dépassée et ne plus fonctionner avec une qualité et quantité de service satisfaisante.

L'enjeu de toute entreprise, qu'elle soit de négoce, industrielle ou des services, mémorise, traite et distribue l'information à un temps record.

La banque est donc une institution dont la principale mission consiste à mobiliser l'épargne à la fructifier par des prêts ou crédits qu'elle accorde aux opérateurs qui en éprouve leurs besoins.

Elle peut autrement être considérée comme un endroit où l'on achète et on vend une marchandise qu'on appelle « argent ».

Les épargnants sont libres de retirer leurs économies, mais comme tous ne se présentent pas au même moment lors du retrait, la banque peut donc utiliser une partie du capital constitué pour satisfaire les demandeurs d'argent sous forme des prêts ou des crédits, de cela, il en découle la mise en place d'un système de contrôle et de suivi du circuit des crédits accordés.

Départ la complexité des opérations effectuées par la banque dont notamment le contrôle et les suivies d'octroi des crédits qui causent comme problème de gestion au sein des institutions.

**3.4 Objectif du Projet**

L’idée de ce projet est de réaliser une application web qui satisfait les besoins du client et lui fait gagner du temps pour ne pas se présenter aux agences.

Par conséquent, l'objectif consiste à développer une application web conviviale et moderne qui propose plusieurs Fonctionnalités bancaires ainsi qu'une simple application web pour la partie administrative de la gestion des services proposés par la partie mobile.

1. **Méthodologie de travail**

Dans cette partie, nous présenterons le processus de développement suivi tout au long du projet, ainsi que la planification qui a été mise en place pour respecter les délais définis au début du projet.

* 1. **Méthode Agile**

La méthode Agile consiste à découper l’implémentation d’un projet en plusieurs phases (appelées itérations). Chacune correspond à un état d’avancement et termine par la fourniture d’un livrable au client. La somme des livrables constitue le produit final.

**Les avantages :**

* La méthode accepte le fait que le besoin peut ne pas être figé, qu’il évolue avec le temps, même si cela doit être cadré. Le design de la solution se construit donc graduellement avec l’avancée du projet : à chaque début de phase, on fixe le périmètre de la solution.
* A chaque début de phase, on planifie le coût de la phase mais aussi la date de livraison du livrable. Les phases correspondent donc à des cycles plus courts.
* Grâce à ces cycles plus courts, le client peut tester plus vite la solution, la valider ou rectifier le tir si nécessaire. Les itérations permettent d’obtenir rapidement les retours des utilisateurs.
* Les priorités peuvent être changées en cours de route : retarder une fonctionnalité qui n’est pas encore bien précisée, ajouter une nouvelle fonctionnalité découverte comme cruciale, abandonner une fonctionnalité obsolète, …

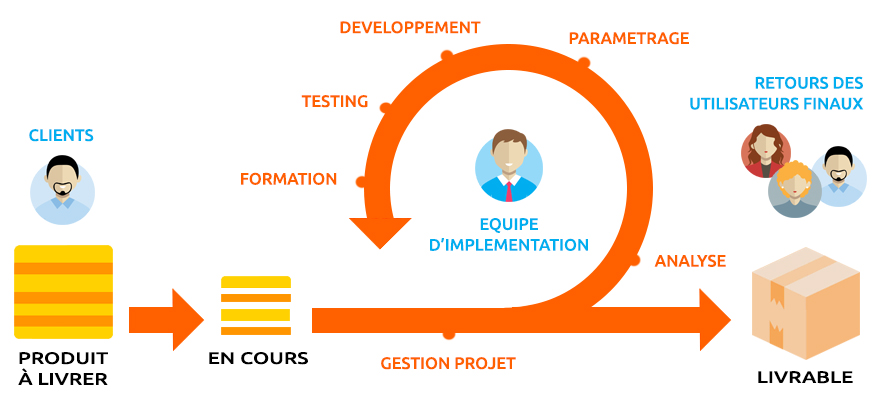


Figure 3 : Méthode Agile

**Planification du Projet**

* **GESTION DE PROJET** : interactions avec les diverses parties prenantes au projet, gestion du planning et des contraintes, ...
* **ANALYSE DES BESOINS** : le but est d’identifier les exigences du client pour la mise en place des fonctionnalités clés du logiciel ainsi que leur faisabilité. Cette étape définit le périmètre de la solution.
* **PARAMÉTRAGE :** paramétrer l’environnement logiciel pour qu’il réponde aux besoins particuliers du client.
* **DÉVELOPPEMENT** : réalisation des parties de code permettant de répondre aux besoins particuliers du client.
* **TESTING** : test de l'adéquation entre le paramétrage/développement et les besoins du client. Recueil des retours des utilisateurs.
* **FORMATION** : des utilisateurs-clés à l'utilisation du programme

**Conclusion**

Dans ce premier chapitre nous avons présenté l’organisme d’accueil, son orientation stratégique, le contexte du projet, des exemples, son problématique générale, son objectif.

Dans le prochain chapitre nous allons nous intéresser aux besoins fonctionnels et non Fonctionnels et à l’étude conceptuelle du projet.

CHAPITRE 2

ANALYSE ET CONCEPTION

**1. Etude de l’existence**

L’étude de l'existence constitue le cœur de la phase d'analyse d'un projet. Cette étape est primordiale pour la mise en route de tout projet informatique ou autre, qui permet de définir le contexte de fonctionnement, ou bien le processus métier, et de dégager les différentes imperfections dans le système actuel afin de les corriger

Megasol présente une application web facile à utiliser avec une interface accueillante et qui offre des services corme :

• Effectuer des recherches multicritères sur les écritures du compte

• Consulter les cours des devises

• Parcourir l'historique de ses opérations de dépôt ou de retrait

• Consulter son portefeuille Crédit, Tire et Sicav

• Consulter ses opérations monétiques

• Planifier un rendez-vous avec le chef d'agence ou le chargé clientèle (avec visioconférence)

• Suivre les cours de devise et l'historique des TMM

Il existe des sites open source prêt à être utilisés, mais nous avons décidé de réaliser notre propre application web pour mettre en pratique nos connaissances informatiques, pour sécuriser notre application car un site prêt peut contenir des failles.

Grâce à cette étude, nous avons eu une idée sur les solutions existantes dans le marché, les points forts qui vont représenter des fonctionnalités nécessaires dans notre plateforme. De plus, nous allons essayer d’améliorer les points faibles des autres concurrents.

**2. Spécification des besoins fonctionnel**

**2.1 Identification Des acteurs :**

L’administrateur : : c’est l’acteur principal de notre application le rôle de cet acteur c’est de gérer les utilisateurs, gérer les alertes, gérer les demandes d’information, gérer l’historique

**2.2 Les besoins fonctionnels :**

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée-sortie du Système.

**2.2.1 Les besoins fonctionnels liés** **à l’administrateur :**

Le système doit permettre à l’administrateur :

* Gérer les alertes
* Traiter les alertes
* Traiter les informations des clientes
* Demander les informations
* Prend des décisions
* Gérer l’historiques
* La gestion des utilisateurs
* Gérer les mots de passe

**3. Besoins non fonctionnels**

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation (langage de programmation, type SGBD, de système d'Exploitation...) Dans le cadre de ce travail, l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités. L'application devra être capable de :

* Tourner en réseau,
* Être compatible avec n'importe quel appareil.

IL faudra aussi noter que l'application devra être hautement sécurisée car les informations ne devront pas être accessibles à tout le monde.

**4. Diagramme de Cas d’utilisation**

Les diagrammes de cas d'utilisation (DCU) sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (Actors), ils interagissent avec les cas d'utilisation (use cases).

Le diagramme de cas d’utilisation représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est utilisé dans les deux étapes de capture des besoins fonctionnels et techniques. A partir de l’étude préliminaire, nous avons pu dégager le diagramme des cas d’utilisation général suivant :

**4.1 Diagramme de cas d’utilisation de l’administrateur**

Cette figure présente les différents besoins cités sur un diagramme de cas d’utilisation. Les associations simples entre l’acteur (Administrateur) et ses cas d’utilisation Elles expliquent qu’il y a une relation directe entre l’acteur et les services rendu par le système. L'administrateur doit obligatoirement s'identifier avant tout.

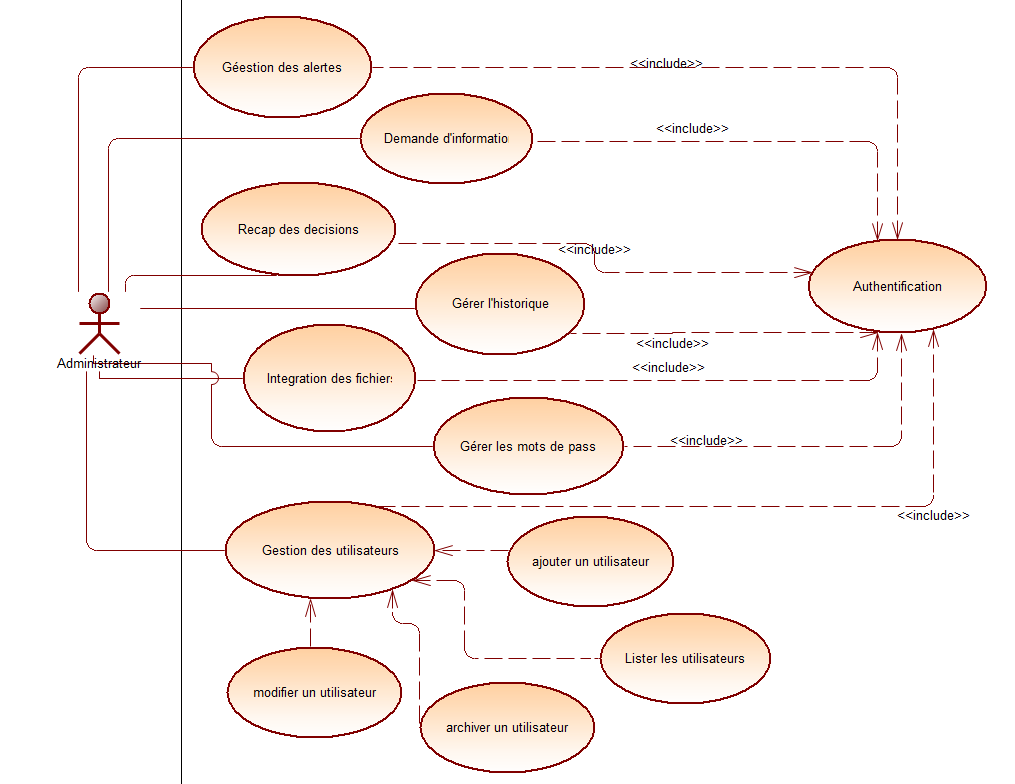


Figure 4 : Diagramme de cas d'utilisation de l’administrateur

**5. Description textuelle du cas d’utilisation**

Dans cette partie nous allons décrire quelques cas d’utilisation.

1. Description textuelle du cas d’utilisation « s’authentifier » :

Le tableau ci-dessous décrit de cas d’utilisation « s’authentifier » et la démarche suivie afin d’accéder à l’application :

**Sommaire d’authentification**

**Titre** : Authentification.

**Acteur** : Administrateur.

**Objectif** : Lors du lancement de l’application, l’administrateur doit se connecter pour accéder à son espace.

**Description des enchaînements**

Le cas d’utilisation commence lorsque l’administrateur saisie l’URL de la page d’authentification.

**Préconditions :** L’administrateur doit être enregistré dans la base de données.

**Enchaînement nominal :**

1. L’administrateur accède au site web.
2. Clique sur le Button se connecter
3. Le système fournit un formulaire d’authentification.
4. L’administrateur doit saisir son identifiant et son mot de passe.
5. Le système vérifie les coordonnées saisies.
6. Le système redirige L’administrateur vers la page administrative.

**Enchaînement alternatif :**

A1. Les coordonnées saisies sont incorrectes :

L'enchaînement démarre au point 5 du scénario nominal. le système indique que l’identifiant ou le mot de passe est erroné La séquence nominale reprend au point 4.

A2. Le champ Login est vide :

L'enchaînement démarre au point 5 de la séquence nominale :

Le système indique que le champ login est obligatoire

La séquence nominale reprend au point 4.   
A3. Le champ mot de passe est vide :  
L'enchaînement démarre après le point 5 de la séquence nominale :   
Le système indique que le champ mot de passe est obligatoire   
La séquence nominale reprend au point 4.   
**Post-conditions :**   
En cas de réussite d'authentification, l'utilisateur est redirigé vers la page administrative

Tableau 1 : Diagramme de cas d'utilisation de l’administrateur

2. Description textuelle du cas d’utilisation « Gérer les alertes » :

Le tableau ci-dessous représente la description textuelle du cas d’utilisation « Gérer les alertes» :

**Sommaire gérer les alertes**

**Titre :** gérer les alertes.   
**Acteurs :** Administrateur.   
**Objectif** : Administrateur gérer une alerte

**Description des enchaînements**

**Préconditions :**

Administrateur authentifié, alertes n’est pas trouver

**Enchaînement nominal :**

1. L’administrateur demande la page « gérer les alertes».
2. Le système lui affiche la page demandée.
3. L’administrateur demande le formulaire « alertes non traiter ».
4. Le système lui affiche le formulaire demandé.
5. L’administrateur doit traiter l’alerte.
6. Le système vérifie la validité des champs remplis.

**Enchaînement alternatif :**

A1. Les champs remplis sont invalides :

L'enchaînement démarre au point 4 de la séquence nominale

Le système indique que les champs ne sont pas valides et affiche un message d'erreur.

La séquence nominale reprend au point 3.

A2. Les champs obligatoires sont vides :

L'enchaînement démarre après le point 4 de la séquence nominale Le système indique que les champs sont obligatoires. La séquence nominale reprend au point 3.

**Post-conditions :**

L’alerte est traité

3. Description textuelle du cas d’utilisation « demande d’information » :

Le tableau ci-dessous représente la description textuelle du cas d’utilisation « demande d’information » :

**Sommaire de demande d’information**

**Titre :** demande d’information.   
**Acteurs :** Administrateur.   
**Objectif** : L’administrateur demande d’information

**Description des enchaînements**

**Préconditions :**

Administrateur authentifié, l’information n’existe pas **:**

**Enchaînement nominal :**

1. L’administrateur demande la page « demande d’information »
2. Le système lui affiche la page demandée
3. L’administrateur choisi la demande
4. Le système valide la demande .

**Enchaînement d'exception :**

E1 : si la demande n’existe pas :

L’enchaînement démarre au point 6 de la séquence nominale !

Le système indique que la demande n’existe pas.

**Post-conditions :** demande d’information

4. Description textuelle du cas d’utilisation « récapitulatif des décision » :

Le tableau ci-dessous représente la description textuelle du cas d’utilisation « récapitulatif des décision » :

**Sommaire de récapitulatif des décisions**

**Titre :** récapitulatif des décisions  
**Acteurs :** Administrateur.   
**Objectif** L’administrateur récapitulatif des décisions

**Description des enchaînements**

**Préconditions :**

Administrateur authentifié, la décision n’est pas trouvée **:**

**Enchaînement nominal :**

1. L’administrateur demande la page « récapitulatif des décisions »
2. Le système lui affiche la page demandée
3. L’administrateur choisi l’un des décisions
4. Le système valide la demande.

**Enchaînement d'exception :**

E1 : si la décision n’est pas trouvée :

L’enchaînement démarre au point 6 de la séquence nominale !

Le système indique que la décision n’existe pas.

**Post-conditions :** récapitulatif des décisions

5. Description textuelle du cas d’utilisation « Gérer l’historique » :

Le tableau ci-dessous représente la description textuelle du cas d’utilisation « gérer l’historique » :

**Sommaire de gérer l’historique**

**Titre :** gérer l’historique  
**Acteurs :** Administrateur.   
**Objectif** L’administrateur gérer l’historique

**Description des enchaînements**

**Préconditions :**

Administrateur authentifié, l’historique n’existe pas **:**

**Enchaînement nominal :**

1. L’administrateur demande la page « Historique »
2. Le système lui affiche la page demandée
3. L’administrateur sélectionné l’alerte
4. Le système afficher l’alerte.

**Enchaînement d'exception :**

E1 : si l’historique n’est pas trouvée :

L’enchaînement démarre au point 6 de la séquence nominale !

Le système indique que l’historique n’existe pas.

**Post-conditions :** affiche l’historique

6. Description textuelle du cas d’utilisation « intégration des fichier » :

Le tableau ci-dessous représente la description textuelle du cas d’utilisation « intégration des fichier » :

**Sommaire de gérer l’intégration des fichiers**

**Titre :** Intégration des fichiers  
**Acteurs :** Administrateur.   
**Objectif** L’administrateur gérer l’intégration des fichiers

**Description des enchaînements**

**Préconditions :**

Administrateur authentifié, fichier n’existe pas **:**

**Enchaînement nominal :**

1. L’administrateur demande la page « intégration des fichiers »
2. Le système lui affiche la page demandée
3. L’administrateur click sur le Betton d’insertion
4. Le système afficher la liste des fichiers
5. L’administrateur choisi le fichier
6. Le système valide l’intégration.

**Enchaînement d'exception :**

E1 : si le fichier n’est pas trouvée :

L’enchaînement démarre au point 6 de la séquence nominale !

Le système indique que le fichier n’existe pas.

**Post-conditions :** intégrer le fichier

7. Description textuelle du cas d’utilisation « Gérer les mots de passe » :

Le tableau ci-dessous représente la description textuelle du cas d’utilisation « gérer les mots de passe » :

**Sommaire de gérer les mots de passe**

**Titre :** gérer les mots de passe  
**Acteurs :** Administrateur.   
**Objectif** L’administrateur gérer les mots de passe

**Description des enchaînements**

**Préconditions :**

Administrateur authentifié, mot de passe existe **:**

**Enchaînement nominal :**

1. L’administrateur demande la page « gérer les mots de passe »
2. Le système lui affiche la page demandée
3. L’administrateur entrer l’ancien mot de passe et écrire le nouveau
4. L’administrateur click sur le Betton confirmer
5. Le système valide le mot de passe.

**Enchaînement d'exception :**

E1 : si le mot de passe n’est pas trouvé :

L’enchaînement démarre au point 6 de la séquence nominale !

Le système indique que le mot de passe n’existe pas.

**Post-conditions :** mot de passe changé

8. Description textuelle du cas d’utilisation « Gérer les utilisateur » :

Le tableau ci-dessous représente la description textuelle du cas d’utilisation « gérer les utilisateur » :

**Sommaire de gérer les utilisateurs**

**Titre :** Ajouter un utilisateur  
**Acteurs :** Administrateur.   
**Objectif** L’administrateur ajouter un nouvel utilisateur

**Description des enchaînements**

**Préconditions :**

Administrateur authentifié, mot de passe existe **:**

**Enchaînement nominal :**

1. L’administrateur demande la page « gérer les utilisateurs ».
2. Le système lui affiche la page demandée.
3. L’administrateur demande le formulaire « ajouter utilisateur ».
4. Le système lui affiche le formulaire demandé.
5. L’administrateur doit remplir les données à ajouter.
6. Le système vérifie la validité des champs remplis.
7. Le système enregistre l’ajout.

**Enchaînement alternatif :**

A1. Les champs remplis sont invalides :

L'enchaînement démarre au point 4 de la séquence nominale

Le système indique que les champs ne sont pas valides et affiche un message d'erreur.

La séquence nominale reprend au point 3.

A2. Les champs obligatoires sont vides :

L'enchaînement démarre après le point 4 de la séquence nominale Le système indique que les champs sont obligatoires. La séquence nominale reprend au point 3.

**Post-conditions :**

L’utilisateur ajoutée

9. Description textuelle du cas d’utilisation « Lister les utilisateur » :

Le tableau ci-dessous représente la description textuelle du cas d’utilisation « Lister les utilisateur » :

**Description des enchaînements**

**Préconditions :**

Administrateur authentifié, l’utilisateur existe **:**

**Enchaînement nominal :**

1. L’administrateur demande la page « gérer les utilisateurs »
2. Le système lui affiche la page demandée
3. L’administrateur click sur liste des utilisateurs
4. Le système affiche les utilisateurs.

**Enchaînement d'exception :**

E1 : si la liste des utilisateurs n’est pas trouvée :

L’enchaînement démarre au point 6 de la séquence nominale !

Le système indique que l’utilisateur n’existe pas.

**Post-conditions :** affiche la liste des utilisateurs

**Sommaire de Lister les utilisateurs**

**Titre :** Lister les utilisateurs  
**Acteurs :** Administrateur.   
**Objectif** L’administrateur lister les utilisateurs

10. Description textuelle du cas d’utilisation « Modifier l’utilisateur » :

Le tableau ci-dessous représente la description textuelle du cas d’utilisation « Modifier l’utilisateur » :

**Sommaire de modifier l’utilisateur**

**Titre :** Modifier l’utilisateur.  
**Acteurs :** Administrateur.   
**Objectif** L’administrateur modifie l’utilisateur

**Description des enchainements**

**Préconditions :**

Administrateur authentifié, utilisateur existe.

**Enchaînement nominal :**

1. L’administrateur demande la page « Les utilisateur »
2. Le système lui affiche la page demandée
3. L’administrateur recherche l’utilisateur.
4. L’administrateur clique sur le Button modifier
5. L’administrateur modifie les donnes de l’utilisateur.
6. Le système vérifie la validité des champs modifiés.
7. Le système enregistre la modification.

**Enchaînement alternatif :**

A1. Les champs remplis sont invalides :

L'enchaînement démarre au point 6 de la séquence nominale

Le système indique que les champs ne sont pas valides et affiche un message d'erreur.

La séquence nominale reprend au point 5.

A2. Les champs obligatoires sont vides :

L'enchaînement démarre après le point 8 de la séquence nominale

11. Description textuelle du cas d’utilisation « Supprimer l’utilisateur » :

Le tableau ci-dessous représente la description textuelle du cas d’utilisation « Supprimer l’utilisateur » :

**Sommaire de supprimer l’utilisateur**

**Titre :** Supprimer l’utilisateur.  
**Acteurs :** Administrateur.   
**Objectif** L’administrateur supprimer l’utilisateur

**Description des enchaînements**

**Préconditions :**

Administrateur authentifié, l’utilisateur existe

**Enchaînement nominal :**

1. L’administrateur demande la page « Les utilisateurs »
2. Le système lui affiche la page demandée
3. L’administrateur chercher l’utilisateur qu’il veut supprimer
4. L’administrateur supprime l’utilisateur.
5. Le système enregistre la suppression.

**Enchaînement d'exception :**

E1 : si l’utilisateur n’existe pas :

L’enchaînement démarre au point 6 de la séquence nominale !

Le système indique que l’utilisateur n’existe pas.

**Post-conditions :** l’utilisateur supprimée

**Conclusion**

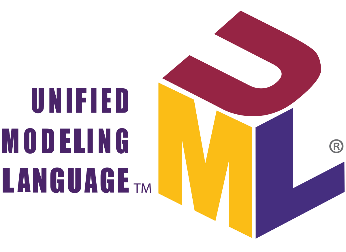
Cette phase nous a permis de comprendre le contexte du système, identifier les principaux cas d’utilisation, ce qui nous a facilité la modélisation de notre base de données. Le chapitre suivant sera consacré à la présentation des outils de travail.

CHAPITRE 3

PRÉSENTATION DES OUTILS DE TRAVAIL

**1.** **Outils de travail :**

1. **UML :**



Le langage UML (**Unified Modeling Language**) 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.



Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.

1. **Les langages et bibliothèques utilisés**

**1. HTML:**

Le HyperText Markup Language, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C’est un langage permettant d’écrire de l’hypertexte, d’où son nom.

Il permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages.



**2. CSS:**



Le terme CSS est l'acronyme anglais de ‘Cascading Style Sheets’ qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.

**3. JavaScript :**

JavaScript est un langage de programmation qui permet d’implémenter des mécanismes complexes sur une page web. À chaque fois qu’une page web fait plus que simplement afficher du contenu statique, du contenu mis à jour à des temps déterminés, des cartes interactives, des animations 2D/3D, des menus vidéo défilants, etc...



**4. TypeScript :**



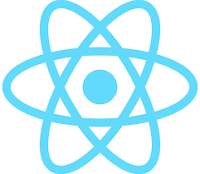
TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Il s'agit d'un sur-ensemble syntaxique strict de JavaScript.

**5. Bootstrap :**

Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web.

C'est un ensemble qui contient des codes HTML et CSS.



**6. React :**

React est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page HTML à chaque changement d'état

**7. Material-ui :**



MUI offre une suite complète d'outils d'interface utilisateur pour vous aider à livrer de nouvelles fonctionnalités plus rapidement. Commencez avec Material UI, notre bibliothèque de composants entièrement chargée, ou apportez votre propre système de conception à nos composants prêts pour la production.

Conclusion

Dans ce chapitre nous avons fait une étude technique de ce projet en présent les outils de travail, les langages et bibliothèques utilisées pour ce projet Dans le chapitre suivant nous allons présenter quelques interfaces principales qui illustrent les différents cas d’utilisation déjà vus dans le chapitre 2.

CHAPITRE 4

REALISATION DE LA SOLUTION

**Introduction**

Dans ce chapitre, nous détaillons l’implémentation de notre application. Ainsi, nous développons les différents modules de l’application. Nous présentons également l’utilisation de l’application via les fenêtres de base.

1. **Partie Administrative**
2. **Mise en œuvre de l'authentification**

Cette interface est primordiale, c’est la phase d’identification pour accéder à l’application. Dès la connexion, la page d’authentification s’affiche. Elle est composée de la zone ‘username’ et la zone ‘password’, l’utilisateur accède à l’interface dédiée selon ses privilèges

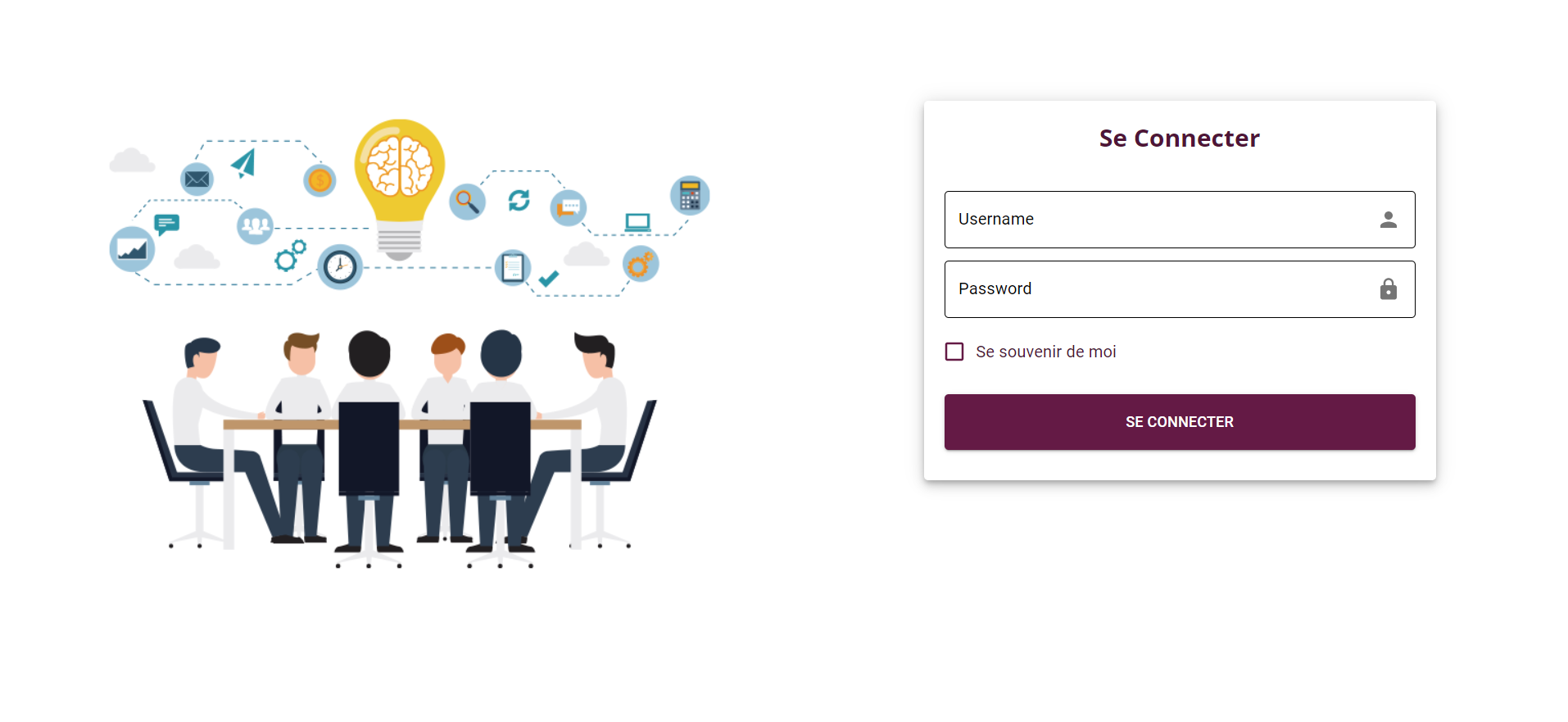


Figure 5 : Page de login

La partie administrative est sécurisée par un mot de passe. Les pages d’administrations sont sécurisés à l’aide de variables de session.

Un message d'erreur spécifique apparait en cas d'erreur :

* **Si le couple login / mot de passe est incorrect**

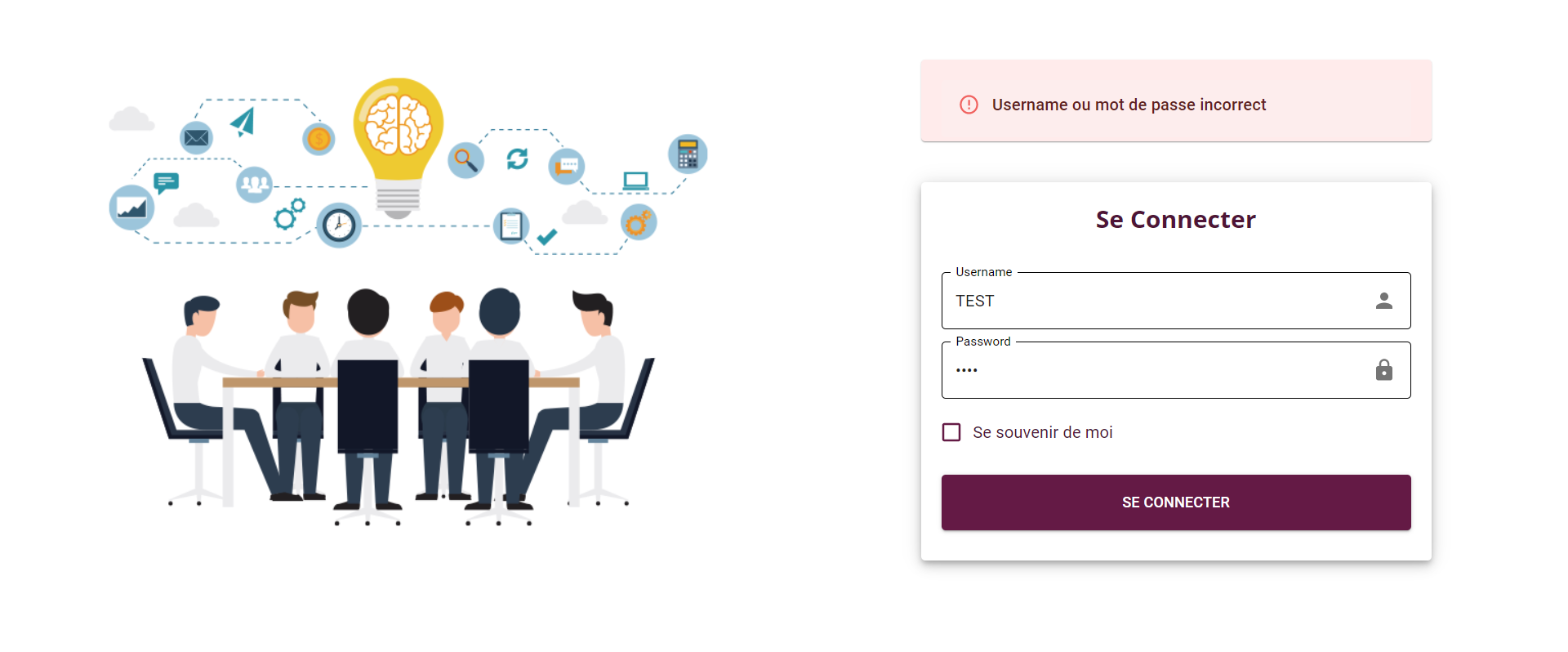


Figure 6 : username ou password incorrect

1. **Page d’accueil Administrative**

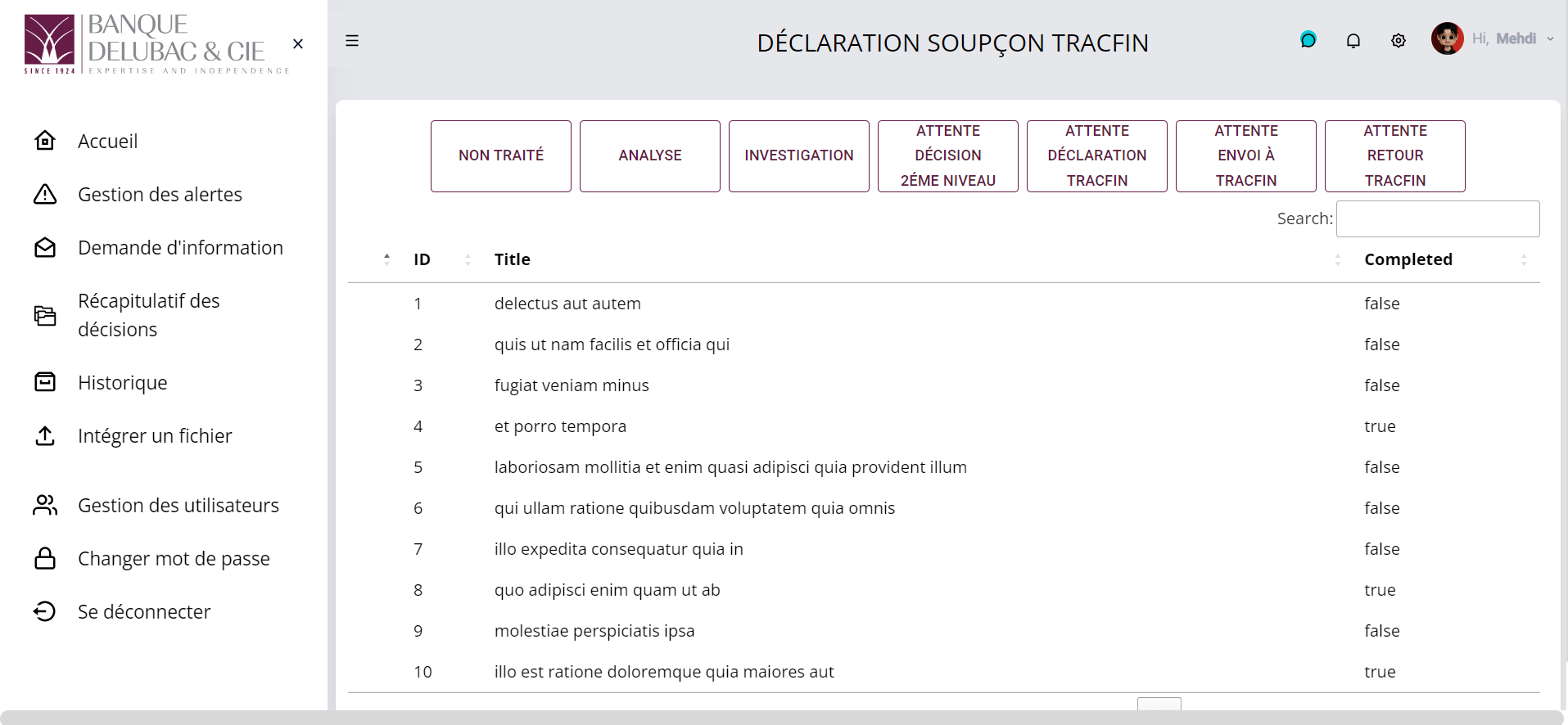


Figure 7 : le tableau de bord de l’Admin

Cette interface, c’est le Tableau de bord de notre application. Dès authentification, cette page sera affichée. Elle est composée d’une barre de navigation pour accéder à d’autre page

1. **Liste des utilisateurs.**

Cette page contient toutes les informations (Id,Nom, numéro de téléphone et email) des utilisateurs   
  
En plus de ces informations l’administrateur a le droit de supprimer ou modifier un utilisateur.

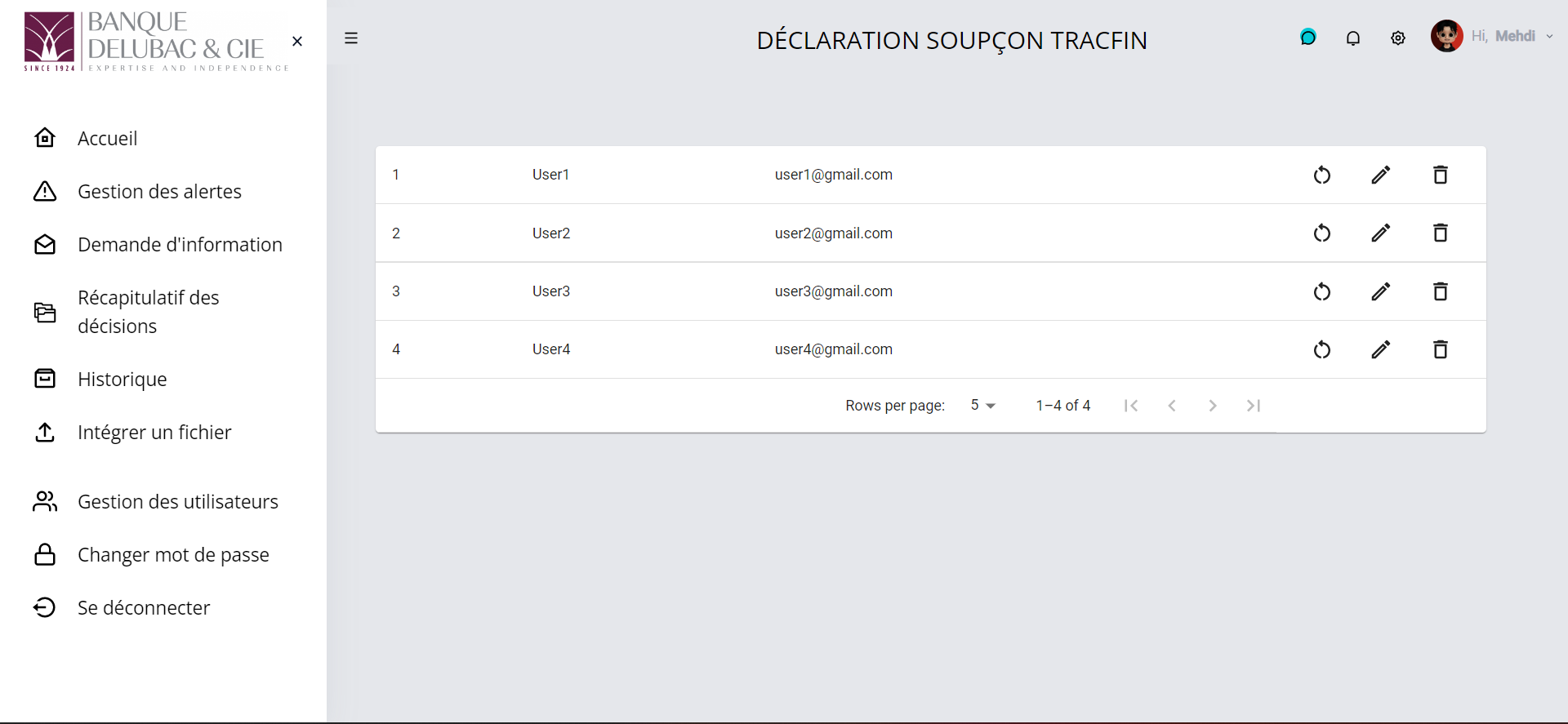


Figure 8 : la liste des utilisateurs

**3.1 Ajouter un utilisateur**

Cette page contient un formulaire qui permet d’ajouter un utilisateur à la base de données, avec les champs : nom, prénom, fonction, sexe, email, téléphone, fax, mot de passe.

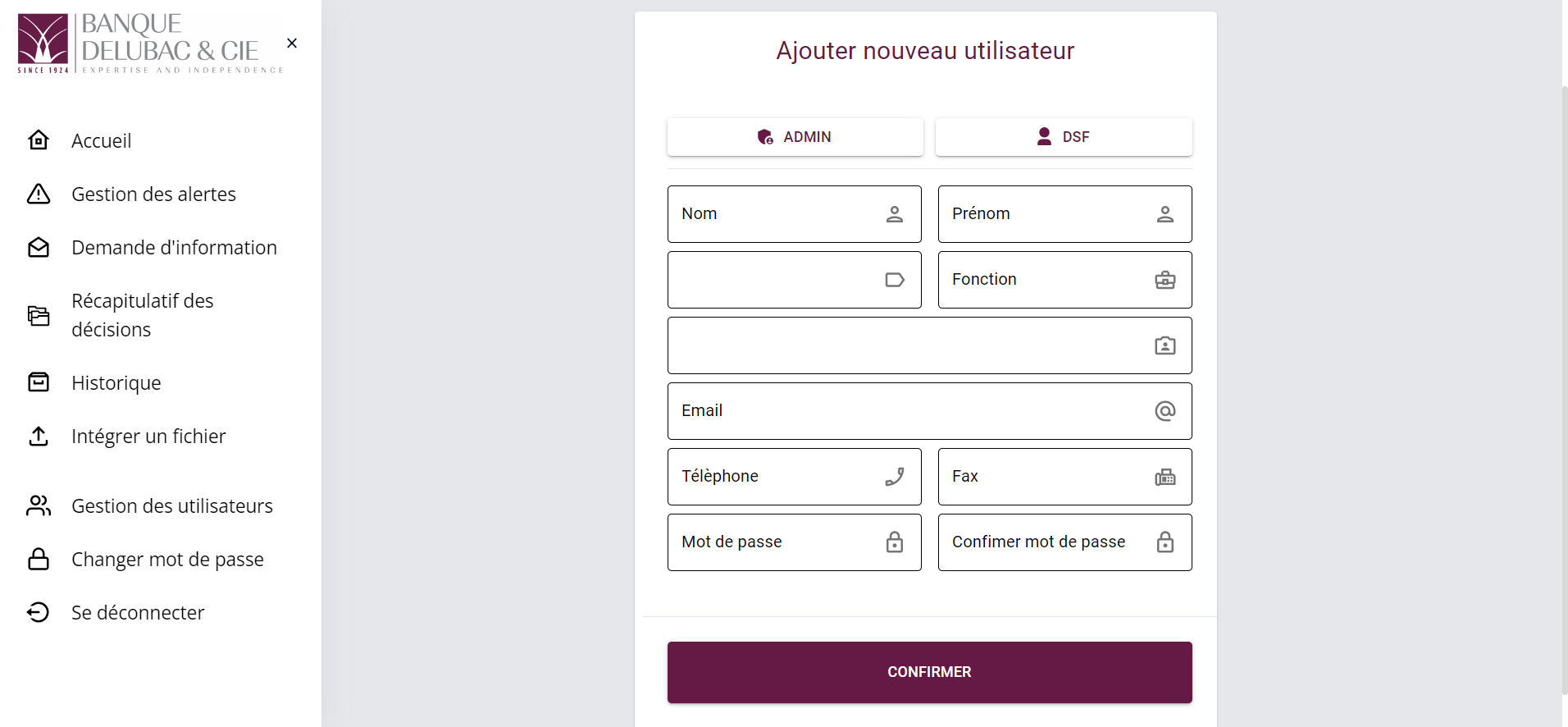


Figure 9 : formulaire d’ajouter un utilisateur

* 1. **Modifier une voiture**

Cette page permet de modifier les informations des utilisateurs.

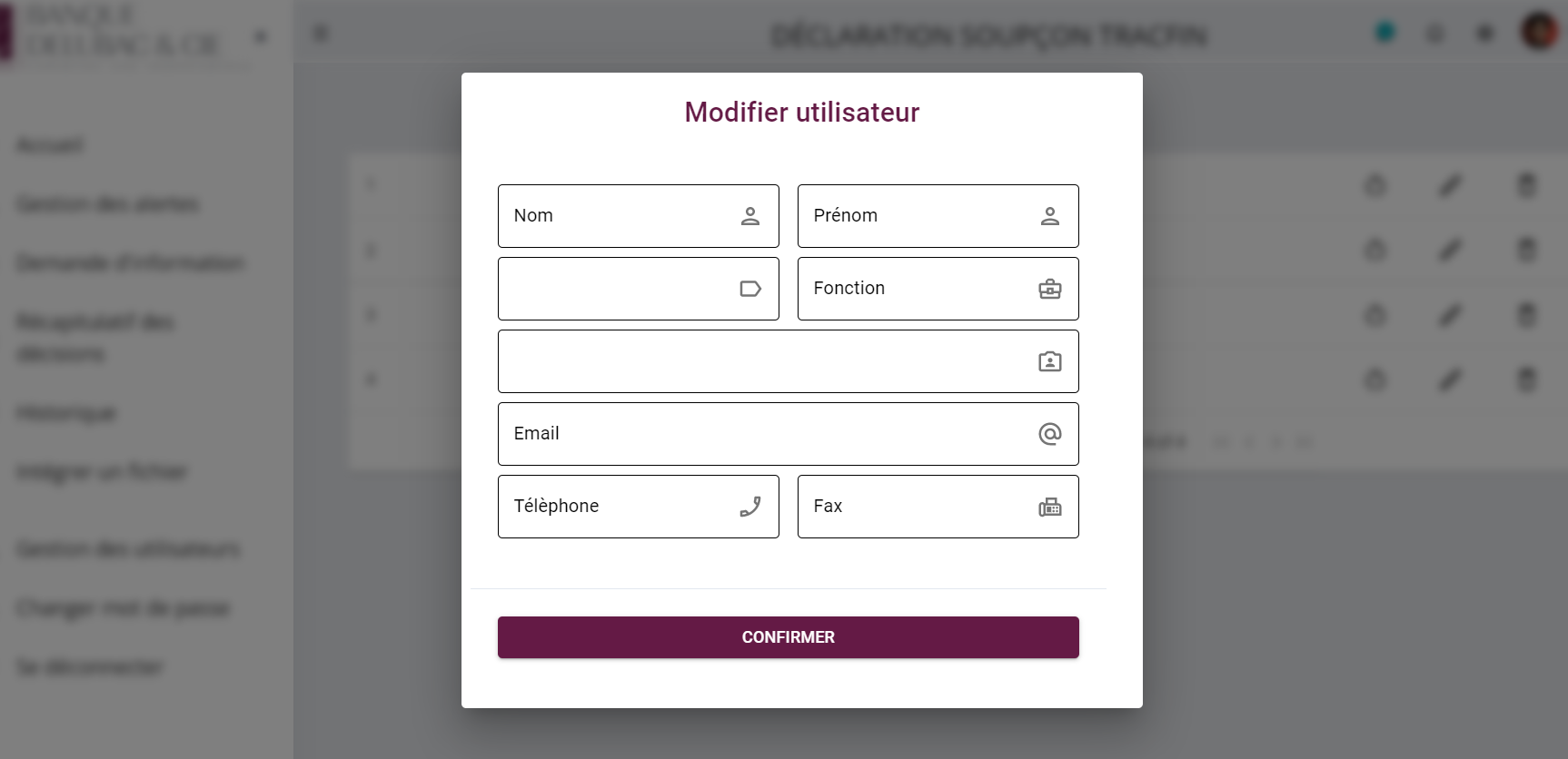


Figure 10 : formulaire de modifier une voiture

1. **Les alertes non traiter.**

Cette page représente toutes les alertes qui on pas en cor traiter.

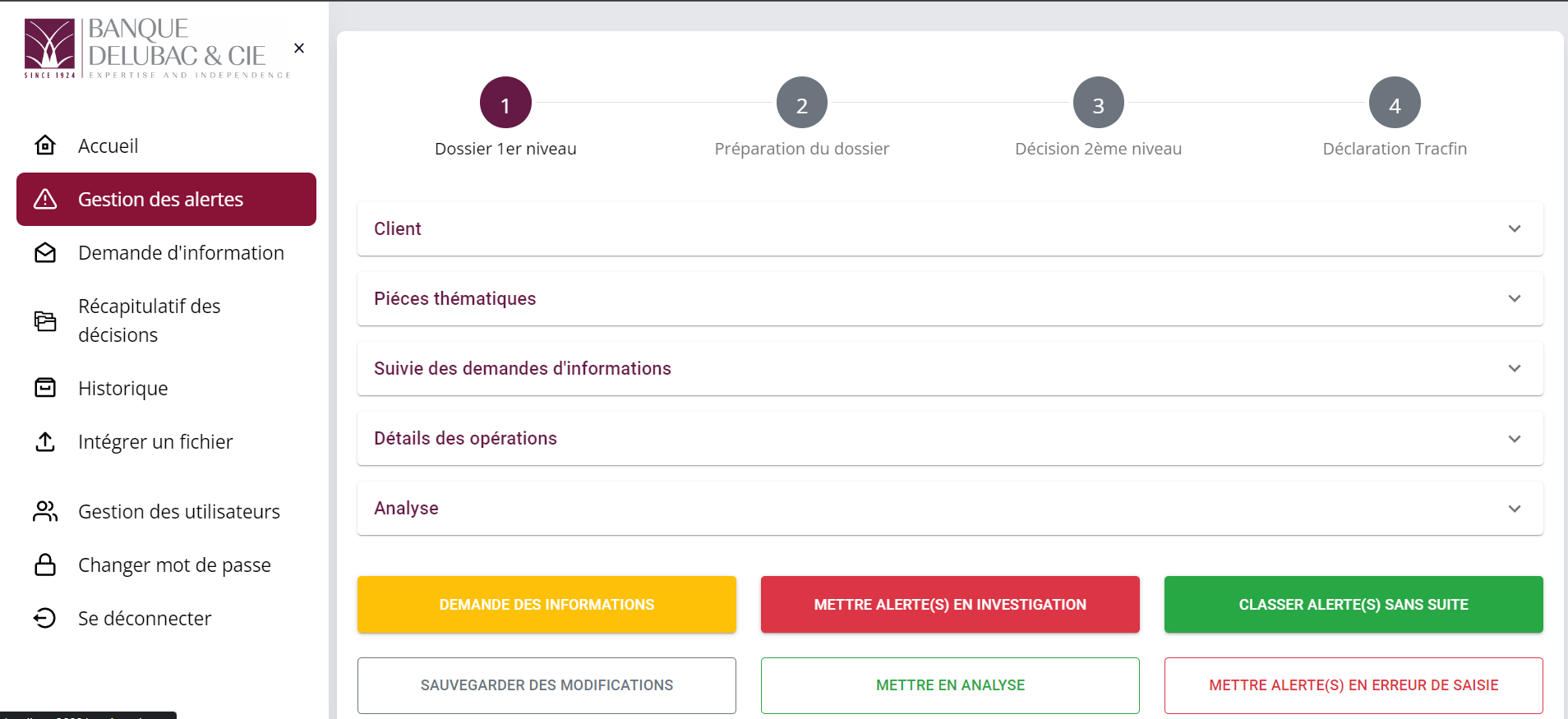


Figure 11 : formulaire des alertes non traiter

1. **Les demandes d’informations.**

Cette page représente les modèles des demandes d’information.

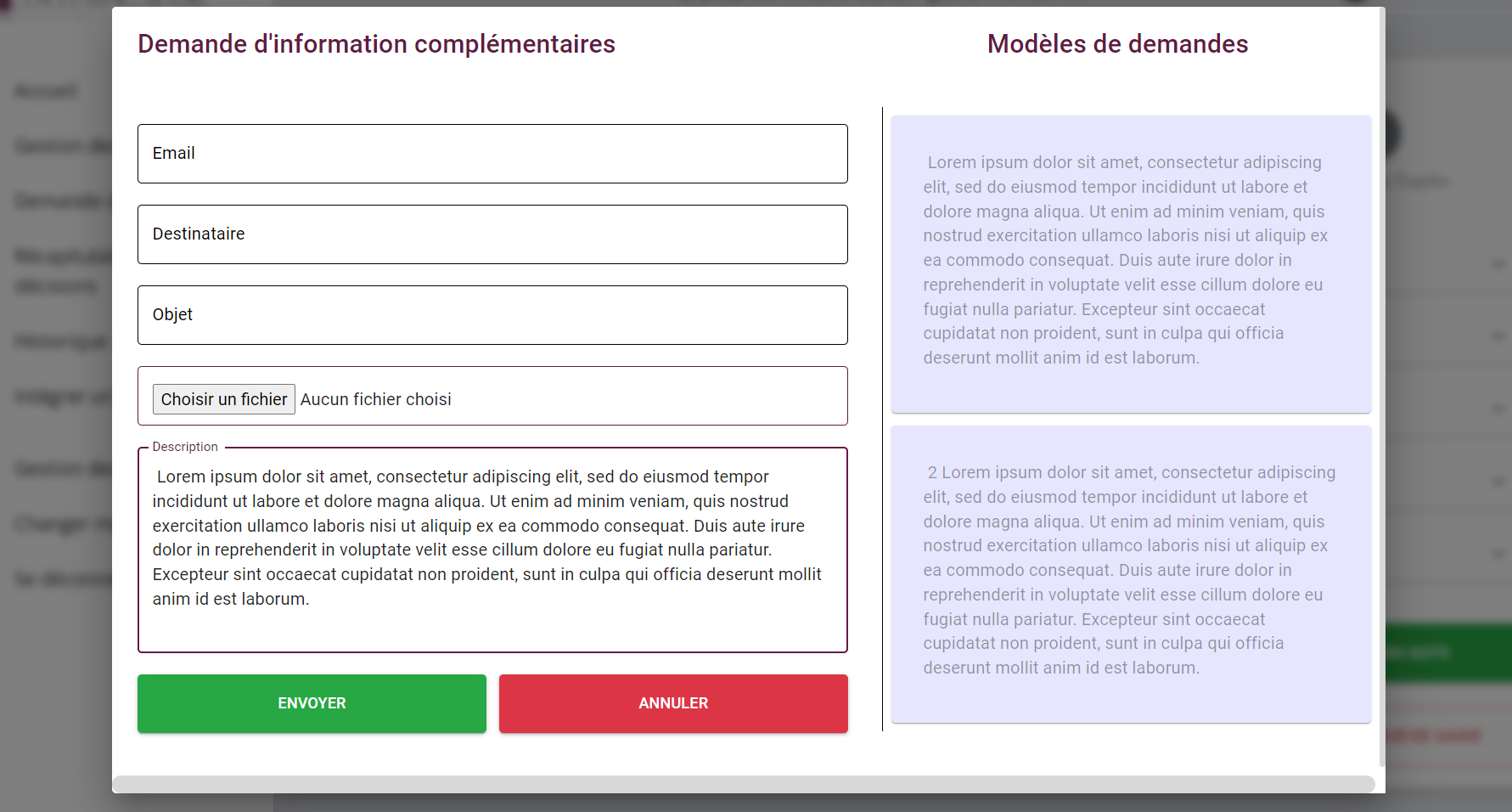


Figure 12 : formulaire de modèle de demande d’information

1. **Intégration des fichiers.**

Cette page représente l’emplacement pour intégrer des fichiers

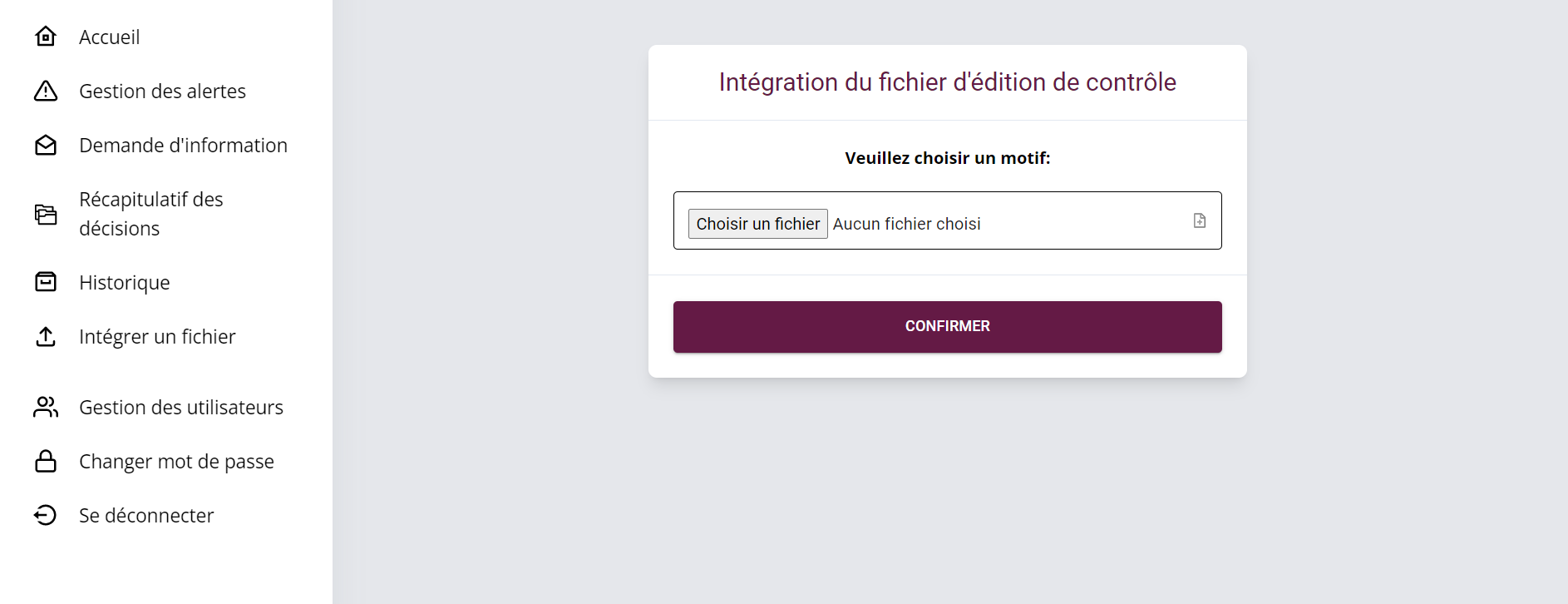


Figure 13 : formulaire d’intégration des fichiers

1. **Changement de mot de passe.**

Cette page représente la page pour changer le mot de passe

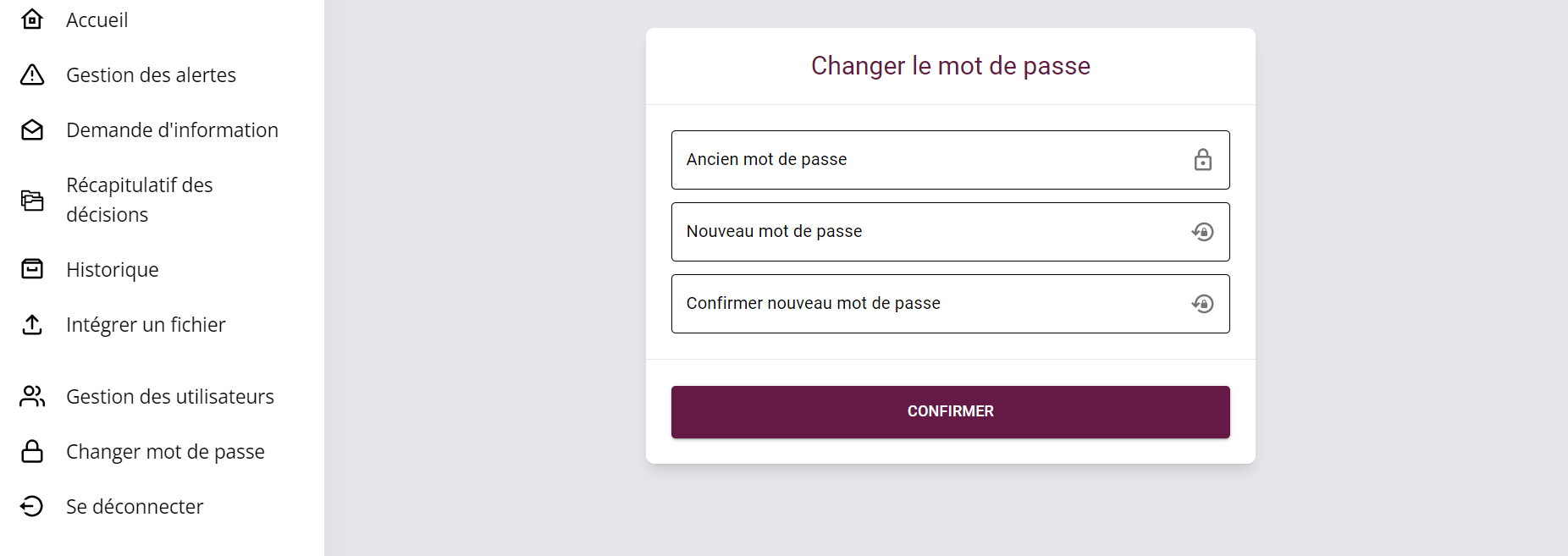


Figure 14 : formulaire de changer le mot de passe

Conclusion

Dans cette partie, on a présenté quelques interfaces de l’application web qu'on a conçu. On général ce chapitre a pour présenter les différents aspects pratiques de projet.

Conclusion Générale

L’objectif principal de ce projet est la conception et la réalisation d'un site web pour la gestion des système bancaires

Afin de mener à bien le projet, la solution que nous proposons réside dans l'adoption d'une démarche qui consiste à suivre une méthodologie de travail. Nous avons commencé par l'étude de l’existant, spécifié les besoins des utilisateurs, procédé la conception et finalement nous avons recours à la réalisation suivie d'un test selon les normes de qualité.

Nous estimons que ce projet nous a été bénéfique sur tous les plans à savoir le plan technique et le plan humain.

Pour ce qui concerne le plan technique, le projet a été une véritable occasion pour découvrir et maitriser la technologie react js.

Pour ce qui concerne le plan humain, nous avons eu la chance d'améliorer nos capacités de se communiquer et du travailler au sein d'une équipe.

Nous avons essayé dans notre travail de respecter les exigences présentées par le cahier de charges et nous estimons par ailleurs que notre application peut être enrichie par d'autres fonctionnalités qui sont les suivantes :

* Améliorer l'aspect sécuritaire en sécurisant d’avantage l’accès à la base de données.
* Ajout d'autres fonctionnalités :
* Une réalisation de backend.
* Envoyé notification au client pour confirmation la demande.

Webographie

<https://mui.com/material-ui/getting-started/overview/>

https://reactjs.org/docs/getting-started.html/

<https://www.w3schools.com/>

<https://developer.mozilla.org/fr/>

<https://getbootstrap.com/>

<https://javascript.info/>

https://stackoverflow.com/