**PROJET PFA**



**N°Année :4ème Année**

**Sujet : Plateforme web pour un générateur de site**

**Ingénierie Informatique et Réseaux**

Encadré par :

IOUNOUSSE Jawad

Réalisé par :

DOUBABI Abderrahman

TANASSA Loubna

BENHIMA Mohammed-Taha

ESSAOUD Oumaima

##### REMERCIEMENTS

##### 

##### Nous souhaitons exprimer notre sincère reconnaissance à M. IOUNOUSSE Jawad, notre professeur encadrant, pour son accompagnement tout au long de ce Projet de Fin d’Année. Sa disponibilité, sa rigueur méthodologique et ses conseils avisés ont grandement contribué à l’orientation et à la qualité de notre travail. Son engagement constant et sa bienveillance à notre égard ont été pour nous une source de motivation et d’inspiration. Grâce à lui, nous avons pu mener à bien ce projet en développant à la fois nos compétences techniques et notre esprit d'équipe.

# RESUMÉ

Dans le cadre de notre Projet de Fin d’Année, nous avons eu l’opportunité de développer une application web innovante baptisée GeneGuard, visant à contribuer à la prévention des maladies héréditaires. Ce projet a été réalisé en équipe de quatre étudiants au sein de l'école EMSI.

Face à la problématique de la détection tardive de certaines maladies génétiques et au coût élevé des tests médicaux, notre client avait besoin d’un outil numérique capable d’évaluer, de manière simple et accessible, le risque qu’une personne développe une maladie héréditaire.

L’application GeneGuard permet à l’utilisateur de sélectionner une maladie à tester (comme le diabète ou certains cancers), de répondre à un questionnaire médical ciblé, et de recevoir une estimation de son risque personnel, accompagnée de recommandations préventives.

Pour concrétiser ce projet, nous avons utilisé des technologies web modernes et un algorithme d’analyse basé sur des données médicales. L’accent a été mis sur l’intuitivité de l’interface et la pertinence des résultats, dans le but de sensibiliser le grand public tout en offrant un outil d’évaluation fiable.

Ce rapport détaille notre démarche depuis la phase d’analyse des besoins jusqu’à la réalisation finale, en passant par la conception UML et le développement de l’application.

Table des matières

[REMERCIEMENTS 2](#_Toc197809887)

[RESUMÉ 3](#_Toc197809888)

[TABLES DES FIGURES 6](#_Toc197809889)

[TABLES DES TABLEAUX 7](#_Toc197809890)

[Introduction 8](#_Toc197809891)

[Chapitre1 : Contexte général du projet 9](#_Toc197809892)

[1. CONDUITE DU PROJET 10](#_Toc197809893)

[1.1. Cycle de développement : 10](#_Toc197809894)

[1.2. Processus de développement : 10](#_Toc197809895)

[2. L'ACCOMPLISSEMENT DU PROJET 13](#_Toc197809896)

[2.1. Les réunions : 13](#_Toc197809897)

[2.2 Planification des sprints 14](#_Toc197809898)

[3. Conclusion : 17](#_Toc197809899)

[Chapitre2: Analyse et conception 18](#_Toc197809900)

[1. Architecture de l’application : 19](#_Toc197809901)

[2. Architecture MVT: 19](#_Toc197809902)

[2.1. Définition 20](#_Toc197809903)

[2.2. Pourquoi Le Model MVT? 20](#_Toc197809904)

[3. Le formalisme UML: 21](#_Toc197809905)

[4. Diagramme de cas d’utilisation 21](#_Toc197809906)

[5. Diagrammes de classes 23](#_Toc197809907)

[6.Diagramme de séquence 25](#_Toc197809908)

[26](#_Toc197809909)

[Chapitre3 : Outils techniques 26](#_Toc197809910)

[Introduction : 27](#_Toc197809911)

[1. Environnements de développement intégrés : Exploration des outils de développement intégrés pour le projet : 27](#_Toc197809912)

[2. Versionning / Collaboration : Approche de la gestion des versions et de la collaboration dans le projet. 27](#_Toc197809913)

[3. Langage de balisage et de mise en forme WEB : Utilisation des langages de balisage et de mise en forme Web. 28](#_Toc197809914)

[5. Réunions quotidiennes : Évaluation des réunions quotidiennes dans le cadre du projet. 30](#_Toc197809915)

[6. Conclusion 30](#_Toc197809916)

[Chapiter 4: Implémentation 31](#_Toc197809917)

[1.Partie : "register" 32](#_Toc197809918)

[2.Partie : "login" 33](#_Toc197809919)

[2.Partie : "login" 33](#_Toc197809920)

[3.Partie : "Interface principale" 34](#_Toc197809921)

[4.Partie : "les maladies " 35](#_Toc197809922)

[5.Partie : "WIKI" 36](#_Toc197809923)

[5.Partie : " Page d’Évaluation" 37](#_Toc197809924)

[6.Partie : " Page Résultat" 38](#_Toc197809925)

[7.Partie : " Page de Sélection des Maladies Héréditaires" 39](#_Toc197809926)

[Conclusion 39](#_Toc197809927)

[Conclusion Générale 41](#_Toc197809928)

[Webographi 42](#_Toc197809929)

# TABLES DES FIGURES

[Figure 2: Trello 14](#_Toc197680963)

[Figure 3: Tableaux du projet Web Genarator 15](#_Toc197680964)

[Figure 4: Liste trello des réunions 15](#_Toc197680965)

[Figure 5: Liste trello de road-map du projet 16](#_Toc197680966)

[Figure 6: Carte de création du tableau de bord administrateur 17](#_Toc197680967)

[Figure 8: Architecture MVT 20](#_Toc197680968)

[Figure 13: Github 28](#_Toc197680969)

[Figure 20: Google Meet 30](#_Toc197680970)

[Figure 22: Page de connexion 33](#_Toc197680971)

[Figure 23: Page de liste des maladies 34](#_Toc197680972)

# TABLES DES TABLEAUX

[Tableau 1: Fiche technique de la société 9](#_Toc177247025)

[Tableau 2: Comparaison entre les différentes méthodes agiles 12](#_Toc177247026)

# Introduction

Dans le cadre de notre Projet de Fin d’Année, nous avons développé une application web nommée GeneGuard, réalisée en équipe de quatre étudiants. Ce projet a été mené avec l’objectif de proposer un outil numérique accessible permettant d’évaluer le risque pour une personne de développer une maladie héréditaire, sans recourir à des tests génétiques coûteux.

L’application permet à l’utilisateur de sélectionner une maladie parmi une liste prédéfinie, de répondre à un questionnaire médical basé sur ses antécédents familiaux, son mode de vie et d’autres facteurs de risque. Un algorithme d’analyse, basé sur des données médicales, fournit ensuite une estimation personnalisée du risque, accompagnée de recommandations de prévention.

Le rapport est structuré en quatre chapitres principaux :

* **Chapitre 1** : Contexte général du projet — présentant la problématique des maladies héréditaires et les besoins auxquels répond GeneGuard.
* **Chapitre 2** : Analyse et conception — décrivant les méthodes d’analyse, la modélisation UML et les choix fonctionnels.
* **Chapitre 3** : Outils techniques — expliquant les technologies utilisées pour le développement.
* **Chapitre 4** : Implémentation — détaillant les différentes fonctionnalités de l’application, accompagnées de captures d’écran.

Ce projet a été une opportunité de mobiliser nos compétences en développement web, en conception logicielle et en gestion de projet pour répondre à un enjeu réel de santé publique.

# Chapitre1 : Contexte général du projet

## 1. CONDUITE DU PROJET

### 1.1. Cycle de développement :

Lorsqu'il s'agit d'entreprendre un projet informatique, il est primordial de prendre en considération l'impératif d'adopter un cycle de vie de développement qui servira de socle essentiel à la mise en place des différentes solutions envisagées.

### 1.2. Processus de développement :

La sélection inappropriée du processus de développement peut entraîner l'aboutissement infructueux d'un projet. C'est pourquoi nous avons soigneusement élaboré une énumération exhaustive des différentes méthodologies employées au sein des projets informatiques. Le tableau ci-dessous synthétise de manière concise cette étude approfondie :



Tableau 1: Comparaison entre les différentes méthodes agiles

Dans le cadre de ce projet, il est d'une importance primordiale de souligner et de mettre en avant la nécessité absolue de choisir avec soin le cycle de vie de développement qui le sous-tendra. En effet, la réussite même du projet repose en grande partie sur cette décision cruciale. Face à cette réalité incontournable, nous avons consacré un temps considérable à mener une analyse approfondie des différentes méthodologies utilisées dans le domaine des projets informatiques. Cette étude exhaustive a permis d'élaborer une liste non exhaustive des méthodes les plus pertinentes, dans le but de choisir celle qui conviendra le mieux à notre projet spécifique. Après mûre réflexion, nous avons porté notre attention avec une extrême minutie sur la méthode Scrum, qui s'est révélée répondre de manière particulièrement adaptée aux exigences spécifiques de notre projet. Scrum, en tant que méthode agile, présente une caractéristique qui la distingue et qui revêt une importance cruciale : elle est capable de réduire de manière significative le cycle de vie du logiciel tout en accélérant son développement. Elle repose sur une approche itérative qui met l'accent sur la réalisation d'une première version minimale du logiciel, suivie par l'intégration progressive des fonctionnalités grâce à des itérations successives. L'origine de cette méthodologie agile peut être retracée dans un contexte marqué par l'instabilité technologique et l'absence fréquente d'une spécification complète des besoins. Le terme "agile" souligne précisément cette capacité à s'adapter aux changements de contexte et aux évolutions des spécifications tout au long du processus de développement. Au sein de Scrum, des rôles clés sont définis afin d'assurer le bon déroulement du projet. Le "Product Owner" joue un rôle central en tant que représentant du client et est chargé de définir la vision du produit. Le "Scrum Master" occupe quant à lui une position de gestionnaire de projet, veillant à l'application des principes et des pratiques de la méthode agile, tout en facilitant la résolution des problèmes et la répartition équilibrée des tâches parmi les membres de l'équipe. Enfin, l'équipe de développeurs, composée de professionnels hautement qualifiés, est responsable de la concrétisation des fonctionnalités du logiciel et fonctionne de manière autonome en prenant des décisions éclairées. En faisant le choix de Scrum, nous nous engageons avec une détermination sans faille à mener ce projet avec une discipline rigoureuse et une efficacité optimale. Nous sommes pleinement conscients des défis auxquels nous serons confrontés, mais nous nourrissons une confiance inébranlable en notre capacité à relever ces défis grâce à cette méthodologie éprouvée. Notre objectif ultime est de fournir une solution logicielle d'une qualité irréprochable, tout en respectant les délais et les attentes du client, et en favorisant une gestion optimisée des ressources tout en encourageant une innovation continue. En adoptant Scrum comme méthode de référence et en faisant preuve d'un engagement total envers l'excellence, nous sommes résolus à faire de ce projet un exemple de succès retentissant. Nous sommes déterminés à créer une valeur ajoutée significative pour notre client, tout en consolidant notre réputation en tant qu'équipe de développement exceptionnellement performante.

## 2. L'ACCOMPLISSEMENT DU PROJET

### 2.1. Les réunions :

Nous avons été initiés à une série de méthodologies de travail inédites, dont l'inspiration trouve sa source dans les méthodes agiles. Ces approches novatrices ont révolutionné notre vision et notre pratique du travail en équipe. L'aube de chaque semaine de labeur débute par une brève rencontre, d'une durée n'excédant pas quinze minutes. Seuls les membres de l'équipe y sont conviés. Au cours de cette réunion matinale, On interroge tour à tour les membres de l'équipe selon trois interrogations précises :

**Quelles réalisations ont été accomplies la semaine précédente ?**

**Quelles tâches se profilent à l'horizon de cette semaine ?**

**Quels écueils ont-ils été rencontrés au cours de ces évolutions ?**

L'implémentation de cette démarche s'est avérée d'une grande pertinence. Elle confère une clarté inégalée à chaque étape du projet, évitant ainsi que les embûches ne s'amoncellent, les résolvant promptement. De surcroît, cette pratique promeut les échanges constructifs entre les collaborateurs, favorisant ainsi une synergie optimale au sein du collectif. Une autre réunion, d'une importance capitale : la réunion de rétrospective. Elle constitue un moment propice pour recueillir des rétroactions quant aux accomplissements réalisés et pour proposer des améliorations substantielles pour le produit final. En parallèle, les indicateurs de qualité sont soigneusement transmis au Professeur dès l'entame de chaque semaine, marquant le commencement de la phase de développement.es rencontres périodiques, par leur assiduité, nous permettent de maintenir une perspective limpide sur la trajectoire empruntée par le projet, nous octroyant ainsi une flexibilité précieuse pour ajuster notre planification en fonction des résultats obtenus. De manière concomitante, elles favorisent une communication fluide entre toutes les parties prenantes, instaurant une compréhension mutuelle des impératifs et des aspirations. Par le biais de ces réunions rigoureuses, nous nous engageons fermement à garantir une gestion méthodique et transparente du projet. Nous demeurons persuadés que cette approche résolue concourra à l'optimisation de notre efficacité et à l'élévation de la qualité de nos réalisations, tout en renforçant notre aptitude à satisfaire les exigences discernées par le client. Ces réunions, essentielles à l'avancement harmonieux du projet, revêtent une importance incontestable. Elles représentent des occasions propices pour identifier les obstacles potentiels, prendre des décisions éclairées et entretenir une dynamique de travail soutenue. À travers notre attachement indéfectible à ces pratiques agiles, nous affirmons notre confiance quant à la réalisation de nos objectifs et à la pleine satisfaction du client.

### 2.2 Planification des sprints

Préalablement à chaque sprint, lors d'une rencontre de planification appelée "sprint planning meeting", nous opérons une sélection minutieuse des exigences prioritaires pour le client à partir du Product Backlog. Ce dernier recense l'ensemble des objectifs majeurs exprimés sous forme de travaux concrets à réaliser. Les exigences choisies seront développées, testées et livrées au client, constituant ainsi le sprint Backlog, qui représente une sous-section du Product Backlog. Pour cela, nous avons utilisé Trello.



Figure 1: Trello

Trello s'est révélé être un outil indispensable dans la gestion efficace de notre projet, en assurant une visualisation claire et continue de l'avancement de chaque étape. Grâce à ses fonctionnalités intuitives, nous avons pu centraliser toutes les tâches à réaliser, les répartir entre les membres de l'équipe et suivre leur progression en temps réel. Chaque carte nous permet d'y associer des détails spécifiques comme les échéances, les responsables ou encore les sous-tâches, garantissant ainsi que rien ne soit omis.

L'utilisation de Trello a également favorisé une meilleure communication entre les différents acteurs du projet. Les fonctionnalités collaboratives, telles que les commentaires sur les cartes, les notifications et les mises à jour en temps réel, ont grandement amélioré la coordination entre les membres de l’équipe. Nous avons pu partager des informations rapidement et réagir promptement aux retours, ce qui a permis de maintenir un rythme soutenu et de minimiser les délais.

Ainsi, à travers l'utilisation de cet outil agile, nous avons pu structurer nos efforts de manière plus méthodique et fluide, garantissant non seulement l’atteinte de nos objectifs, mais aussi la satisfaction du client. En définitive, Trello nous a permis de simplifier la gestion du projet, tout en nous assurant de maintenir un niveau élevé de qualité dans nos réalisations, grâce à une organisation optimisée et une collaboration renforcée (figure3).

Trello est basé sur un système intuitif de tableaux, listes et cartes :

**tableaux** : représentent des projets ou des espaces de travail

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Figure 2: Tableaux du projet Web Genarator

**Listes** : Correspondent aux différentes étapes d'un projet (par exemple, "À faire", "En cours", "Terminé").

Une image contenant texte, capture d’écran, Publicité en ligne

Description générée automatiquement

Figure 3: Liste trello des réunions

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Figure 4: Liste trello de road-map du projet

**Cartes** : Représentent les tâches spécifiques, sur lesquelles on peut ajouter des descriptions, des dates limites, des checklists, des pièces jointes, des étiquettes, etc.

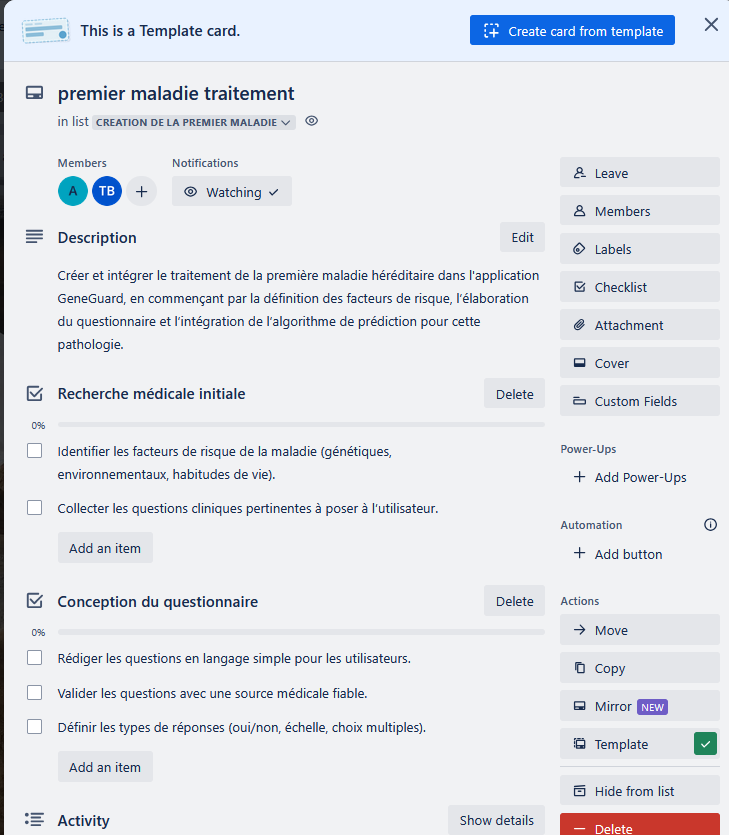


Figure 5: Carte de création du tableau de bord administrateur

## 3. Conclusion :

Ce chapitre marque le point de départ de notre projet. La première moitié de ce chapitre a été consacrée à la présentation détaillée de l'organisme d'accueil, tandis que la seconde partie s'est entièrement concentrée sur la présentation du sujet et de la démarche adoptée pour son développement.

# Chapitre2: Analyse et conception

## 1. Architecture de l’application :

La conception architecturale de mon application se présente sous la forme d'une structure équilibrée, où chaque élément joue un rôle spécifique et contribue au bon fonctionnement de l'ensemble. Au sommet de cette architecture se trouvent les utilisateurs, notamment les administrateurs et les demandeurs, qui interagissent avec l'application à travers une interface utilisateur conviviale. Ils peuvent accéder à toutes les fonctionnalités et soumettre des demandes de ressources selon leurs besoins spécifiques. Ensuite, nous avons le serveur d'applications qui fait office de passerelle entre les utilisateurs et la base de données. Il est responsable de la gestion des requêtes entrantes, de leur traitement et de la récupération des données nécessaires. Le serveur d'applications assure également la sécurité et la confidentialité des informations échangées entre les utilisateurs et la base de données. Enfin, la base de données joue un rôle crucial dans l'application. Elle stocke et organise les données de manière efficace, permettant ainsi une récupération rapide et précise lorsqu'une requête est effectuée. La base de données est conçue pour garantir l'intégrité des données et offrir des fonctionnalités avancées telles que la recherche, la mise à jour et la suppression des informations. Dans l'ensemble, cette architecture met en place une infrastructure robuste et évolutive, favorisant la performance, la sécurité et la fiabilité de l'application. Chaque composant joue un rôle précis et s'articule harmonieusement pour offrir une expérience utilisateur optimale et répondre aux besoins fonctionnels et non fonctionnels de l'application.

## 2. Architecture MVT:

Une image contenant texte, capture d’écran, Police, diagramme

Le contenu généré par l’IA peut être incorrect.

Figure 6: Architecture MVT

### 2.1. Définition

La conception architecturale de notre application repose sur une structure équilibrée suivant le modèle MVT (Modèle–Vue–Template), une architecture adoptée par le framework Django. Ce modèle est dérivé de l’architecture MVC (Modèle–Vue–Contrôleur), avec des adaptations propres à l’environnement Django.Le Modèle représente la couche de données et de logique métier de l'application. Il définit la structure des données via des classes Python (modèles ORM) et assure leur gestion, leur manipulation et leur persistance dans la base de données. Le Modèle est totalement indépendant de l’interface utilisateur, ce qui garantit une séparation nette des responsabilités et facilite la réutilisation. La Vue, dans le contexte Django, joue un rôle similaire au contrôleur de l’architecture MVC. Elle contient la logique métier nécessaire pour répondre aux requêtes utilisateurs, interagit avec les modèles pour accéder ou modifier les données, et renvoie une réponse adaptée (souvent via un Template HTML). Le Template est chargé de la présentation visuelle. Il reçoit les données traitées par la Vue et les affiche à l'utilisateur à travers une interface claire et dynamique. Grâce au système de Template de Django, il est possible de créer des pages web interactives tout en gardant une séparation stricte entre le code métier et l’affichage.

### 2.2. Pourquoi Le Model MVT?

Le choix de l’architecture MVT s’est imposé naturellement dans le cadre de notre projet, compte tenu de son alignement avec le framework Django, mais surtout en raison de ses nombreux avantages structurels. Cette architecture permet une claire séparation des responsabilités entre la logique métier, la gestion des données et la présentation. Le Modèle gère les données de manière centralisée et cohérente. Il permet une abstraction de la base de données, facilitant les interactions sans se soucier des requêtes SQL, ce qui améliore la productivité et la fiabilité du code. La Vue, équivalent logique du contrôleur, permet de traiter chaque requête de manière autonome, en orchestrant l’interaction entre les modèles et les templates. Elle constitue le cœur de la logique de traitement, et peut facilement être testée ou modifiée indépendamment des autres composants. Les Templates, quant à eux, assurent une personnalisation rapide de l’interface utilisateur, en séparant totalement la logique de rendu de la logique applicative. Cela permet aux développeurs et aux designers de travailler en parallèle, tout en conservant une cohérence dans l’apparence et l’expérience utilisateur. L’architecture MVT facilite ainsi le développement modulaire, la réutilisation de composants, et la maintenabilité du projet. Elle permet également une grande évolutivité : de nouveaux modules ou fonctionnalités peuvent être ajoutés sans altérer le reste du système. Grâce à cette séparation claire, chaque membre de l’équipe peut se concentrer sur un aspect précis du projet, optimisant ainsi la collaboration et la qualité du produit final. En résumé, le modèle MVT s’est révélé être un choix stratégique dans la réalisation de notre application GeneGuard, en offrant un équilibre optimal entre organisation, performance, évolutivité et maintenabilité.

## 3. Le formalisme UML:

Notre démarche de conception repose sur l'application rigoureuse des principes de l'Unified Modeling Language (UML), un langage graphique standardisé largement utilisé dans le domaine du génie logiciel pour représenter et documenter les systèmes informatiques. Nous avons suivi une approche méthodique et structurée, en utilisant divers diagrammes UML pour capturer les différents aspects de notre application et garantir une conception solide et cohérente.

Analyse des besoins : Nous avons effectué une analyse approfondie des besoins fonctionnels et non fonctionnels de notre application. Cette étape nous a permis de comprendre les objectifs des utilisateurs, leurs attentes ainsi que les contraintes et exigences auxquelles notre système doit répondre.

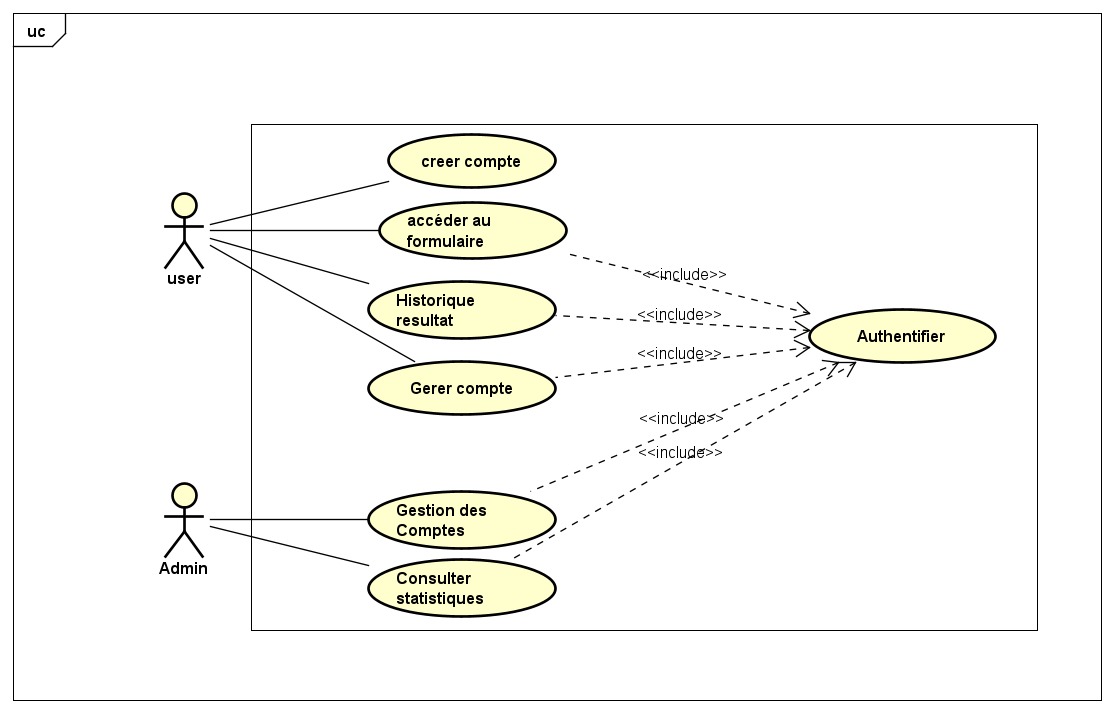
Modélisation conceptuelle : À partir des résultats de l'analyse des besoins, nous avons utilisé des diagrammes de cas d'utilisation pour représenter les différentes fonctionnalités de notre application et les interactions entre les utilisateurs et le système. Ces diagrammes nous ont aidé à identifier les acteurs, à définir les scénarios clés et à clarifier les flux d'exécution, ce qui nous a permis de définir les principales fonctionnalités de mon système.

Modélisation structurelle : nous utilisons des diagrammes de classes pour décrire la structure interne de notre application. Ces diagrammes nous permettent d'identifier les classes, les relations entre les classes, les attributs, les méthodes, les interfaces et les packages. Cette modélisation nous aide à organiser notre code de manière cohérente et à visualiser les dépendances et les interactions entre les différents composants de notre système.

## 4. Diagramme de cas d’utilisation

Un cas d'utilisation est une description des actions et des interactions entre les acteurs et le système, visant à atteindre un résultat spécifique et observable. Il permet de regrouper les intentions fonctionnelles des acteurs en unités logiques, en se focalisant sur les messages échangés avec le système. Ainsi, en identifiant et en regroupant ces intentions fonctionnelles de manière cohérente, nous créons des cas d'utilisation qui capturent les besoins et les objectifs des acteurs dans le contexte de notre système.

Dans ce système, le User (utilisateur standard) joue un rôle central dans l’interaction avec l’application. Il peut créer un compte, accéder à un formulaire pour soumettre ou consulter des données, consulter l’historique de ses résultats, et gérer son propre compte. Toutes ces actions nécessitent une authentification préalable, assurant ainsi la sécurité et la personnalisation de l’expérience utilisateur. L'utilisateur est principalement concerné par l'utilisation fonctionnelle du système pour ses besoins personnels, comme l'accès à ses informations ou la gestion de son profil. Quant à l’Admin (administrateur), son rôle est davantage orienté vers la supervision et la gestion globale du système. Il peut accéder à la gestion des comptes, ce qui implique des droits supplémentaires comme la modification, l’activation ou la suppression de comptes utilisateurs. L’administrateur peut également consulter des statistiques, fournissant une vision globale et analytique du système, utile pour la prise de décisions. Comme pour l'utilisateur, toutes les actions de l'admin sont protégées par une authentification, garantissant que seules les personnes autorisées peuvent accéder à ces fonctionnalités critiques.

*Figure 9:* *Diagramme de cas d’utilisation*

## 5. Diagrammes de classes

Le diagramme de classes offre une représentation statique du système en mettant en évidence les classes et les relations qui existent entre elles. Il permet d'identifier les attributs qui définissent les propriétés des objets et les méthodes qui représentent les services offerts par le système. En visualisant les classes et leurs associations, le diagramme de classes permet de comprendre la structure globale du système, de définir les responsabilités de chaque classe et d'analyser les dépendances entre les différentes parties du système. Il est donc un outil essentiel pour la modélisation et la conception orientée objet.

1. USER

Cette classe représente un utilisateur final du système. Chaque utilisateur est associé à un seul compte (relation 1-1). Elle contient les informations d'identification de base de l’utilisateur.

2. ADMIN

La classe Admin représente un administrateur qui a un accès privilégié au système. Un administrateur peut gérer plusieurs comptes utilisateurs. Elle contient des attributs comme :

nom : le nom de l’administrateur,

MotDePass : le mot de passe de connexion.

3. COMPTE

Cette classe représente le compte utilisateur contenant les informations nécessaires pour s’authentifier et interagir avec le système :

id, nom, email, MotDePass.

Chaque compte peut être associé à plusieurs maladies, à un historique de résultats, et à un user unique. Il est géré par un admin.

4. MALADIE

Représente une maladie que le système peut diagnostiquer ou traiter. Chaque maladie est :

liée à un compte (créée ou suivie par un utilisateur),

associée à un formulaire pour la détection ou l’analyse.

Elle contient : id, nom, description, idCompte.

5. FORMULAIRE

Un formulaire est lié à une seule maladie. Il contient :

une liste de questions à poser,

une liste de réponses possibles.

Chaque formulaire a un identifiant unique et est le point de départ pour générer une réponse.

6. REPONSE

Cette classe représente les résultats fournis par un utilisateur à un formulaire donné. Elle contient :

un pourcentage de risque (pourcentageDerisque),

l’identifiant du formulaire concerné.

Une réponse peut être liée à plusieurs historiques de résultats.

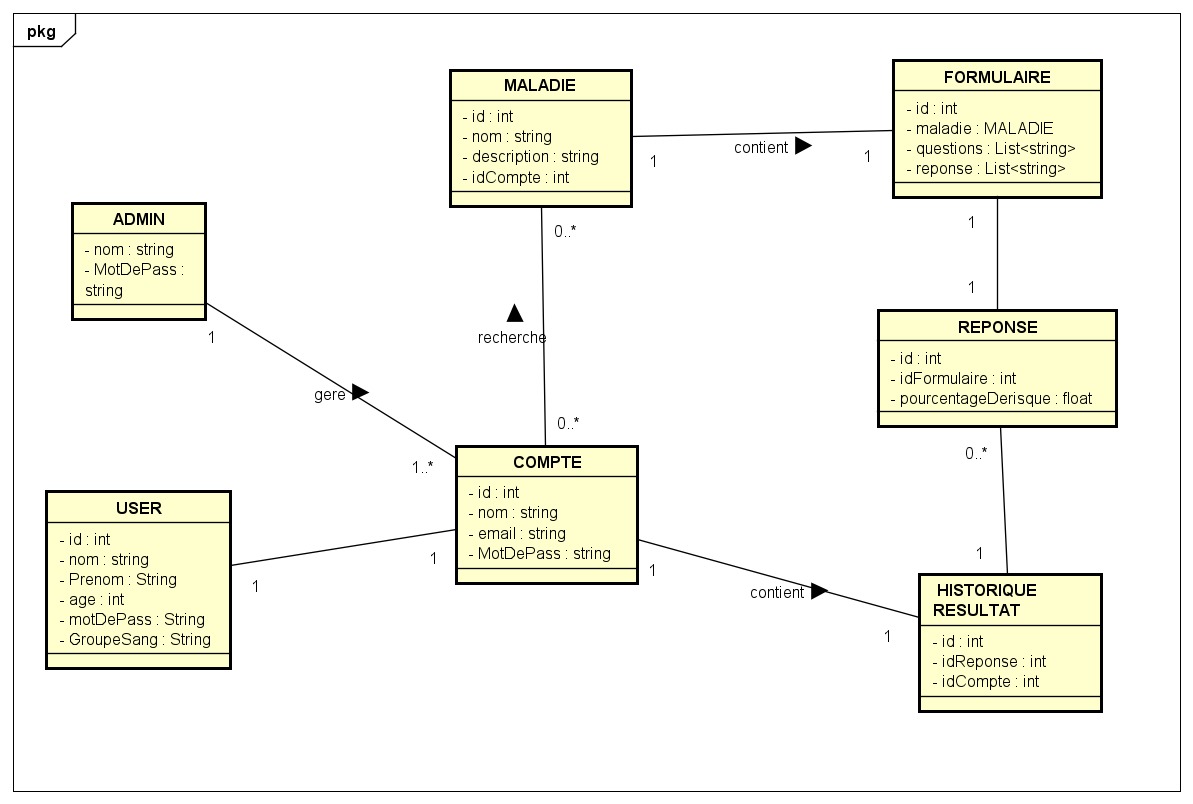
7. HISTORIQUE\_RESULTAT

Permet de tracer les réponses précédemment fournies par les utilisateurs. Elle associe :

un idReponse,

un idCompte.

Chaque historique correspond à un enregistrement des résultats d’un utilisateur pour un formulaire donné.

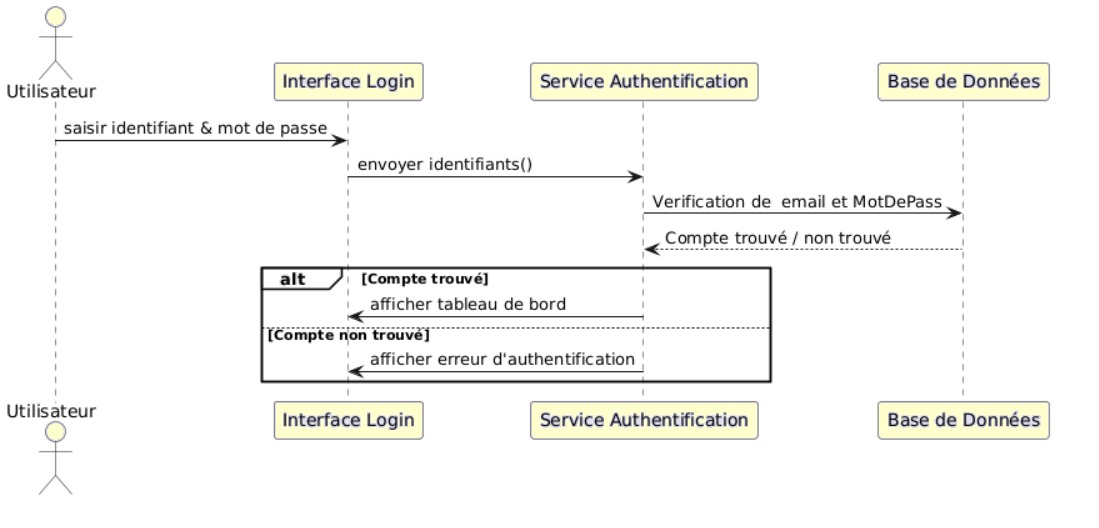


*Figure 10 :Diagramme de classe*

## 6.Diagramme de séquence

Les diagrammes de séquence que j'ai créés illustrent certains scénarios spécifiques d'exécution des cas d'utilisation, me permettant ainsi de mieux visualiser les interactions entre les acteurs et le système.

Ce diagramme de séquence illustre le processus d’authentification d’un utilisateur (qu’il soit User ou Admin) dans le système. Tout commence lorsque l’utilisateur saisit ses identifiants (email et mot de passe) via l’interface de connexion. Ces informations sont ensuite transmises au service d’authentification, qui agit comme intermédiaire entre l’interface et la base de données. Le service interroge la base de données pour vérifier si un compte correspondant aux identifiants saisis existe. Si un compte valide est trouvé, l’utilisateur est redirigé vers son tableau de bord ou son espace personnel. En revanche, si les identifiants sont incorrects ou inexistants, le système affiche un message d’erreur signalant un échec d’authentification. Ce scénario garantit une vérification sécurisée des accès au système, en respectant les règles définies dans le modèle de classes.

 *Figure 11 : diagramme de séquence*

# 

# Chapitre3 : Outils techniques

## Introduction :

Dans cette section, nous abordons l'approche technique que nous avons adoptée et présentons les différents frameworks utilisés. Nous avons utilisé un ensemble d'outils technologiques que nous détaillons ci-dessous :

## Environnements de développement intégrés : Exploration des outils de développement intégrés pour le projet :

**Visual Studio Code (VSCode)** est un éditeur de code source léger et puissant, largement utilisé par les développeurs pour sa flexibilité et ses nombreuses fonctionnalités. Il prend en charge une large gamme de langages de programmation, dont Python, et offre une riche collection d'extensions permettant d'adapter l'environnement de développement aux besoins spécifiques de chaque projet. Dans le cadre de l'application de prédiction des maladies, VSCode a été un choix idéal grâce à ses outils intégrés de débogage, son terminal intégré, ainsi que la prise en charge de Git pour une gestion efficace du versioning. Les extensions telles que Pylance, Python, et Jupyter ont facilité le développement et l'exécution du code, offrant un environnement de travail optimisé pour l'intelligence artificielle et l'analyse des données. Son interface simple, combinée à des fonctionnalités de personnalisation avancées, a permis de maintenir une productivité élevée tout au long du projet

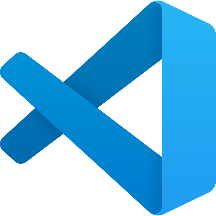


Figure 12 : visualstudio

## 2. Versionning / Collaboration : Approche de la gestion des versions et de la collaboration dans le projet.

**GitHub** : est une plateforme basée sur Git qui offre un espace de stockage en ligne pour les projets Git. Cela permet aux développeurs de partager leur code avec d'autres membres de l'équipe ou de la communauté, de suivre les problèmes et de gérer les demandes de fusion. GitHub offre également des fonctionnalités sociales, telles que la possibilité de suivre les projets d'autres personnes, de collaborer sur des projets open source et de contribuer à la communauté du développement logiciel.

Une image contenant noir, obscurité

Description générée automatiquement

Figure 7: Github

Git : est un système de contrôle de version qui permet de suivre les modifications apportées au code source d'un projet. Il permet de créer des instantanés du code à différents stades de développement, de gérer les branches de travail et de fusionner les modifications entre les différentes versions. Git facilite la collaboration en permettant à plusieurs personnes de travailler sur le même projet et de fusionner leurs contributions de manière fluide.

Une image contenant Graphique, Police, Panneau de signalisation, symbole

Description générée automatiquement

Figure 14: Git

## 3. Langage de balisage et de mise en forme WEB : Utilisation des langages de balisage et de mise en forme Web.

JavaScript : est un langage de programmation utilisé pour rendre les sites web interactifs et dynamiques. Il permet d'ajouter des fonctionnalités avancées, de manipuler et modifier le contenu des pages, de valider les formulaires, de créer des animations et d'interagir avec les utilisateurs en temps réel.

Une image contenant jaune, symbole, Rectangle, Graphique

Description générée automatiquement

Figure 15: JavaScript

HTML : est le langage de balisage utilisé pour structurer et organiser le contenu des pages web. Il définit la structure des éléments tels que les titres, les paragraphes, les listes, les images et les liens. HTML permet de créer une mise en page cohérente et de présenter le contenu de manière structurée.

Une image contenant Graphique, graphisme, rouge, conception

Description générée automatiquement

Figure 16: Html

CSS : est un langage utilisé pour définir l'apparence visuelle des pages web. Il permet de styliser les éléments HTML en spécifiant les couleurs, les polices, les marges, les espacements, les bordures et autres propriétés visuelles. CSS offre une grande flexibilité pour créer des designs attrayants et adaptés à différents appareils.

Une image contenant Graphique, capture d’écran, Rectangle, conception

Description générée automatiquement

Figure 17: Css

Bootstrap : est un framework CSS populaire qui facilite la création de sites web réactifs et esthétiques. Il fournit des composants prédéfinis et des classes CSS pour créer rapidement des mises en page réactives, des menus déroulants, des modales, des carrousels et d'autres éléments d'interface utilisateur interactifs. Bootstrap simplifie le processus de conception en offrant des styles et des fonctionnalités prêts à l'emploi, ce qui permet d'économiser du temps et des efforts de développement.

Une image contenant Graphique, symbole, graphisme, Police

Description générée automatiquement

Figure 18: Bootstrap

4.Technologies et Frameworks : Présentation des technologies et des frameworks utilisés dans le projet.

Django est un framework web open-source écrit en Python, conçu pour faciliter le développement rapide d'applications web robustes et sécurisées. Il repose sur le principe "DRY" (Don't Repeat Yourself), permettant de réduire la duplication du code et d'encourager une architecture propre et modulaire. Django est particulièrement apprécié pour sa gestion intégrée des bases de données, son système d'administration automatique, ainsi que ses fonctionnalités de sécurité avancées, telles que la gestion des sessions, la protection contre les attaques CSRF et la gestion des mots de passe cryptés.



Figure 19: Django

## 5. Réunions quotidiennes : Évaluation des réunions quotidiennes dans le cadre du projet.

Pour faciliter la communication et les échanges d'informations au sein du projet, nous avons choisi d'utiliser Google Meet pour nos réunions quotidiennes. Google Meet nous a offert une plateforme pratique et conviviale pour se réunir en ligne, peu importe notre emplacement géographique. Ces réunions quotidiennes nous ont permis de partager les mises à jour sur l'avancement du projet, de discuter des problèmes rencontrés, de prendre des décisions importantes et de coordonner nos actions. Grâce à Google Meet, nous avons pu maintenir une communication régulière et efficace entre les membres de l'équipe, favorisant ainsi la collaboration et le bon déroulement du projet.

Une image contenant Graphique, Caractère coloré, capture d’écran, graphisme

Description générée automatiquement

Figure 8: Google Meet

## 6. Conclusion

Ce chapitre a présenté les outils techniques utilisés lors du développement de notre application dans le cadre de notre stage. Le chapitre suivant consiste en une série de captures d'écran illustrant la partie spécifique que nous avons développée.

# Chapiter 4: Implémentation

## 1.Partie : "register"

La section d'inscription de notre plateforme offre aux utilisateurs un formulaire convivial pour créer un compte, avec des instructions claires et des fonctionnalités supplémentaires telles que la connexion et la gestion des erreurs comme le présente la figure 19.

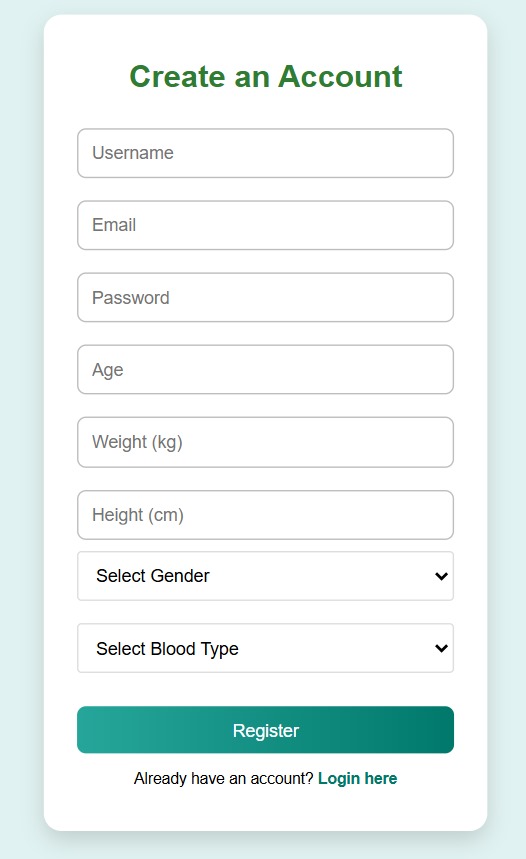


Figure 21: Page d'inscription

## 2.Partie : "login"

La section de connexion de votre page web offre aux utilisateurs un formulaire convivial pour se connecter à leur compte, avec des instructions claires et des fonctionnalités supplémentaires telle que l'inscription.



Figure 9: Page de connexion

## 2.Partie : "login"

## 3.Partie : "Interface principale"

Une image contenant texte, capture d’écran, diagramme

Le contenu généré par l’IA peut être incorrect.

Figure 10 : page principale

L'image ci-dessus présente la **page d'accueil** de l’application. Son interface est **claire, épurée et conviviale**, conçue pour offrir une navigation fluide et accessible à tous les utilisateurs.

En haut de la page, une **barre de navigation horizontale** permet d'accéder rapidement aux différentes sections :

* **Home**
* **Formulaire**
* **Résultat**
* **Wiki**
* **Se connecter**

Sous cette barre, l’utilisateur découvre **quatre grandes cartes interactives**, représentant les principales fonctionnalités de l’application :

* **Évaluer les maladies** : Lancement d’une évaluation personnalisée via un formulaire médical.
* **Estimer les risques** : Analyse des données génétiques et personnelles pour générer des probabilités.
* **Formulaire rapide** : Accès à un mode simplifié pour une estimation express.
* **Probabilités exactes** : Consultation de résultats détaillés et chiffrés sur les risques potentiels.

Chaque carte est accompagnée d’une **icône visuelle intuitive**, renforçant l’aspect dynamique et engageant de l’interface.

## 4.Partie : "les maladies "

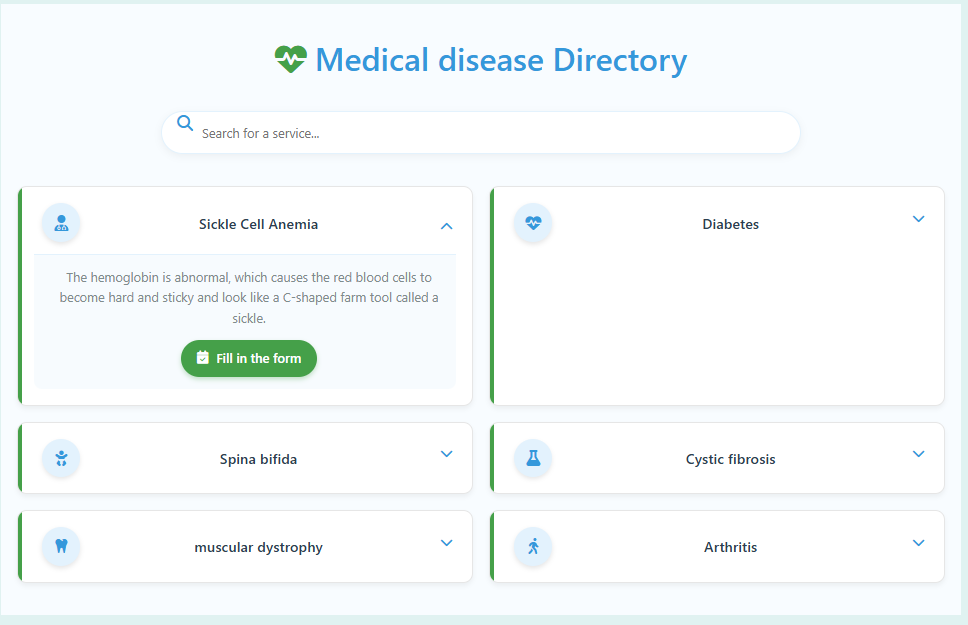


Figure 11: Page de liste des maladies

Cette image représente l’interface d’un **annuaire médical interactif** intitulé Medical Disease Directory. Conçue avec une esthétique épurée et des couleurs douces, cette interface vise à faciliter l’accès à des informations sur diverses maladies. En haut, on retrouve un champ de recherche intuitif permettant aux utilisateurs de rechercher rapidement un service ou une pathologie spécifique. Juste en dessous, l’interface affiche une grille de six cartes, chacune représentant une maladie bien connue. Les maladies listées sont : **Sickle Cell Anemia (anémie falciforme)**, **Diabetes (diabète)**, **Spina bifida**, **Cystic fibrosis (mucoviscidose)**, **Muscular dystrophy (dystrophie musculaire)** et **Arthritis (arthrite)**. Chaque carte comporte une icône bleue symbolisant la maladie, son nom, et une flèche vers le bas suggérant qu’un clic permettra d’en savoir plus. Cet outil est manifestement conçu pour offrir aux patients, professionnels de santé ou chercheurs une navigation fluide et un accès rapide aux données médicales essentielles.

Un bouton vert “**Fill in the form**” est également présent, suggérant la possibilité pour l’utilisateur de soumettre des informations

## 5.Partie : "WIKI"

Une image contenant texte, capture d’écran, logiciel, Page web

Le contenu généré par l’IA peut être incorrect.

Figure 12 : page « WIKI »

La page "Wiki" de l’application **Gueneguard** constitue une **véritable base de connaissances médicales** dédiée aux **maladies héréditaires**. Elle a été conçue dans le but de fournir à chaque utilisateur une **information claire, fiable et vulgarisée**, facilitant la **compréhension des affections génétiques** et de leurs implications.

Accessible à tous, ce module s'adresse aussi bien aux personnes curieuses, qu'à celles ayant des antécédents familiaux ou simplement désireuses d’approfondir leurs connaissances à des fins personnelles, éducatives ou préventives.

L’interface est organisée sous forme de **fiches détaillées**, chacune portant sur une maladie héréditaire spécifique. Par exemple, dans la capture d’écran fournie, on peut voir la fiche consacrée à l’**Anémie Falciforme (Sickle Cell Anemia)**. Chaque fiche est structurée de manière pédagogique avec les sections suivantes :

* **Brève description** de la maladie, expliquant ses origines génétiques et son impact sur la santé.
* **Symptômes courants**, permettant aux utilisateurs de reconnaître les signes les plus fréquents.
* **Causes principales**, souvent liées à des mutations génétiques transmises de génération en génération.
* **Traitements disponibles**, incluant les approches médicales actuelles, les thérapies de soutien et les conseils de suivi.

Ce module favorise l’**éducation à la santé** et l’**autonomie des utilisateurs**, en leur donnant les outils nécessaires pour mieux comprendre leur situation ou celle de leurs proches, tout en encourageant une démarche proactive de prévention.

## 5.Partie : " Page d’Évaluation"

Une image contenant texte, Police, nombre, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Figure 13 :page d’évaluation

Une fois que l’utilisateur sélectionne la maladie héréditaire qu’il souhaite explorer, il est redirigé vers une **page d’évaluation interactive**  dédiée à cette pathologie. Cette page présente un **questionnaire médical personnalisé**, conçu pour recueillir des informations essentielles permettant d’estimer le **risque potentiel de développer la maladie choisie**.

Par exemple, dans la capture d’écran fournie, on voit le **formulaire spécifique au diabète**. Ce dernier contient une série de questions portant sur différents aspects de la santé, du mode de vie et des antécédents familiaux de l’utilisateur

L’objectif de ce questionnaire est de collecter suffisamment de données pour permettre à l’algorithme de Gueneguard de **calculer une estimation personnalisée du risque**, exprimée ensuite en pourcentage clair et visuel.

Ce module joue un rôle central dans la mission de prévention de l’application, car il encourage l’utilisateur à prendre conscience de ses **facteurs de risque individuels**. En complément, des **recommandations adaptées** sont automatiquement générées à la fin du test, offrant des conseils pratiques pour améliorer sa santé et **réduire les probabilités de développer la maladie**.

Cette approche permet de **combiner sensibilisation, auto-évaluation et prévention**, tout en proposant une expérience fluide et accessible, même pour les utilisateurs non familiers avec le domaine médical.

## 6.Partie : " Page Résultat"

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Le contenu généré par l’IA peut être incorrect.

Figure 14 : page résultat

Une fois le **questionnaire complété**, l'utilisateur est redirigé vers la **page de résultats**, qui affiche une **estimation claire et visuelle du risque** d’être atteint par la maladie héréditaire sélectionnée. Ce score, exprimé en pourcentage, est présenté sous forme de graphique circulaire coloré pour une meilleure lisibilité.

Mais cette page ne se limite pas à une simple donnée chiffrée. Elle propose également :

* **Des recommandations personnalisées** basées sur les réponses fournies par l’utilisateur, comme par exemple :
  + Adopter une alimentation équilibrée
  + Pratiquer une activité physique régulière
  + Planifier des bilans de santé périodiques
* **Une description synthétique de la maladie** pour aider l’utilisateur à mieux comprendre de quoi il s’agit, ses causes, ses symptômes et les traitements existants. Cela permet de **démystifier la pathologie** tout en informant de manière accessible et pédagogique.

L’objectif principal de cette page est de **sensibiliser l’utilisateur**, de l’accompagner dans une démarche de prévention, et de lui offrir des **conseils adaptés à sa situation personnelle**, sans jugement, tout en respectant la confidentialité de ses données.

Cette approche rend l’expérience utilisateur à la fois informative, interactive et centrée sur le bien-être.

## 7.Partie : " Page de Sélection des Maladies Héréditaires"

Une image contenant texte, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Figure 15 : page de sélection des maladies héréditaires

Cette page constitue le **point de départ essentiel** pour les utilisateurs souhaitant entamer une évaluation ou en apprendre davantage sur une maladie héréditaire spécifique. Elle se présente comme une **interface claire, intuitive et conviviale**, facilitant la navigation et le choix.

Sous la barre de recherche, s'affiche une **liste organisée de maladies génétiques**. L'affichage est simple mais efficace, permettant un accès rapide à l’information ou aux tests associés. Voici quelques exemples :

* **Sickle Cell Anemia (Anémie Falciforme)**
* **Diabetes (Diabète)**
* **Spina Bifida (Spina Bifida)**
* **Cystic Fibrosis (Mucoviscidose)**
* **Muscular Dystrophy (Dystrophie Musculaire)**
* **Arthritis (Arthrite)**

Chaque élément de la liste agit comme un **bouton cliquable** menant l’utilisateur vers un **formulaire spécifique ou une fiche détaillée**, en fonction de son choix.

Cette page joue un rôle central dans l’expérience utilisateur en **simplifiant l'accès aux services proposés** par la plateforme Gueneguard, qu’il s’agisse d’effectuer un test d’évaluation ou d’accéder à des contenus éducatifs sur les maladies héréditaires. Elle contribue à rendre la prévention et la sensibilisation plus accessibles, même aux non-initiés.

## 

## Conclusion

Cette partie vise à présenter la phase de réalisation finale du projet, où nous avons détaillé les outils et l'environnement de développement utilisés. Nous avons également décrit l'architecture de l'application adoptée. En plus de cela, nous avons présenté quelques captures d'écran de l'application ainsi que des rapports générés par le système. L'objectif est de donner une vision concrète de l'application et de ses fonctionnalités dans cette phase de réalisation.

# Conclusion Générale

Le présent rapport retrace de manière détaillée le déroulement de notre Projet de Fin d’Études, centré sur la conception et le développement de **GeneGuard**, une plateforme web interactive de **prédiction du risque de maladies héréditaires**, réalisée à l’aide des technologies modernes telles que **Django** et **Visual Studio Code**.

Tout au long des différents chapitres, nous avons analysé les aspects clés du projet, en mettant en évidence les choix techniques, les solutions apportées et les résultats obtenus.

Le **chapitre I** présente le **contexte général** du projet, soulignant l’importance croissante de la prévention dans le domaine de la santé, en particulier pour les maladies héréditaires. L’objectif principal de GeneGuard est de fournir un **outil simple, accessible et informatif** permettant aux utilisateurs d’évaluer leur risque sans avoir recours à des tests médicaux coûteux.

Dans le **chapitre II**, une **étude approfondie des besoins fonctionnels et techniques** a été réalisée. Cette phase a permis d’identifier les attentes des utilisateurs, de définir les fonctionnalités essentielles de la plateforme (sélection de maladies, questionnaires dynamiques, retour de résultats personnalisés), et de concevoir une solution alignée sur ces exigences.

Le **chapitre III** est consacré à la **conception et à l’architecture technique**. Nous avons adopté l’architecture MVT propre à Django, structuré les modèles de données avec l’ORM intégré, et mis en place les vues et templates nécessaires au bon fonctionnement du système. L’environnement de développement s’est appuyé sur **VS Code**, facilitant ainsi la collaboration et le suivi du projet.

Enfin, le **chapitre IV** présente les **résultats obtenus**, incluant la démonstration des fonctionnalités clés de l’application : génération du score de risque, affichage de recommandations, interface fluide et responsive. Des **tests fonctionnels et unitaires** ont été menés pour garantir la robustesse de l’application et assurer une expérience utilisateur de qualité.

En conclusion, **ce projet a été une aventure formatrice**, nous permettant de renforcer nos compétences en développement web avec Django, en gestion de projet agile (Scrum), et en réflexion autour de problématiques de santé publique. Nous sommes fiers du résultat obtenu, qui reflète notre engagement, notre rigueur et notre volonté de proposer une solution utile et innovante.

# Webographi

[**https://www.tuleap.org/fr/agile/comprendre-methode-agile-scrum-10-minutes**](https://www.tuleap.org/fr/agile/comprendre-methode-agile-scrum-10-minutes)

[**https://hkrtrainings.com/mvt-tutorial**](https://hkrtrainings.com/mvt-tutorial)

[**https://en.wikipedia.org/wiki/Unified\_Modeling\_Language**](https://en.wikipedia.org/wiki/Unified_Modeling_Language)