République Islamique de Mauritanie
Ministre de l'Enseignement Supérieur et de la
Recherche Scientifique

MÉMOIRE DE STAGE DE FIN D'ÉTUDES

Présenté en vue de l'obtention du
Diplôme de Master Professionnel en Informatique
Appliquée à la Gestion (MPIAG)

Par :

Mohamed Lemine Salem M'bedah (IE18689)

Thème

CREATION D'UN SYSTEME DE GESTION DE RELATION CLIENT (CRM) ET DE KYC (KNOW YOUR CUSTOMER)

Encadré par :

Dr. Cheikh Dhib

Mr. Mboirick Mohamed

Réalisé au sein de CADORIM

CADORIM
YOUR WAY TO SEND MONEY

Année universitaire 2021-2022

DEDICACES

Les mots ne sauront traduire ce qui est dans le cœur, mais je vais rassembler mon vocabulaire et dire : je dédie cet humble travail à mes chers parents pour leur soutien tout au long de mon parcours universitaire, qui ont été les encouragements dans les moments de détresse, l'exhortation en période de prospérité, le soutien lors de la chute et le guide lors de la montée, qui n'a pas hésité un instant à m'aider jusqu'à ce que j'ai arrivé là où je suis maintenant. Tous les remerciements et gratitudes ne remplissent pas leur droit.

REMERCIEMENTS

Ce n'est pas parce que la tradition l'exige que cette page se trouve dans ce rapport, mais par ce que les gens à qui s'adressent mes remerciements les méritent vraiment.
J'adresse mes remerciements les plus chaleureux envers :

- ***Dr. Cheikh Dhib,*** *l'ancien coordinateur de MPIAG, ainsi que* ***Dr. Emani Mohamed Sidi*** *le nouveau coordinateur et l'ensemble du corps professoral et administratif de l'ISCAE. Et pour avoir suivi et dirigé l'évolution de mon travail, ainsi que les encouragements, conseils et orientations réguliers qu'il m'a prodigué. Je vous suis infiniment reconnaissant pour votre grande disponibilité et l'intérêt que vous avez porté à ce travail ;*
- ***Mr. Mboirick Mohamed,*** *responsable technique et mon maître de stage, pour sa disponibilité, sa gentillesse, son attention particulière à l'égard de ce travail. Et pour avoir mis les moyens nécessaires au bon déroulement de ce stage de six mois et à la réalisation de ce travail ; Je vous suis infiniment reconnaissant pour votre grande disponibilité et l'intérêt que vous avez porté à ce travail ;*

Toutes les personnes qui de près ou de loin, ont contribué à la réalisation de ce travail.

AVANT-PROPOS

L'Institut Supérieur de Comptabilité et d'Administration des Entreprises est un établissement public d'enseignement supérieur et de recherche, placé sous la tutelle du Ministère en charge de l'Enseignement Supérieur, crée en 2009, par décret N° 2009-161 du 29 avril 2009.

L'ISCAE compte deux (2) départements et dans le cadre de leur formation, les étudiants qui sont en fin de cycle sont tenus d'effectuer un stage pratique au sein d'une entreprise ou d'un service informatique.

RESUME

Ce document présente les travails suivants :

1. Mise en place d'un système d'extration des donnees à partir des images et traitement des ces donnees.
2. Elaboration d'un système de verificitation des donners KYC pour lutter contre les fraudes a la carte bancaire.
3. Tracking de l'activite d'un client sur une application.
4. Mise en place d'un système de messagerie instantané (Live chat).
5. Système pour suivre les demandes de clients, de la soumission juqu'au traitement et l'archivage.
6. Mise en place d'un système de repporting et de monitoring.
7. Mise en place d'un système de notification automatique.
8. La documention.

LISTE DES ABRÉVIATIONS

Je présente ici certains sigles et abréviations que nous utiliserons dans le document.

ISCAE : Institut Supérieur de Comptabilité et d'Administration des Entreprises

MPIAG : Master Professionnel en Informatique Appliquée à la Gestion

API : Application Programming Interface

MVC : Modèle Vue Contrôleur

HTTP : Hypertext Transfer Protocol

HTTPS : HyperText Transfer Protocol Secure

UML : Unified Modeling Language

SQL : Structured Query Language

JSON : JavaScript Object Notation

XP : eXtreme Programming

IIS : Internet Information Services

DAO : Data Access Object

AAB : Android App Bundles

APK : Android Package Kit

IDE : Integrated Development Environment (Environnement de Développement Intégré)

MERISE : Méthode d'Étude et de Réalisation Informatique par les Sous-Ensembles ou pour les Systèmes d'Entreprise

OPT : One-Time Password

SGBD : Système de Gestion de Bases de Données

SGBDR : Système de Gestion de Bases de Données Relationnelle

SOA : Service Oriented Architecture

OCR : Optical Character Recognition

MRZ : Machine-Readable Zone

Table des matières

Chapitre 1

Introduction générale

Savoir qui est votre client et adopter des protocoles pour prévenir la criminalité financière sont des défis permanents pour les institutions financières. De manière significative, les institutions financières (y compris les banques, les coopératives de crédit et les sociétés financières du Fortune 50) doivent se conformer à un ensemble de réglementations de plus en plus complexes pour la vérification de l'identité des clients appelée KYC.

KYC, également connu sous le nom de "Know Your Customer" ou "Know Your Client", est un ensemble de procédures permettant de vérifier l'identité d'un client avant ou pendant les transactions avec les banques et autres institutions financières. Le respect des réglementations KYC peut aider à tenir à distance le blanchiment d'argent, le financement du terrorisme et d'autres stratagèmes de fraude courants. En vérifiant d'abord l'identité et les intentions d'un client au moment de l'ouverture du compte, puis en comprenant ses habitudes de transaction, les institutions financières sont en mesure d'identifier plus précisément les activités suspectes.

Les institutions financières sont soumises à des normes de plus en plus strictes en matière de lois KYC. Ils doivent dépenser plus d'argent pour se conformer à KYC ou être passibles de lourdes amendes. Ces réglementations signifient que presque toutes les entreprises, plateformes ou organisations qui interagissent avec une institution financière pour ouvrir un compte ou effectuer des transactions devront se conformer à ces obligations.

La gestion de la relation client (CRM) est la combinaison de pratiques, de stratégies et de technologies que les entreprises utilisent pour gérer et analyser les interactions et les données client tout au long du cycle de vie du client. L'objectif est d'améliorer les relations de service client, de contribuer à la fidélisation de la clientèle et de stimuler la croissance

des ventes. Les systèmes CRM compilent les données client à travers différents canaux, ou points de contact, entre le client et l'entreprise, qui peuvent inclure le site Web de l'entreprise, le téléphone, le chat en direct, le publipostage, les supports marketing et les réseaux sociaux. Les systèmes CRM peuvent également donner aux membres du personnel en contact avec les clients des informations détaillées sur les informations personnelles des clients, l'historique des achats, les préférences et les préoccupations d'achat.

1.1 Motivations

KYC est un moyen de rendre la vérification de l'identité des clients plus précise et moins vulnérable à la fraude.

KYC doivent être effectuées lors de l'intégration d'un nouveau client, mais il est préférable de répéter ces vérifications de temps en temps, pour s'assurer que tout est comme il se doit. En surveillant les comptes clients de cette manière, les comportements suspects peuvent être signalés plus rapidement.

Un système CRM fournit des flux de travail automatisés qui permettent à votre équipe marketing de consacrer plus de temps à des tâches stratégiques, telles que la création de campagnes marketing qui résonnent, l'analyse des données de ces campagnes et le test de différentes approches basées sur ces analyses. Les agents du service client peuvent passer leur temps à travailler avec des clients qui ont des questions, des problèmes ou des besoins plus complexes. En bref, avec des processus de service client plus efficaces, les entreprises peuvent établir de meilleures relations avec leurs clients.

1.2 Problématiques

En réalité, la réalisation d'une application,qui applique le principe de KYC et integre un système CRM, nécessite de faire face à des problématiques diverses et complexes. Ainsi, la société a décidé de se contenter, dans un premier temps, Mise en place d'un système d'extration des donnees à partir des images et traitement des ces donnees(carte d'identité ou passeport). Ce sujet soulève de nombreuses questions aux implications différentes. Comment peut extraire le texte apartir de l'image ? Comment sera-t-il traité ? Comment peut-il être utilisé dans le principe KYC ? Comment pouvons-nous obtenir un système CRM intégré ?

1.3 Objectifs

La mise en place d'une application pour appliquer l'ide de KYC en basant sur les différent technologie disponible . En basan sur l'extraction du text apartir d'une imange OCR on peut extracter la code MRZ apartir d'une imange du piece d'idendite ou passport est passe le code a un algorithem qui permer de d'etecter les information personnel.

Chapitre 2

Environnement de travail

L'objectif de ce chapitre consiste à expliciter tous les éléments relatifs à l'environnement logiciel sur lequel j'ai travaillé pour réaliser cette application. Il s'agit donc des logiciels, langages, frameworks, et motifs d'architecture auxquels j'ai fait recours tout au long du processus de mise en œuvre de ma solution.

2.1 Gestion du projet

Dans cette section, je présente la méthodologie de travail et le logiciel de gestion de projet que j'ai utilisé pour déclencher, organiser et gérer le travail sur ma solution.

2.1.1 Méthodologie de travail

C'est la manière de mener un processus de développement. Il s'agit d'une démarche, un ensemble d'étapes ou procédures à mettre en œuvre dans une logique méthodologique, accompagnés d'outils et de techniques. L'utilisation d'une méthode est incontournable dans l'entreprise de tout projet, particulièrement dans la réalisation de projets informatiques. Dans ce cas-ci, les méthodes utilisées sont des méthodes d'analyse et de conception qui ont pour but la formalisation des étapes préliminaires au développement de systèmes logiciels, en d'autres termes : analyse, modélisation et conception.
L'urgence de l'utilisation de ces méthodes trouve son explication dans un certain nombre de facteurs :

1. De nombreux échecs de projets informatiques dans le passé dus à un manque d'organisation, ou une non satisfaction des besoins ;

2. La révolution de l'industrie logicielle engendrée par les échecs informatiques et qui introduit de nouveaux facteurs de validation de la qualité logicielle : le génie logiciel ;
3. Les nombreuses exigences liées au coût, aux délais et à la complexité des projets informatiques.

L'utilisation de méthodes de développement de logiciels permet ainsi l'élaboration de systèmes informatiques de manière fiable et viable tout en répondant à l'ensemble des exigences du client et du génie logiciel.

Il existe plusieurs méthodes de développement informatique. L'on distingue deux approches : l'approche traditionnelle et l'approche agile. Les deux approches se distinguent essentiellement dans la manière de décomposer le projet. Les méthodes cartésiennes ou fonctionnelles ou encore traditionnelles se sont imposées les premières.

L'approche traditionnelle

Cette approche s'inspire directement de l'architecture des ordinateurs. Les méthodes traditionnelles prônent une démarche strictement planifiée avec une séquence d'activités bien définie. La succession des activités et le planning doivent être clairement respectés et aucun changement n'est permis. Il est attendu du client une spécification des besoins globale, détaillée, claire, précise et validée en entrée. Ainsi, tout doit être prévisible, du début du projet à la livraison du produit, d'où l'appellation de méthodes prédictives.
Selon le planning adopté, les méthodes cartésiennes proposent plusieurs modèles d'exécution des activités du projet :

1. Le **modèle en cascade** : dans ce modèle, le processus de développement est découpé séquentiellement et de façon linéaire selon les activités intrinsèques du cycle de vie du développement logiciel : l'analyse, la conception, le codage et les tests. Le plan de déroulement des phases (planification prédictive) est élaboré en tout début de processus. Le passage à une phase donnée n'est fait que si le résultat de la phase précédente a été validé et jugé satisfaisant par le client et les utilisateurs.
2. Le **modèle en V** : Le cycle en V est à la base de tout développement logiciel, il en représente les activités intrinsèques. Il tient d'avantage compte de la réalité que le modèle en cascade, le processus de développement n'est pas réduit à un enchainement de tâches séquentielles. Le modèle en V permet d'anticiper sur les

phases ultérieures de développement du produit en particulier les plans de test de qualifications et de performance.

Parmi les méthodes traditionnelles, nous pouvons citer : SADT, CORIG, ...

L'approche agile

Cette approche est définie par les concepts suivants : la simplicité, la légèreté, la souplesse, un lien fort avec le client. C'est dans cette optique que certains apparentent le développement agile aux notions de flexibilité, de rétroaction et d'adaptation au changement rapide et continu.

Une méthode agile est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif avec juste ce qu'il faut de formalisme. Elle génère un produit de haute qualité tout en prenant en compte l'évolution des besoins des clients et en anticipant sur les risques. Il y'a continuellement des aller et retour avec le client. L'application logicielle est livrée par version incrémentale. Les versions successives sont aussi fiables que le livrable final en termes de tests et de validation. En quelque sorte le processus est déroulé comme un enchaînement de « mini-cascades ». A chaque nouvelle itération, l'ensemble de l'architecture et de la conception logicielle est reconsidéré, le code est retravaillé.

Les méthodes agiles aspirent donc à améliorer la réactivité et l'adaptabilité des sociétés de logiciels et constituent un moyen de survie dans un environnement instable en s'accompagnant des valeurs suivantes :

1. Les individus et les interactions plutôt que les processus et les outils ;
2. L'application fonctionnelle plutôt que la documentation compréhensive ;
3. La collaboration avec le client plutôt que la négociation des contrats ;
4. La réponse au changement plutôt que le suivi d'un plan.

L'agilité comprend plusieurs courants de pensée qui ont conduits à des méthodes différentes, reposant sur les mêmes concepts mais présentant des singularités. Les méthodes Scrum, Kanban, et XP (eXtreme Programming) sont des exemples de ces méthodes.

La méthode SCRUM

Scrum est une méthode agile de gestion de projet qui permet de produire la plus grande

valeur métier dans la durée la plus courte. Elle a pour objectif d'améliorer la cohésion de l'équipe et la rapidité du processus de développement. Le nom Scrum renvoie à une pratique généralement connue au rugby signifiant la « mêlée ».
Cette méthode qualifie un ensemble de rôles, d'instruments de gestion et de pratiques managériales favorisant un environnement basé sur la transparence, l'inspection, le suivi et l'adaptation. Le cycle de vie d'un projet Scrum peut être découpé en trois parties :

1. Phase d'**initiation ou démarrage** : il s'agit d'une phase linéaire où l'on définit le périmètre fonctionnel du système et la liste des fonctionnalités (**Backlog**) agencées par ordre de priorité, d'effort, de complexité et de risque. C'est aussi à ce niveau que l'architecture est définie.
2. Phase de **développement** est un processus empirique : le projet est découpé en cycles itératifs d'une durée de deux semaines ou **sprints**. Chaque sprint regroupe une ou plusieurs fonctionnalités du Backlog. Tout au long de cette phase, le travail réalisé est mesuré et contrôlé et une amélioration constante du prototype est faite.
3. Phase de **Clôtures** est une phase linéaire de gestion de la livraison du produit final.

La figure 2.1 montre l'articulation générale de Scrum.
Les responsabilités managériales sont réparties sur trois rôles fondamentaux :

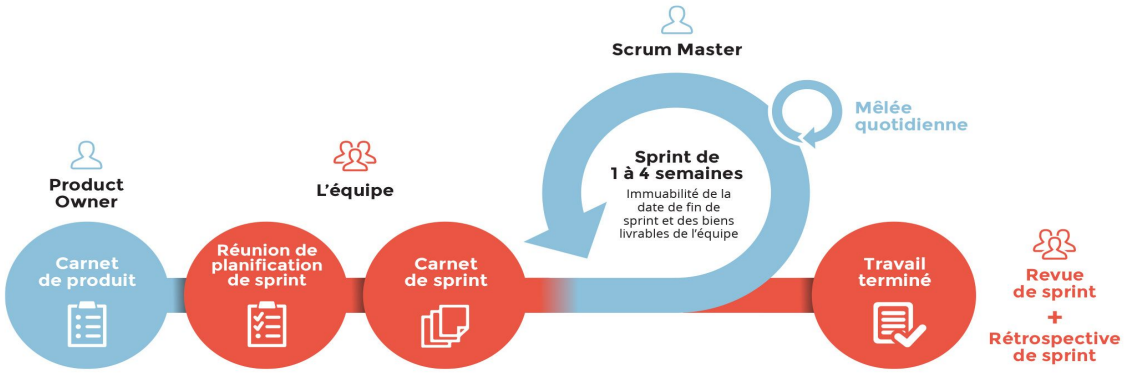

FIGURE 2.1 – Articulation générale de la méthode Scrum

1. **Scrum Master**
2. **Product Owner**
3. **Équipe Scrum**

Les artéfacts et pratiques de Scrum

1. **Product Backlog** : état courant des tâches à accomplir ;
2. **Sprint** : itération de deux semaines ;
3. **Effort-Estimation** : permanente sur les entrées du Backlog ;
4. **Sprint Backlog** : Product Backlog limité au sprint en cours ;
5. **Daily Scrum Meeting** : ce qui a été fait, ce qui reste à faire ;
6. **Sprint Review Meeting** : Présentation des résultats du Sprint. 2.1

SCRUM contre KANBAN

Scrum est plus prescriptif que Kanban, qui évite de définir les rôles et les équipes et qui n'a pas de structure formelle de réunions. Kanban ne prescrit pas non plus d'itérations – bien qu'elles puissent être incorporées si vous le souhaitez.

Les techniques de visualisation des processus de Kanban le rendent idéal pour les équipes colocalisées qui travaillent sur un backlog d'éléments sujets à des changements fréquents (par exemple, Kanban est souvent utilisé par les équipes de support).

Le tableau Kanban est cependant souvent adopté par les équipes Scrum sous la forme d'un tableau de tâches et est utilisé pour suivre la progression tout au long d'un sprint.

La limite de la règle Work In Progress dans Kanban la rend également adaptée aux équipes ayant des ressources limitées ou lorsque l'entrée de chaque membre est requise sur chaque élément. Cela pourrait s'appliquer, par exemple, à une équipe de communication au sein d'une grande organisation.

Alors que Scrum limite la quantité de travail dans chaque sprint, la charge de travail est déterminée par l'estimation relative de la taille de chaque histoire (en points) et est approuvée par l'équipe Scrum à chaque session de planification.

Alors qu'une équipe Kanban suit le « temps de cycle » et optimise les délais d'exécution aussi courts et prévisibles que possible, une équipe Scrum vise à améliorer son rendement sur les sprints successifs et à améliorer la « vélocité » de l'équipe (le nombre de points d'estimation relatifs complétés dans un sprint). Cela rend sans doute Scrum plus adapté à la mise à l'échelle – il semble certainement plus familier et prévisible, ce qui peut être rassurant pour les grandes organisations.

SCRUM contre XP

Dans Scrum, les équipes et les réunions sont assez gravées dans le marbre [1] alors que la question de savoir comment le travail est réellement fait est laissée aux équipes pour décider elles-mêmes. XP, d'autre part, est livré avec un ensemble de pratiques de base qui pourraient sembler accablantes pour le débutant Agile.

On pourrait dire que Scrum est une méthodologie, qui est plus concernée par la productivité tandis que XP est plus préoccupé par l'ingénierie.

La valeur que les pratiques XP peuvent ajouter est incontestable et de nombreuses organisations qui utilisent Scrum adoptent la programmation par paires, le développement piloté par les tests et le refactoring comme des pratiques qui améliorent la qualité, accélèrent le processus de publication et / ou réduisent le besoin de revoir le travail en raison de la dette technique.

Outre les itérations plus courtes, d'autres éléments importants qui différencient XP de Scrum sont les suivants :

1. Les équipes XP travaillent sur des éléments dans un ordre de priorité strict alors qu'une équipe Scrum ne s'attaque pas nécessairement à chaque élément dans l'ordre de priorité une fois dans le sprint ;
2. Les équipes XP peuvent intégrer de nouveaux éléments de travail dans une itération et changer d'éléments de taille équivalente (tant qu'ils n'ont pas été démarrés) si le client décide d'une nouvelle priorité.

En termes de similitudes, le rôle du client dans XP est très similaire à celui du Product Owner dans Scrum – en ce sens qu'ils aident à écrire des user stories, à les hiérarchiser et sont toujours disponibles pour les développeurs – bien que moins bien définis.

Scrum et XP imposent tous deux une réunion debout quotidienne. Bien que les deux soulignent l'importance de la co-localisation, seul XP le rend décisif. Voir le site https://manifesto.co.uk/kanb
vs-scrum-vs-xp-an-agile-comparison/.

1. Dans l'antiquité, les engagements pour la constructions de bâtiments importants étaient gravés sur des plaques de marbre (Athènes : arsenal du Pirée, Delphes). Les travaux s'étendant sur de nombreuses années, on ne pouvait faire confiance aux tablettes de cire ou aux papyrus. Sur ces plaques, on définissait par exemple la grandeur du bâtiment ou le montant des amendes pour les retards. Ce qui n'était pas « gravé dans le marbre » n'était donc pas contractuel. Voir le lien https://fr.wiktionary.org/wiki/graver-dans-le-marbre

2.1.2 Logiciel de gestion du projet : Trello

Trello est un outil de gestion de projet en ligne, lancé en septembre 2011 et inspiré par la méthode Kanban. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement. Pour en savoir plus, veillez visiter le lien https://fr.wikipedia.org/wiki/Trello.

2.2 Conception

Dans cette section, je présente le langage et le logiciel de modélisation que j'ai utilisé pour concevoir notre solution.

2.2.1 Langage de modélisation : UML

UML est un langage de modélisation orientée objet permettant aux développeurs de modéliser un système d'information en considérant plusieurs vues chacune reflétant un aspect comportemental [2] ou structurel [3] du système.
En effet, nous avons opté pour UML au détriment de la MERISE car nous avons besoin d'une approche de conception prenant en considération l'aspect orienté objet pour :

1. Pouvoir mettre le focus sur le rôle temporel des instances d'objets lors de déclenchement desactions (à travers le diagramme de séquence) ;
2. Faciliter par la suite la génération des classes DAO à partir du diagramme de classes.

Certes, UML est très riche en matière de modélisation et propose au total 13 diagrammes chacun fournissant une vision particulière du système à concevoir. Dans notre contexte, je me suis limités à 4 diagrammes explicités sur le tableau 2.1.

2.2.2 Logiciel de modélisation

Modelio est un logiciel open source et multiplateforme permettant, entre autres, la modélisation UML et Business Process Model and Notation (BPMN). Pour en savoir plus,

2. Diagramme de cas d'utilisation, diagramme d'activité, diagramme de séquence, diagramme d'interaction, etc.

3. Diagramme de classe, diagramme de composants, diagramme de déploiement, diagramme de structure composite, etc.

veuillez visiter le lien https://www.modelio.org/about-modelio/features.html.
Sans doute, les logiciels de modélisation UML sont nombreux, à savoir, Visual Paradigm, Eclipse Papyrus, StarUML, PowerDesigner, Umbrello, etc. Vu que les diagrammes UML que nous voulons réaliser sont disponibles dans tous ces logiciels, il n'y avait pas en effet un choix à argumenter car tous les choix étaient satisfaisants. Mais, de façon subjective, nous pouvons préciser que l'avantage de Modelio dans notre contexte est le fait que je m'y suis déjà habitués. Les diagrammes que nous avons réalisés avec Modelio sont ceux mentionnés ci-après.

Diagramme	**Rôle**
Diagramme de cas d'utilisation	Présenter les acteurs du système, ses fonctionnalités, les relations entre les acteurs et entre les fonctionnalités.
Diagramme d'activité	Déterminer l'enchaînement des différentes étapes qui composent une fonctionnalité du système.
Diagramme de séquence	Fournir une vue détaillée du diagramme d'activité en mettant le focus sur l'ordre chronologique et sur les objets crées et les méthodes appelées.
Diagramme de classe	Représenter la structure interne du système sous forme de classes et d'interfaces et préciser les différentes relations entre elles.

TABLE 2.1 – Rôles des diagrammes UML utilisés.

2.3 Implémentation

Dans cette section, je présente les langages, les logiciels, les frameworks et les motifs d'architecture que j'ai utilisé.

2.3.1 Front-end

Editeur de texte : VS Code

VS Code (Visual Studio Code) est un éditeur de code source réalisé par Microsoft pour Windows, Linux et macOS. [9] Les fonctionnalités incluent la prise en charge du débogage, lacoloration syntaxique, la saisie semi-automatique intelligente du code, les extraits de code, la refactorisation du code et Gitintégré. Les utilisateurs peuvent modifier le thème,les raccourcis clavier,les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires. Pour en savoir plus, veillez visiter le lien https://en.wikipedia.org/wiki/VisualStudioCode.

Languages

Langage	**Contexte d'utilisation**
Dart	Création des applications moible
PHP	Langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP

TABLE 2.2 – Contexte d'utilisation des différents langages utilisés.

Framework : Flutter

Flutter est un cadre de développement d'applications mobiles open source permettant de développer des applications mobiles natives Andriod et iOS en un seul code.

Flutter a été introduit par Google. La version stable de Flutter est Flutter 1.0 qui a été publiée le 4 décembre 2018. Le ciel est la première application Flutter qui a fonctionné dans l'OS Andriod. .

Framework Web : Laravel

Laravel est un framework d'application Web avec une syntaxe expressive et élégante. Nous croyons que le développement doit être une expérience agréable et créative pour être vraiment épanouissante. Laravel tente de simplifier le développement en facilitant les

tâches courantes utilisées dans la majorité des projets Web, telles que l'authentification, le routage, les sessions et la mise en cache.

Laravel vise à rendre le processus de développement agréable pour le développeur sans sacrifier les fonctionnalités de l'application. Les développeurs heureux font le meilleur code. À cette fin, nous avons tenté de combiner le meilleur de ce que nous avons vu dans d'autres frameworks Web, y compris des frameworks implémentés dans d'autres langages, tels que Ruby on Rails, ASP.NET MVC et Sinatra.

Laravel est accessible, mais puissant, fournissant des outils puissants nécessaires pour les applications volumineuses et robustes. Une superbe inversion du conteneur de contrôle, un système de migration expressif et une prise en charge des tests unitaires étroitement intégrée vous offrent les outils dont vous avez besoin pour créer n'importe quelle application qui vous est confiée. .

IDE : Android Studio

Android Studio est l'IDE officiel pour le système d'exploitation Android de Google,construit sur le logiciel IntelliJ IDEA de JetBrainset conçu spécifiquement pour le développement Android. Pour en savoir plus veillez visiter le lien
https://en.wikipedia.org/wiki/AndroidStudio.
Essentiellement, je l'ai utilisé pour lancer l'application sur android.

2.3.2 Back-End

IDE : Visual Studio

Microsoft Visual Studio est un IDE de Microsoft. Il est utilisé pour développer des programmes informatiques, ainsi que des sites Web, des applications Web, des services Web et des applications mobiles. Pour en savoir plus, veillez visiter le lien https://en.wikipedia.org/wiki/Microso

MVC

L'objectif du motif MVC (Model-View-Controller ou Modèle-Vue-Contrôleur) est un modèle dans la conception de logiciels. Il met l'accent sur la séparation entre la logique métier et l'affichage du logiciel. Cette «séparation des préoccupations» permet une meilleure répartition du travail et une maintenance améliorée. Le tableau ci-dessous en présente une

explication.

Couche logicielle	**Rôle**
Modèle	Gère les données et la logique métier
Vue	Gère la disposition et l'affichage
contrôleur	Chemine les commandes des parties "model" et "view"

TABLE 2.3 – Rôle des trois couches logicielles du motif MVC.

La figure ci-dessous explique les moments d'intervention de chaque couche du motif MVC.

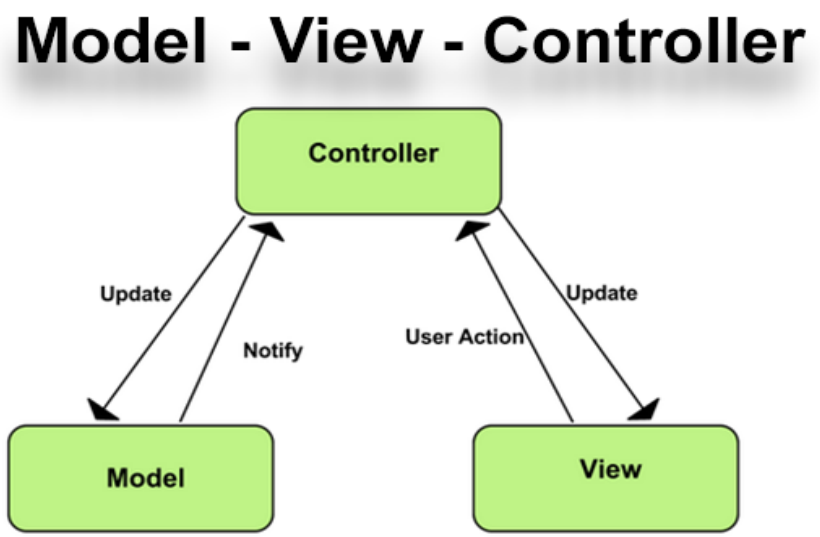

FIGURE 2.2 – Rôle des trois couches logicielles du motif MVC

2.4 Sécurité

Dans cette section, je présente les algorithmes et techniques de chiffrement que nous avons utilisés pour sécuriser davantage l'application.

2.4.1 Hachage des mots de passe