

**ENCADREMENT ACADEMIQUE PAR**

**ENCADREMENT ACADEMIQUE PAR**

**THEME**

**FILIERE**

**OPTION**

**REALISE PAR**

**2019-2020**

**ANNÉE UNIVERSITAIRE**

**Méthodes informatiques appliquées à la gestion des entreprises**

**Ingénierie informatique et réseaux**

**LOGO ENTREPRISE**

**Omar Ouakili**

**Conception et Réalisation d’une Solution Informatique de Test de Personnalité**

**M. Bousmah**

**A. Cheikhaoui**

# Dédicace

A *mes très chers parents et mes sœurs ;*

A*ucun mot, aucune expression ne pourra témoigner ma reconnaissance et ma gratitude à votre égard, vos prières, vos sacrifices abondants et votre soutien innombrable resteront à jamais gravés dans mon cœur et mon esprit ;*

A *mes collègues de classe ;*

A *mes amis et amies ;*

M***erci pour les instants inoubliables que nous avons partagés ensemble ;***

A *tous ceux qui m’aiment…*



A *tous ceux que j’aime…*

Je *dédie ce travail…*

***Omar Ouakili***

# Remerciements

Je tiens à remercier dans un premier temps La **Royale Marocaine d’Assurance** et l’ensemble de son personnels pour leur accueil sympathique et leur disponibilité.

Je tiens tout particulièrement à remercier mon encadrant **Mr cheikaoui**

Pour ses conseils et le temps qu’il m’a consacré tout au long de cette période, sachant répondre à toutes mes interrogations.

Mes sincères remerciements sont tout particulièrement adressés à **Mr Bousmah Mohamed** Qui m’a fait l’honneur de m’encadrer, et grâce à qui j’ai pu mener à bien ce travail. Je tiens à lui témoigner notre reconnaissance pour la qualité de son encadrement, son soutien, ses précieux conseils et pour l’aide inconditionnelle qu’il m’a apportée tout au long de la période de mon stage de fin d’étude.

Et pour conclure, mes remerciements s’adressent aussi à toute l’équipe pédagogique de **L’Ecole Marocaine des Sciences de l'Ingénieur** **(EMSI) et** surtout à mes professeurs pour la bonne formation qu’ils m’ont donnés tout au long de mon cycle ingénieur



# Résumé

Le présent rapport décrit en détails les phases de réalisation du projet de mon stage de fin d’études, que j’ai effectué au sein de Royale Marocaine d’Assurance « RMA ». L’objet de mon stage était l’analyse, la conception et l’implémentation d’une Solution Informatique de Test de Personnalité Pour mener à bien la mission, j’ai essayé de respecter les bonnes pratiques du processus de développement logiciel. Au début, une étude préliminaire du système existant était nécessaire pour mieux interpréter les besoins et les exigences exprimés par le client. Ensuite, en fonction des besoins du client, la phase de conception du projet a été formalisée par la création d'un ensemble de diagrammes UML modélisant la solution. Pour développer le projet, j’ai adopté l’approche basée sur les micro-services avec des cycles de développement courts dans un contexte d’agilité.

L’architecture technique de la solution proposée est basée sur le principe de l’inversion de contrôle en se basant sur des Frameworks. Coté Backend en utilisant le Framework Spring. Pour l’aspect sécurité, Coté Frontend, c’est le Framework Angular qui a été choisi. Le choix de cette architecture vise de permettre aux développeurs de se concentrer sur les aspects métier de l’application et déléguer aux Frameworks les aspects techniques tels que la sécurité et la gestion des performances.



Après une phase de réalisation et les tests, l’application actuellement répond parfaitement bien aux besoins.

# Liste des figures

[Figure 1 : Le groupe FinanceCom 13](#_Toc51784009)

[Figure 2 :HIERARCHIE DE RMA ASSURANCE 18](#_Toc51784010)

[Figure 3: méthode de calcul 21](#_Toc51784011)

[Figure 4 : La représentation en V 24](#_Toc51784012)

[Figure 5 : diagramme de Gantt 25](#_Toc51784013)

[Figure 6 : Diagramme de cas d’utilisation 29](#_Toc51784014)

[Figure 7 : Diagramme des classes 35](#_Toc51784015)

[Figure 8 : Diagramme de séquence d'un utilisateur 36](#_Toc51784016)

[Figure 9 : Diagramme de séquence d'Admin 37](#_Toc51784017)

[Figure 10 : Spring 40](#_Toc51784018)

[Figure 11 : Hibernate 40](#_Toc51784019)

[Figure 12 : Restfull api 41](#_Toc51784020)

[Figure 13 : Java 41](#_Toc51784021)

[Figure 14 : Eclipse 41](#_Toc51784022)

[Figure 15 : Postman 42](#_Toc51784023)

[Figure 16 : JEE 42](#_Toc51784024)

[Figure 17 : JEE 42](#_Toc51784025)

[Figure 18 : Maven 43](#_Toc51784026)

[Figure 19 : Angular 43](#_Toc51784027)

[Figure 20 : Bootstrap 44](#_Toc51784028)

[Figure 21 : Angular Material 44](#_Toc51784029)

[Figure 22 : TypeScript 45](#_Toc51784030)

[Figure 23 : Visual Studio Code 45](#_Toc51784031)

[Figure 24 : Npm 45](#_Toc51784032)

[Figure 25 : Node.js 45](#_Toc51784033)

[Figure 26 : Architecture 3 tiers 46](#_Toc51784034)

[Figure 27 : SCHEMA DES DIFFERENTS COUCHE DE L’APPLICATION 49](#_Toc51784035)

[Figure 28 : model MVC 51](#_Toc51784036)

[Figure 29 : Architecture applicative 51](#_Toc51784037)

[Figure 30 : Inscription 54](#_Toc51784038)

[Figure 31 : Explication sur le test 55](#_Toc51784039)

[Figure 32 : Début du test 56](#_Toc51784040)

[Figure 33 : Remerciement 56](#_Toc51784041)

[Figure 34 : Authentification admin 57](#_Toc51784042)

[Figure 35 : List des candidats 57](#_Toc51784043)

[Figure 36 : résultat 58](#_Toc51784044)

[Figure 37 : La description des échelles 59](#_Toc51784045)

[Figure 38 : Ajouter des commentaire 60](#_Toc51784046)

|  |  |
| --- | --- |
| Abréviation | Désignation |
| **RMA** | Royale Marocaine d’assurance |
| **UML** | Unified Modeling Language |
| **REST** | Representational state transfer |
| **JSON** | JavaScript Object Notation |
| **HTML** | Hypertext Markup Language |
| **DSI** | Directeur des systèmes d'information |
| **CSS** | Cascading Style Sheets |
| **API** | Application Programming Interface |
| **HTTP** | Hypertext Transfer Protocol |
| **MVC** | Model Vue Conroller |
| **SCBD** | Système de gestion de base de données |
| **JEE** | Java Entreprise Edition |

# Liste des abréviations

SOMMAIRE

[Dédicace 2](#_Toc51786478)

[Remerciements 3](#_Toc51786479)

[Résumé 4](#_Toc51786480)

[Liste des figures 5](#_Toc51786481)

[Liste des abréviations 7](#_Toc51786482)

[Introduction général 10](#_Toc51786483)

[Chapitre1 : CONTEXTE GENERAL DU PROJET 11](#_Toc51786484)

[l. La Royale Marocaine d’Assurance 12](#_Toc51786485)

[1. Le groupe FinanceCom 13](#_Toc51786486)

[2 . Présentation de l’organisme d’accueil Royale Marocaine d’Assurance 14](#_Toc51786487)

[3. Département Développement des Systèmes d’Information 16](#_Toc51786488)

[4. Organigrammes 16](#_Toc51786489)

[II. Présentation générale du projet 19](#_Toc51786490)

[1. Qu’est-ce que le test PAPI? 19](#_Toc51786491)

[2. Structure du test papi 20](#_Toc51786492)

[3. Structure du questionnaire 21](#_Toc51786493)

[4. Exploitation du questionnaire 21](#_Toc51786494)

[5. Structure de l’application/introduction 22](#_Toc51786495)

[III. Conduite du projet 23](#_Toc51786496)

[1. Processus de développement 23](#_Toc51786497)

[2. Planification et organisation du projet 25](#_Toc51786498)

[Chapitre 2 : Conception 26](#_Toc51786499)

[1. Introduction 27](#_Toc51786500)

[2. Choix de la méthodologie de conception 27](#_Toc51786501)

[3. Présentation d’UML 27](#_Toc51786502)

[4. Diagramme de cas d’utilisation 29](#_Toc51786503)

[5. Diagramme des classes 35](#_Toc51786504)

[6. Diagrammes de séquence 36](#_Toc51786505)

[7. Conclusion 38](#_Toc51786506)

[Chapitre 3 : Etude technique 39](#_Toc51786507)

[I. Environnement de développement 40](#_Toc51786508)

[1. Backend 40](#_Toc51786509)

[2. Frontend 43](#_Toc51786510)

[ll. Architecture de la solution 46](#_Toc51786511)

[1. Capture des besoins techniques 46](#_Toc51786512)

[2. Principes généraux d’architecture 50](#_Toc51786513)

[3. Architecture applicative 51](#_Toc51786514)

[Chapitre 4 : Réalisation de la solution 53](#_Toc51786515)

[1. Inscription 54](#_Toc51786516)

[2. Explication sur le test 55](#_Toc51786517)

[3. Le test 56](#_Toc51786518)

[4. Remerciement 56](#_Toc51786519)

[5. Authentification admin 57](#_Toc51786520)

[6. List des candidats 57](#_Toc51786521)

[7. Résultat 58](#_Toc51786522)

[8. La description des échelles 59](#_Toc51786523)

[9. Ajouter des commentaires 60](#_Toc51786524)

[Conclusion générale 61](#_Toc51786525)

[Bibliographie et webographie 62](#_Toc51786526)

# Introduction générale

La formation des jeunes et leur insertion dans la vie professionnelle est un sujet d'actualité dont tous les responsables soulignent l'urgence et le nécessaire développement.

Pour les dirigeants des écoles, cet élan national confine un travail de longue haleine entreprise par eux depuis de nombreuses années pour affiner d’adéquation emploi et permettre aux étudiants de réussir leur entrée sur le monde du travail.

Il est banal de souligner le nécessaire complémentaire entre l'enseignement théorique et la connaissance Pratique. Cependant, pour atteindre cet objectif, il faut développer une collaboration entre les écoles de formation De l'entreprise. Celle-ci reconnaît de plus en plus l'utilité de cet apport en ressources humaines afin de gagner la bataille économique et technologique.

S'il y a une cause aujourd'hui qui doit mobiliser l'attention et l'énergie de toutes les forces vives de la notion, elle ne peut être que celle de la lutte pour l’emploi. Les stages en entreprises constituent pour les étudiants, le meilleur moyen d'adaptation aux exigences du marché d'emploi et contribue à atteindre les objectifs prioritaires suivant :



* La découverte du monde du travail.
* La confrontation des méthodes et techniques enseignées avec les pratiques en vigueur au sein de l’entreprise.
* L'établissement de relations plus étroites entre les établissements de formation et leur environnement économique, en vue de rechercher la meilleure adaptation entre la formation et les besoins du marché de l’emploi.

La mission de l’EMSI est d’assurer à ses lauréats un savoir, un savoir-faire et un savoir être qui leur permettent d’entrer rapidement dans la vie active. Grâce à une compétence reconnue et constatée, par le secteur professionnel, lors de leur travail de fin d’études, ils intègrent aisément le monde de l'emploi et s'y distinguent par leur faculté à être très vite productifs.

Après cette introduction générale, le présent rapport sera structure en 4 chapitres :

* Contexte général du projet
* Conception
* Etude technique
* Réalisation de la solution

# Chapitre1 : CONTEXTE GENERAL DU PROJET

## l. La Royale Marocaine d’Assurance

RMA Assurance est une Compagnie d'assurance marocaine filiale du groupe FinanceCom.

RMA Assurance a hérité de deux histoires collectives, celle de la Royale Marocaine d'Assurance et celle d'Al Wataniya.

En 1949, une élite de marocains nationalistes, visionnaires, fonda la première compagnie d'assurance marocaine, la Royale Marocaine d'Assurance, qui, à force de conviction et de ténacité a su se forger un destin exceptionnel pour devenir l'une des compagnies les plus représentatives du marché marocain de l'assurance.

Le véritable tournant a été pris en 1988 avec l'arrivée de Monsieur Othman BENJELLOUN, qui a accepté de présider aux destinées de cette institution et qui a insufflé une nouvelle dynamique à la Royale Marocaine d'Assurance, qui a vu sa croissance fortifiée, sa rentabilité renforcée et sa notoriété améliorée.



Le point d'orgue de notre activité a été, sans aucun doute, la participation à la privatisation de BMCE Banc dans le cadre d'un consortium de partenaires de renom.

Via les filiales BMCE Banc et Royale Marocaine d'Assurance et accompagnés des partenaires Commerzbank et UBP, le Groupe a procédé au rachat des parts du GAN dans AL Wataniya et l'Alliance Africaine d'assurances en Décembre 1998.

Le scénario retenu fut l'absorption par Al Wataniya de l'Alliance Africaine. Cette opération a pris effet opérationnellement le 1er janvier 2001.

Le 31 mars 2003, le Président Othman BENJALLOUN annonçait le coup d'envoi du projet de fusion entre la Royale Marocaine d'Assurance et Al Wataniya.

Ce rapprochement donna naissance le 1er janvier 2005 à RMA WATANYA, la 1ère compagnie du secteur marocain des assurances.

Le 1er janvier 2016, le nom de RMA WATANYA s’est changé pour s’appeler RMA-Royale Marocaine d’Assurance.

### 1. Le groupe FinanceCom

Comme on a déjà mentionné, RMA est une filiale du groupe FinanceCom qui est un groupe financier de premier plan qui intervient dans divers secteurs d’activités à fort potentiel de croissance, regroupant Banque, Assurance, Gestion d’Actifs, Technologies, Media, et Télécommunications.

L’ensemble des activités du groupe FinanceCom est synthétisé dans la figure ci-dessous.

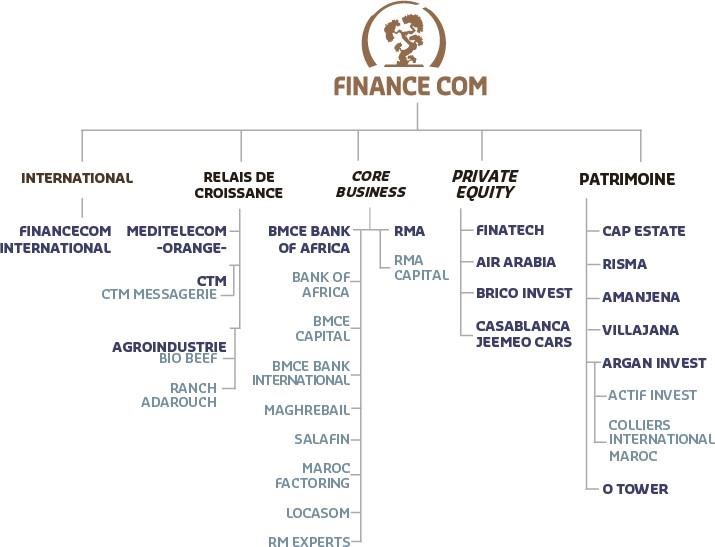


Figure : Le groupe FinanceCom

### 2 . Présentation de l’organisme d’accueil Royale Marocaine d’Assurance

La RMA Assurance est dotée d'une organisation orientée réseaux qui s’articule autour de trois Pôles :

* + - Pôle Production et Distribution
    - Pôle Support et Finances
    - Pôle Prestation Technique et Développement

* + Le Pôle Production et Distribution :

Regroupant les directions :

* + - Direction Courtage :

Cette Direction compte 5 Départements :

* + **Département Commercial :** Responsable de la prospection, le suivi et la recherche de la clientèle
  + **Département Ingénierie :** Tarification, Visites de risques, Prévention.



* + **Département Assurances Incendie Multirisques :** Production, Gestion sinistres.
  + **Département Maritime :** Production, Tarification, Gestion sinistres et recours, Gestion certificats et pool.
  + **Département Risques techniques- Aviation :** Production, Tarification, Gestion sinistres.

* + - Direction Réseaux Exclusif :

Elle gère un réseau de Bureaux Directs (Salariés de la compagnie) et d'Agents exclusifs. Cette Direction bénéficie d'une large autonomie et pouvoirs de décisions dans le cadre d'un plan d'action annuel (objectif de CA et de rentabilité).

* + Le Pôle Support & Finances : Regroupant :
    - Direction Informatique :

Elle dispose d'une équipe d'informaticiens de haut niveau, d'un puissant réseau informatique, elle développe et met à la disposition des différentes entités, les applications informatiques répondant à leurs besoins qui permettent une simplification des tâches et un gain important en productivité.

* + - Direction des Ressources Humaines :

Elle met à la disposition des entités les ressources humaines et matérielles nécessaires, à l'accomplissement de leurs missions. Elle a adopté des moyens modernes de gestion des ressources humaines dans le cadre d'un plan Schéma Directeur. Elle s'est dotée en outre d'un puissant logiciel de gestion du matériel et de stocks.

* + - **Direction Comptable et Financière :**

La Direction Comptable est chargée des enregistrements comptables, de suivi de la trésorerie et de l'établissement des comptes périodiques de fin d'exercice et des états comptables et extra comptables exigés par les différentes instances légales. La gestion financière permet la gestion optimale de la trésorerie et des placements mobiliers et immobiliers et le contrôle de gestion.

* + - **Direction des immeubles :**

Elle gère et développe (acquisition, constructions) le patrimoine immobilier de la Compagnie.



* + - **Pôle Prestation Technique et Développement :**

Ce pôle gère les Sinistres Contentieux importants et regroupant :

* + - * **Direction Santé**
      * **Direction Automobile Corporel**
      * **Direction Prestation AT**
      * **Direction Marketing**
      * **Direction Technique**
      * **Département Assurances de Personnes**

### 3. Département Développement des Systèmes d’Information

Durant mon stage, j’ai pu travailler en collaboration avec le chef de service et les développeurs.

La mission principale du département est de :

* + Coder, paramétrer et implémenter la solution
  + Analyser des besoins fonctionnels et techniques
  + Dérouler les tests et valider son développement
  + Gérer la maintenance corrective et évolutive des sites et des applications
  + Réaliser la documentation associée à la solution ▪ Assister / former les utilisateurs

### 4. Organigrammes

La RMA Assurance est dotée d'une organisation orientée réseaux qui s’articule autour de trois Pôles :

* Pôle Production et Distribution
* Pôle Support et Finances
* Pôle Prestation Technique et Développement

§ **Le Pôle Production et Distribution :**

Regroupant les directions :

**Direction Courtage :**

Cette Direction compte 5 Départements :

* **Département Commercial :** Responsable de la prospection, le suivi et la recherche de la

clientèle

* **Département Ingénierie :** Tarification, Visites de risques, Prévention.
* **Département Assurances Incendie Multirisques :** Production, Gestion sinistres.
* **Département Maritime :** Production, Tarification, Gestion sinistres et recours, Gestion certificats et pool.
* **Département Risques techniques- Aviation :** Production, Tarification, Gestion sinistres.

**Direction Réseaux Exclusif :**

Elle gère un réseau de Bureaux Directs (Salariés de la compagnie) et d'Agents exclusifs. Cette Direction bénéficie d'une large autonomie et pouvoirs de décisions dans le cadre d'un plan d'action annuel (objectif de CA et de rentabilité).

§ **Le Pôle Support & Finances :**

Regroupant :

* **Direction Informatique :**

Elle dispose d'une équipe d'informaticiens de haut niveau, d'un puissant réseau informatique, elle développe et met à la disposition des différentes entités, les applications informatiques répondant à leurs besoins qui permettent une simplification des tâches et un gain important en productivité.

* **Direction des Ressources Humaines :**

Elle met à la disposition des entités les ressources humaines et matérielles nécessaires, à l'accomplissement de leurs missions. Elle a adopté des moyens modernes de gestion des ressources humaines dans le cadre d'un plan Schéma Directeur. Elle s'est dotée en outre d'un puissant logiciel de gestion du matériel et de stocks.

* **Direction Comptable et Financière :**

La Direction Comptable est chargée des enregistrements comptables, de suivi de la trésorerie et de l'établissement des comptes périodiques de fin d'exercice et des états comptables et extra comptable exigés par les différentes instances légales. La gestion financière permet la gestion optimale de la trésorerie et des placements mobiliers et immobiliers et le contrôle de gestion.

* **Direction des immeubles :**

Elle gère et développe (acquisition, constructions) le patrimoine immobilier de la Compagnie.

§ **Pôle Prestation Technique et Développement :**

Ce pôle gère les Sinistres Contentieux importants et regroupant :

* **Direction Santé**
* **Direction Automobile Corporel**
* **Direction Prestation AT**
* **Direction Marketing**
* **Direction Technique**
* **Département Assurances de Personnes**

La hiérarchie de la direction assurance est illustré par le l’organigramme



Ci- dessous

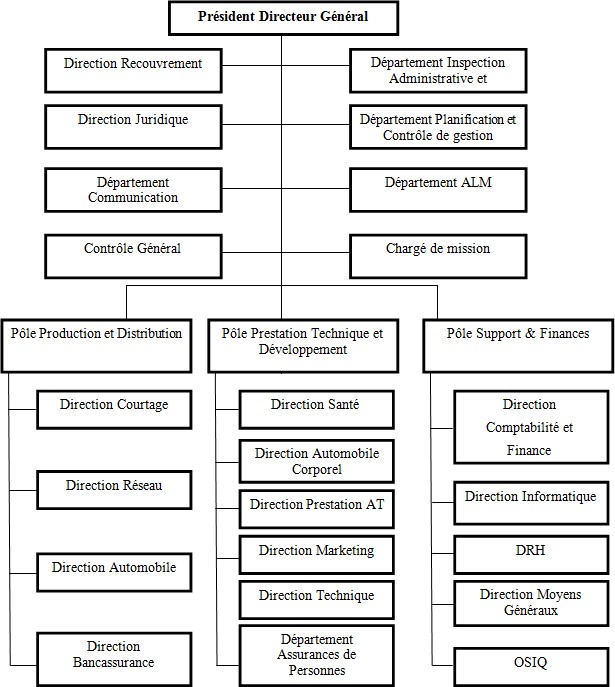


Figure :HIERARCHIE DE RMA ASSURANCE

## Présentation générale du projet

Mise en place d’une application web de test de personnalité basé sur une architecture moderne, qui permet aux nouveaux employeurs qui vent embaucher chez RMA pour avoir la structure personnalité de chaque personne

### Qu’est-ce que le test PAPI?

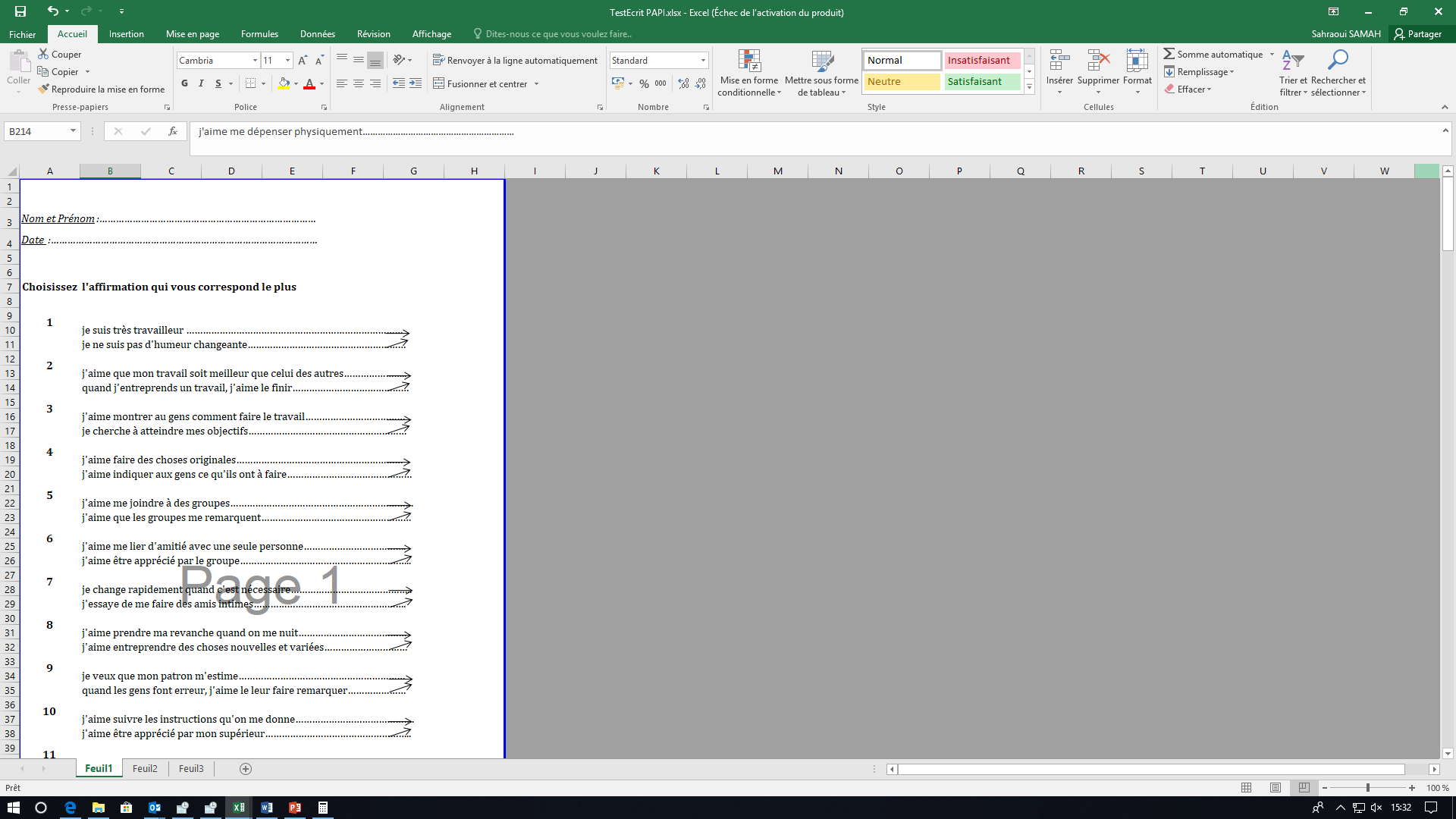
* 1. Il s’agit d’un test de personnalité appelé « Inventaire des Perceptions et Préférences » qui prends en compte les préférences et les perceptions pertinentes dans la sphère professionnelle.
  2. Les résultats PAPI sont représentés visuellement sous la forme d’une rosace. Cette dernière possède un certain nombre de caractéristiques :
     1. Les « Facteurs », regroupant un grand groupe d’échelles, sont situés à la bordure de la rosace PAPI (par exemple, « Style de travail»).
     2. Les « échelles » se situent à l’intérieur de la rosace PAPI, chacune ayant un facteur qui lui est propre (par exemple, « Rythme de travail » et « Expression de la vitalité» pour le facteur « Dynamisme»).
     3. Les « Rôles » apparaissent en noir (par exemple, « mode de réflexion») tandis que les Besoins sont colorés (par exemple, « volonté de se mettre en avant»).
     4. Chaque échelle possède une lettre qui lui est propre (par exemple, « R » pour «mode de réflexion» et « X » pour «volonté de se mettre en avant»).
  3. Le PAPI possède 20 échelles, regroupées en 7 facteurs, qui comprennent chacun entre 2 et 4 échelles. Parmi ces 20 échelles, 10 sont des échelles de Besoins et 10 sont des échelles de Rôles.

### Structure du test papi

* 10 Besoins :
  + - Type de responsabilités souhaitées (P)
    - Volonté de se mettre en avant (X)
    - Volonté d’appartenir à un groupe (B)
    - Implication affective (O)
    - Réactions face au changement (Z)
    - Assertivité (K)
    - Relations à l’autorité (F)
    - Attachement aux règles et structures (W)
    - Gestion de la tâche (N)
    - Gestion des objectifs (A)
* 10 Rôles :
  + Image du leader (L)
  + Rythme de décision (I)
  + Rythme de travail (T)
  + Expression de la vitalité (V)
  + Nature des contacts sociaux (S)
  + Mode de réflexion (R)
  + Intérêt pour les détails (D)
  + Type d’organisation (C)
  + Contrôle émotionnel (E)
  + Gestion de l’énergie (G)

### Structure du questionnaire

* Le questionnaire est composé de 90 items/questions avec deux affirmations; le participant est invité à en choisir celle qui lui correspond le plus (le questionnaire n’est valide qu’avec le renseignement de la totalité des questions).
* Chaque échelle possède 9 items/questions qui lui sont propres, ce qui donne au total 180 items de personnalité.
* Exemple d’item/question:



Affirmation 2

Affirmation 1

### Exploitation du questionnaire

* Le calcul des scores se fait par la somme des items cochés par le participant à travers la grille qui suit :
* L’échelle G (Gestion de l’énergie) regroupe les affirmations 1 des items/questions 1,11,21,31,41,51,61,71 et 81

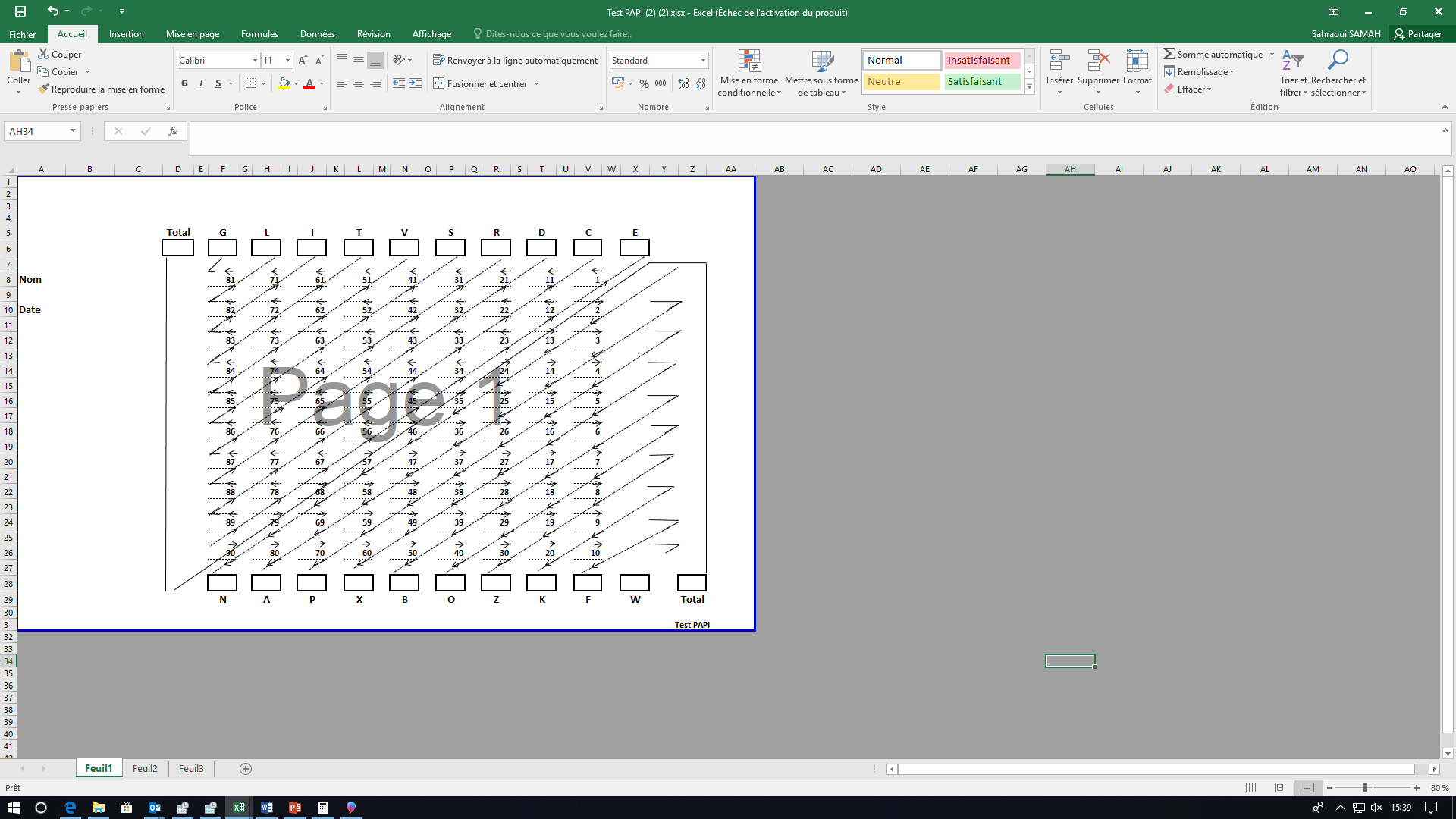


Figure : méthode de calcul

### Structure de l’application/introduction

Le test de personnalité est conçu pour découvrir vos traits de personnalité dans l’environnement professionnel. Il vous interroge sur la façon dont vous réagissez dans différentes situations et comment vous interagissez avec d'autres personnes dans le milieu du travail. Dans ce test, il n'y a pas de bonnes ou de mauvaises réponses. Il suffit de répondre d'une manière spontanée.

Dans chacune des 90 items/questions, vous serez confronté à 2 affirmations. On vous demande de choisir celle qui vous ressemble le plus.

Par exemple :

* + J’aime m’intégrer à des groupes.
  + J’aime aider les gens à se décider.

Lorsque vous répondez au questionnaire, n'oubliez pas ce qui suit :

* + Assurez-vous de répondre à toutes les questions.
  + Même si aucune des affirmations ne vous correspond, donnez la réponse que vous considérez la moins lointaine de votre personnalité.
  + Ce test n'est pas chronométré; cependant, il devrait prendre environ 20 minutes.

## Conduite du projet

### Processus de développement

#### 1.1 Définition de cycle de vie

Le cycle de vie d'une application comprend toutes les étapes depuis sa conception et sa réalisation jusqu’à sa mise en œuvre. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement du logiciel et la vérification de son processus de développement. L'origine de ce découpage provient du constat que les erreurs ont un coût si élevé qu'elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs le plutôt possible.

#### 1.2 Les activités d’un cycle de vie

Le cycle de vie suivi pour réaliser un site e-commercial, comprend généralement au minima les activités suivantes :

**Spécification des besoins** : elle consiste à définir la finalité du projet et son intégration dans une stratégie globale.

**Conception générale** : dans cette activité, il s'agit de la préparation de l'architecture générale du logiciel.

**Conception détaillée** : elle consiste à définir précisément chaque sous-ensemble du logiciel.

**Développement** : il s’agit d’une traduction des fonctionnalités définies dans la phase de conception en langage de programmation.

**Tests unitaires :** ils permettent de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux normes définies dans la conception.

**Intégration** : dite aussi tests systèmes, elle consiste à vérifier que le logiciel correspond exactement au cahier des charges du projet en obtenant enfin un manuelle d’utilisation bien détaillé aux utilisateurs.

**Validation :** c'est-à-dire la validation de conformité du site avec les buts spécifiés à la première étape du cycle de vie.

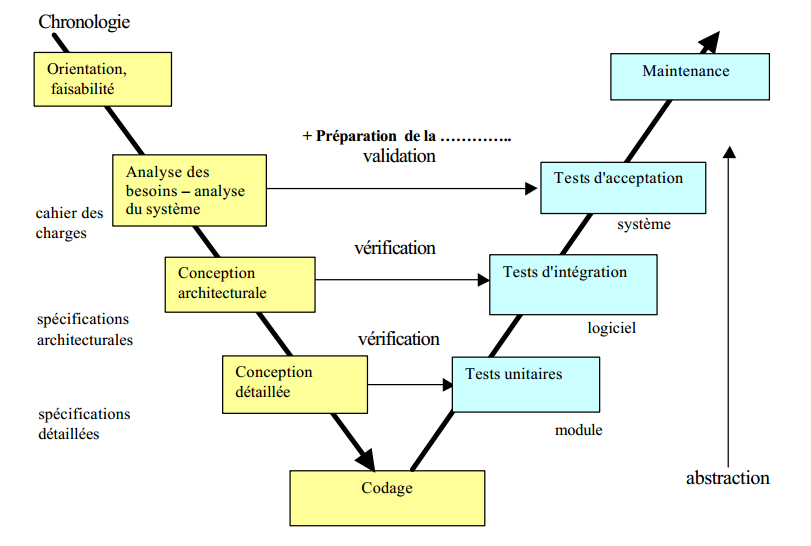


Figure : La représentation en V

La représentation en V tient d'avantage compte de la réalité, le processus de développement n'est pas réduit à un enchaînement de tâches séquentielles. Elle montre que :

* + C’est en phase de spécification que l'on se préoccupe des procédures de qualification
  + C’est en phase de conception globale que l'on se préoccupe des procédures d'intégration
  + C’est en phase de conception détaillée que l'on prépare les tests unitaires

Le modèle de cycle de vie en V permet d'anticiper sur les phases ultérieures de développement du produit. En particulier le modèle en V permet de commencer plus tôt:

* + Plan de tests de qualification,
  + Plan d'évaluation des performances,

Le modèle en V comme celui en cascade conduisent à commencer plus tôt la documentation utilisateur.

### 2. Planification et organisation du projet

La planification d’un projet est considérée comme une étape importante d’avant-projet, elle est la clé de sa réussite et consiste à prévoir le déroulement du projet tout au long de ses phases.

Mon stage du projet de fin d’étude au sein de la Royale Marocaine d’Assurance représente une période de travail de 6 mois et demi.

Après la réunion de démarrage, on a pu établir un planning de différentes tâches à réaliser.

Ce planning avait pour objectif de nous aider à respecter le temps qui nous a été accordé dans le cadre de ce stage.

La durée des phases de travail a été estimée objectivement par le volume de chaque partie et, bien entendu, pouvait varier en fonction des imprévus ou des difficultés rencontrés au cours de la réalisation.

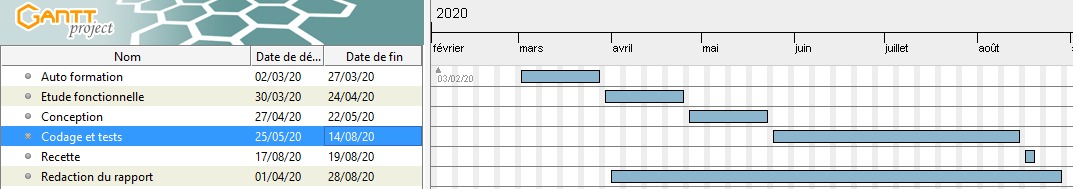


Figure : diagramme de Gantt

**Conclusion**

Dans ce chapitre nous avons défini le contexte global du projet, en présentant l’organisme d’accueil, sa structure, ces départements et son organisation, ensuite nous avons décrit le projet à réaliser en définissant le domaine duquel il relève, ainsi la précision de l’objectif de celui-ci. Finalement, et dans

Un dernier temps, nous nous sommes penchées sur le choix d’un processus de développement du projet dont le déroulement de ces phases est explicité par le diagramme de Gantt.

# Chapitre 2 : Conception

## Introduction

Le Modèle conceptuel de données est une représentation statique du système d’information. Il a comme objectif de constituer une représentation claire et cohérente des données manipulées dans le système d’information. Cette section, sera présentée comme suit : nous commençons par le choix de la méthodologie de conception et justification puis nous présentons quelques diagrammes d’activités et de séquences à titre indicatif. Ensuite nous présentons le diagramme de classe.

## Choix de la méthodologie de conception

Dans la cadre de notre projet, nous avons opté pour le langage UML comme une approche de conception. Ci-dessous, nous présentons ce langage puis nous justifions notre choix.

## Présentation d’UML

UML (Unified Modeling Language) est un langage formel et normalisé en termes de modélisation objet. Son indépendance par rapport aux langages de programmation, aux domaines de l’application et aux processus, son caractère polyvalent et sa souplesse ont fait lui un langage universel. En plus UML est essentiellement un support de communication, qui facilite la représentation et la compréhension de solution objet. Sa notation graphique permet d’exprimer visuellement une solution objet, ce qui facilite la comparaison et l’évaluation des solutions. L’aspect de sa notation, limite l’ambigüité et les incompréhensions. UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues. Une vue est constituée d'un ou plusieurs diagrammes. On distingue deux types de vues :



**La vue statique**, permettant de représenter le système physiquement :

* + Diagrammes de classes : représentent des collections d'éléments de modélisation statiques (classes, paquetages...), qui montrent la structure d'un modèle.
  + Diagrammes d’objets : ces diagrammes montrent des objets (instances classes dans un état particulier) et des liens (relations sémantiques) entre objets.
  + Diagrammes de cas d’utilisation : identifient les utilisateurs du système (acteurs) et leurs interactions avec le système.
  + Diagrammes de composants : permettent de décrire l'architecture physique statique d'une application en termes de modules : fichiers sources, librairie exécutable, etc.
  + Diagrammes de déploiement : montrent la disposition physique du matériel qui compose le système et la répartition des composants sur ce matériel.

**La vue dynamique**, montrant le fonctionnement du système :

* + Diagrammes de collaboration : montrent des interactions entre objet (instances de classes et acteurs). Diagrammes de séquence : permettent de représenter des collaborations eu objets selon un point de vue temporel on y met l'accent sur la chronologie (envois de messages).
  + Diagrammes d'états-transitions : permettent de décrire les changements d'états d'un objet ou d'un composant, en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.
  + Diagrammes d’activités : (une variante des diagrammes d'états-transitions) servent à représenter graphiquement le comportement d'une méthode ou déroulement d'un cas d'utilisation. La conception de notre interface a été élaborée en suivant la démarche suivante :
  + L'élaboration des diagrammes de cas d'utilisation. Cette étape a été réalisée suite à la spécification fonctionnelle de l’application.



* + Recensement des classes candidates et élaboration du diagramme des classes.
  + Dresser les diagrammes de collaboration et de séquences pour mettre en évidence interactions entre les différents objets du système.
  + Elaborer le diagramme d'états-transitions pour montrer les différents états l'interface.

## Diagramme de cas d’utilisation

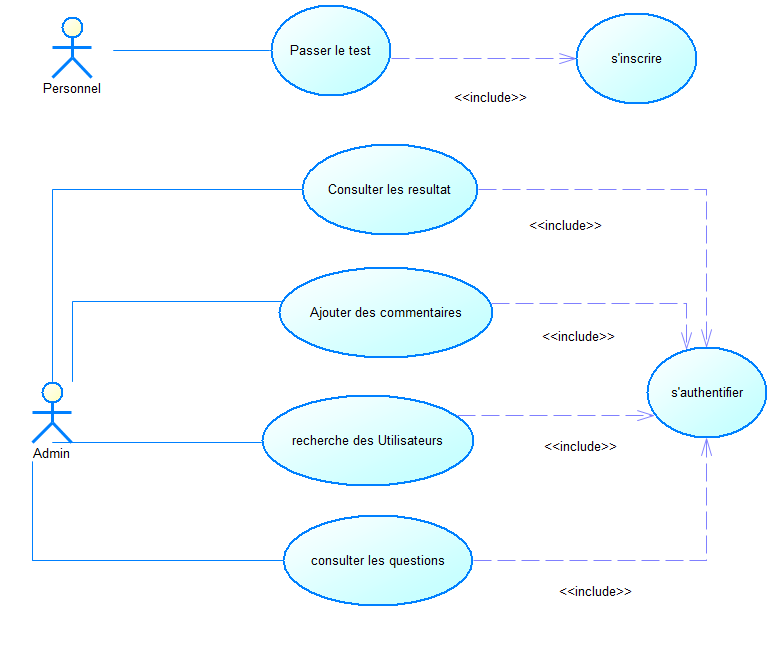


Figure : Diagramme de cas d’utilisation

• **Description textuelle du cas d’utilisation « S'inscrire » :**

|  |  |  |
| --- | --- | --- |
| **Sommaire d’identification** | | |
| **Titre** | **S'inscrire** | |
| **But** | **S'inscrire et autorisation d’accès** | |
| **Résumé** | **L’utilisateur faire une inscription pour accéder au système** | |
| **Acteur** | **Utilisateur** | |
| **Description des enchainements** | | |
| **Précondition** | | **Post condition** |
| **L’admin doit avoir un compte** | | **Accès à son espace privé** |
| **Scenario nominale Système** | | |
| **1.L’utilisateur accède à l’application** | | **2- Le système affiche le formulaire d’inscription** |
| **5.L’utilisateur saisit les champs obligatoire** | | **3- La système vérifie si l’utilisateur faire une inscription correcte** |
|  | | **5- le formulaire d’inscription est Présenté.** |
|  | | **6-L’utilisateur est redirigé vers l’application** |
| **Scénario d’erreur** | | |
| **E1 : les champs obligatoires vides**   * **Le système afficher un message d’erreur.** * **Le système demande de rentrer les champs.**   **E2 : login et mot de passe non valide**   * **Le système affiche un message d’erreur « accès refusé »** * **Le système demande de rentrer les champs.** | | |

• **Description textuelle du cas d’utilisation « S’authentifier » :**

|  |  |  |
| --- | --- | --- |
| **Sommaire d’identification** | | |
| **Titre** | **S’authentifier** | |
| **But** | **Authentification et autorisation d’accès** | |
| **Résumé** | **L’admin introduit son login et mot de passe pour accéder au système** | |
| **Acteur** | **Admin** | |
| **Description des enchainements** | | |
| **Précondition** | | **Post condition** |
| **L’admin doit avoir un compte** | | **Accès à son espace privé** |
| **Scenario nominale Système** | | |
| **1.L’utilisateur accède à l’application** | | **2- Le système affiche le formulaire d’authentification** |
| **5.L’utilisateur saisit son login et mot de passe** | | **3- La système vérifie si l’utilisateur dispose déjà autorisation pour l’application** |
|  | | **5- le formulaire d’authentification est Présenté.** |
|  | | **6-Le système vérifie le login et le mot de passe**  **L’utilisateur est redirigé vers l’application** |
| **Scénario d’erreur** | | |
| **E1 : les champs obligatoires vides**   * **Le système afficher un message d’erreur.** * **Le système demande de rentrer les champs.**   **E2 : login et mot de passe non valide**   * **Le système affiche un message d’erreur « accès refusé »** * **Le système demande de rentrer les champs.** | | |

* **Consulter Les Résultats**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sommaire article** | | | | |
| **Titre** | | **Consulter** | | |
| **But** | | **Admin consulte un article.** | | |
| **Résumé** | | **L’Admin consulte l’interface des Résultats.** | | |
| **Acteur** | | **Admin** | | |
| **Description des enchainements** | | | | |
| **Pré-condition** | | | **Post condition** | |
| **L’Admin est authentifié** | | | **Consulter article** | |
| **Scenario nominale** | | | | |
| **L’Utilisateur** | | | **Système** | |
| **1. l’Admin s’authentifié**  **3. l’Admin demande au système la page des résultats**  **5-l’Admin choisit un utilisateur pour consulter les résultats** | | | **2- Le système vérifie login et mot de passe**  **4- le système affiche la liste des Utilisateurs.**  **6-Le système affiche le résultat** | |
| **Scénario alternatif** | | | | |
| **L’Admin** | | | **Système** | |
| **1. L’Admin s’authentifié**  **3. L’Admin demande au système la page des utilisateurs pour voir les résultats** | | | **2- Le système vérifie login et mot de passe** | |
| **Scénario d’erreur** | | | | |
| • **Le système affiche un message d’erreur « échec de la connexion ».** | | | | |

• **Description textuelle du cas d’utilisation « Ajouter commentaire»**

|  |  |  |
| --- | --- | --- |
| **Sommaire article** | | |
| **Titer** | **Ajouté un commentaire** | |
| **But** | **L’administrateur créer un commentaire.** | |
| **Résumé** | **L’administrateur consulte l’interface d’ajout un commentaire, rempli les champs nécessaires et valide l’ajout** | |
| **Acteur** | **Administrateur** | |
| **Description des enchainements** | | |
| **Pré-condition** | | **Post condition** |
| **L’administrateur est authentifié** | | **Ajouter article** |
| **Scenario nominale** | | |
| **Administrateur** | | **Système** |
| **1. L’administrateur s’authentifié**  **3. L’administrateur demande au système la page système la page des résultats.**  **5- L’administrateur choisit l’opération d’ajout commentaire.**  **7- L’administrateur rempli les champs nécessaires.** | | **2- Le système vérifie login et mot de passe**  **4- le système affiche la liste des utilisateurs.**  **6-Le système affiche le formulaire d’ajout un commentaire.**  **8- si l’ajout est valide le système affiche**  **Un message « Commentaire ajoutée avec succès ».** |
| **Scénario alternatif** | | |
| **L’administrateur** | | **Système** |
| **1. L’administrateur s’authentifié**  **3. L’administrateur demande au système la page système la page des résultats.**  **5- L’administrateur choisit l’opération d’ajout commentaire.**  **7- L’administrateur rempli les champs nécessaires.** | | **2- Le système vérifie login et mot de passe**  **4- le système affiche la liste des utilisateurs.**  **6-Le système affiche le formulaire d’ajout un commentaire.**  **8- si l’ajout est non valide le système affiche un message « échec d’ajout ».** |
| **Scénario d’erreur** | | |
| **E1 : les champs obligatoires**  • **Le système affiche un message d’erreur « commentaire invalide ».** | | |

➢ **Recherche des utilisateurs**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sommaire article** | | | | |
| **Titre** | | **Recherche des utilisateurs** | | |
| **But** | | **Admin Recherche des utilisateurs.** | | |
| **Résumé** | | **L’Admin consulte l’interface des Utilisateurs.** | | |
| **Acteur** | | **Admin** | | |
| **Description des enchainements** | | | | |
| **Pré-condition** | | | **Post condition** | |
| **L’Utilisateur est authentifié** | | | **Consulter article** | |
| **Scenario nominale** | | | | |
| **L’Utilisateur** | | | **Système** | |
| **1. l’Admin s’authentifié**  **3. l’Admin demande au système la page des Utilisateurs.**  **5-l’Admin choisit l’opération de visualisé les Utilisateurs.** | | | **2- Le système vérifie login et mot de passe**  **4- le système affiche la liste des Utilisateurs possibles.**  **6-Le système affiche le continue de Utilisateurs.** | |
| **Scénario alternatif** | | | | |
| **L’Utilisateur** | | | **Système** | |
| **1. L’Admin s’authentifié**  **3. l’Admin demande au système la page des Utilisateurs.** | | | **2- Le système vérifie login et mot de passe**  **4-Le système affiche message**  **« aucun Utilisateur trouver !!»** | |
| **Scénario d’erreur** | | | | |
| • **Le système affiche un message d’erreur « échec de la connexion ».** | | | | |

## Diagramme des classes

Le diagramme de classes fournit une vue globale de la structure statique du système en présentant ses classes et les relations entre elles. Il présente également la structure interne de la classe, en exposantses attributs et opérations. Le diagramme de classes permet de structurer le travail de développementde manière très efficace et correcte, c'est pour cette raison qu'il est considéré comme le plus important par rapport aux autres diagrammes UML.

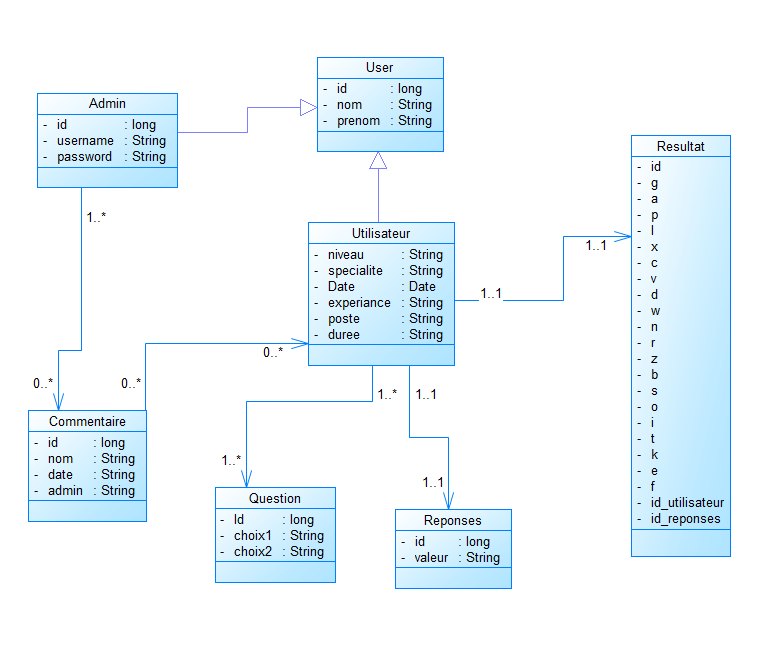


Figure : Diagramme des classes

## Diagrammes de séquence

Le diagramme de séquence modélise l’aspect dynamique du système. Il s’agit d’une séquence d’interactions d’un point de vue temporel entre le système et les acteurs.

On va tracer le scénario de quelque cas d’utilisation de notre application :

* Le premier diagramme modélise le cas d’utilisation d’un utilisateur

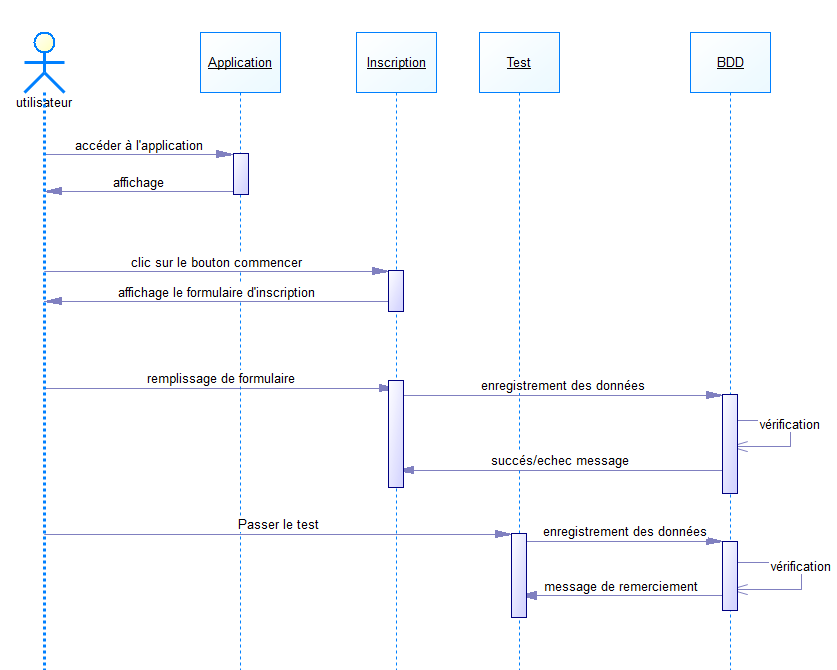


Figure : Diagramme de séquence d'un utilisateur

* Le deuxième diagramme sert à modéliser le cas d’utilisation d’un Admin

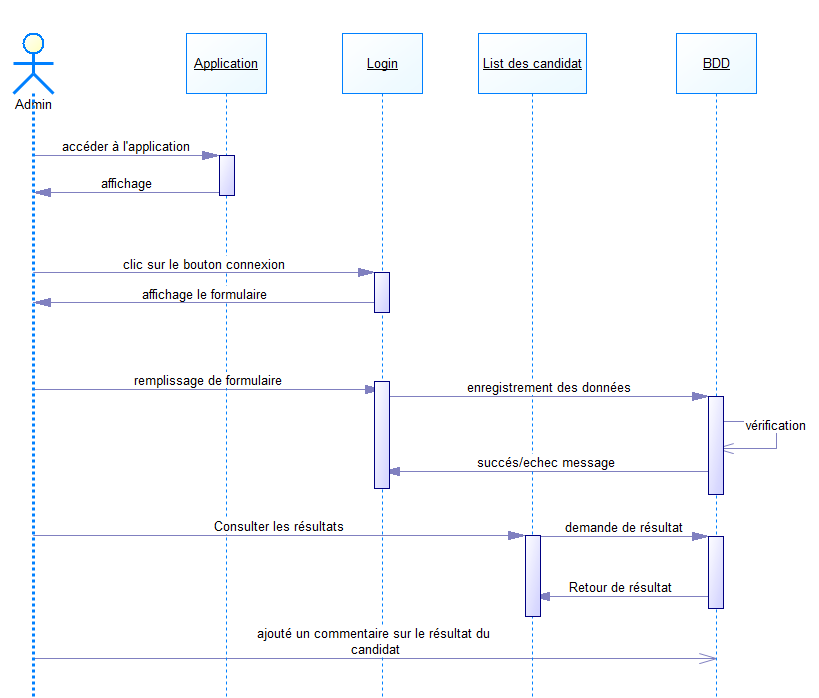


Figure : Diagramme de séquence d'Admin

## Conclusion

Dans ce chapitre, on a représenté l’étude préliminaire et l’analyse fonctionnelle du projet, or on a identifié d’abord les acteurs et les interactions entre les acteurs et le système en modélisant un diagramme d’activité, ensuite les objectifs métiers en détaillant chaque objectif avec unemodélisation d’un diagramme de cas d’utilisation et une description textuelle puis on a réalisé des diagrammes de séquence pour l’ordre du déroulement de chaque objectif métier, enfin, après la phase d’analyse des cas d’utilisation, on a pu extraire les différentes entités et les relations entrecelles-ci via le diagramme de classes.

# Chapitre 3 : Etude technique

## I. Environnement de développement

### 1. Backend

Le développement du backend était relativement simple car Spring est un bon Framework, facile à utiliser et que j’avais bien pris en main lors de mon travail de début de stage, à ce moment, la base de données de l’application était encore dans l'étape d'initialisation, les évolutions fréquentes de la base de données entraînant également une modification du backend, donc ce que j’ai fait sur le backend est principalement le travail de création d'une structure de code permettant l’héritage et la réutilisabilité, pendant le développement du backend, j'ai ajouté la bibliothèque Restfull API, qui est une bibliothèque pour simplifier le processus de conception de l'API, j'ai également ajouté Postman comme un outil pour simplifier les tests de l’API, la documentation de l'API.

▪ Framework



Figure : Spring

* **Spring** : Spring est un Framework libre pour construire et définir l'infrastructure d'une application java, dont il facilite le développement et les tests.
* 

Figure : Hibernate

* **Hibernate** : Hibernate est un Framework open source gérant la persistance des objets en base de données relationnelles. C’est un outil de mapping objet/relationnel pour le monde Java. Le terme mapping objet/relationnel (ORM) décrit la technique consistant à faire le lien entre la représentation objet des données et sa représentation relationnelle basée sur un schéma SQL. ▪ Librairie





Figure : Restfull api

* **Restfull api** : REST est un style d'architecture logicielle définissant un ensemble de contraintes à utiliser pour créer des services web. Les services web conformes au style d'architecture REST.



Figure : Java

* **Java** : est un langage de programmation orienté objet La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d’exploitation tels que Unix, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et Framework associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.
* Autre



Figure : Eclipse

* + **Eclipse** : Est un outil devenu indispensable pour la programmation Java ; le gain de temps est énorme. C’est un IDE extensible, libre, lancé par IBM La spécificité d’Eclipse vient du fait de son architecture qui est totalement développée autour de la notion de plugin.



Figure : Postman

* + **Postman** : permet de construire et d’exécuter des requêtes HTTP, de les stocker dans un historique afin de pouvoir les rejouer, mais surtout de les organiser en Collections.



Figure : JEE

* + **JEE** : La plateforme JEE désigne l’ensemble des technologies java pour le développement d’applications d’entreprise distribuées (multicouches, n-tiers). C’est une plateforme qui présente une forte orientation serveur et qui se compose de deux volets principaux



Figure : JEE

* + **Oracle** : est un SGBD (système de gestion de bases de données) édité par la société du même nom (Oracle Corporation), oracle est un SGBD permettant d'assurer :
    - * La définition et la manipulation des données
      * La cohérence des données
      * La confidentialité des données



* + - * L'intégrité des données
      * La sauvegarde et la restauration des données
      * La gestion des accès concurrents



Figure : Maven

* + **Maven** : Couramment appelé Maven, Apache Maven est un outil de gestion et d'automatisation de production des projets logiciels Java en général et Java EE en particulier.

### 2. Frontend

La conception et la mise en place du frontend est la partie que j’ai mené entièrement. C'est un travail où j’ai donc investi le plus de temps pour apprendre et approfondir.

Angular est un Framework jeune, donc il existe également une limitation lorsqu'il s'agit de l’utiliser avec d'autres bibliothèques.

* Framework



Figure : Angular

* + - **Angular :** est un cadriciel côté client open source basé sur Type Script et un

Framework JavaScript Open Source, il utilise l’architecture MVM (Modèle Vue Modèle), proche du modèle MVC. Cela va permettre de structurer le code et bien séparer la vue (l’interface) des modèles (fonctionnement).



Figure : Bootstrap

* + - **Bootstrap** : est une [collection d'outils](https://fr.wikipedia.org/wiki/Framework) utile à la création de [sites](https://fr.wikipedia.org/wiki/Site_web) et [d'applications web.](https://fr.wikipedia.org/wiki/Application_web) C'est un ensemble qui contient des codes HTML et [CSS,](https://fr.wikipedia.org/wiki/Cascading_Style_Sheet) des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript e option.



Figure : Angular Material

* + - **Angular Material** : est une bibliothèque de composants d'interface utilisateur destinée aux développeurs de Angular JS. Les composants Angular Material aident à la construction de pages Web et d'applications Web attrayantes, cohérentes et fonctionnelles, tout en respectant les principes de conception Web modernes tels que la portabilité du navigateur, l'indépendance du périphérique et la dégradation progressive. Cela aide à créer des sites Web rapides, beaux et réactifs. Il s'inspire du design de matériel de Google.



* + - **Ng-Bootstrap** : contient un ensemble de directives angulaires natives basées sur le balisage et le CSS de Bootstrap. Par conséquent, aucune dépendance à jQuery ou à JavaScript de Bootstrap n'est requise. Ng-Bootstrap est basé sur Bootstrap 4.
* Langage

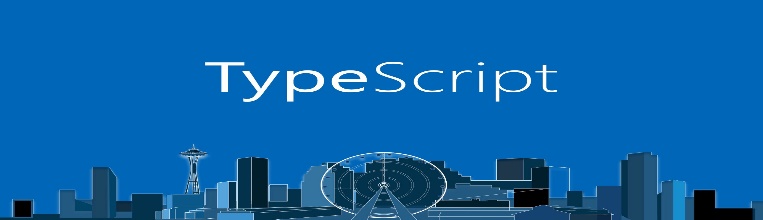


Figure : TypeScript

* + - **TypeScript** : est un langage de programmation libre et open source développée par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript.
* Autre



Figure : Visual Studio Code

* + **Visual Studio Code :** est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS2.



Figure : Npm

* + **Npm** : est le gestionnaire de paquets officiel de Node.js. Depuis la version 0.6.3 de Node.js, npm fait partie de l'environnement et est donc automatiquement installé par défaut**.**



Figure 25 : Node.js

* + **Node.js** : est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge.



## ll. Architecture de la solution

### Capture des besoins techniques

La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent pas de la description du métier des utilisateurs,elle est primordiale pour la conception d’architecture.

#### 1.1 Architecture logique

L’architecture logique, décrit le mode de communication ainsi la répartition des tâches au sein du réseau et l’organisation de la circulation des données.

On cite par exemple les différentes architectures logiques existantes : paire-à-paire et client-serveur.

Dans le cadre de mon projet l’architecture adoptée est l’architecture 3-tiers qui est une extension d’une architecture client-serveur. Elle est composée de 3 éléments :

Niveau 1 : C’est la machine cliente munie d’une interface homme-machine chargée de la présentation en demandant une ressource à travers des requêtes. Elle ne fait aucun traitement.

Niveau 2 : La couche fonctionnelle, comprend le serveur qui est dans notre cas un serveur Web muni d'extensions applicatives.

Niveau 3 : le serveur de base de données.

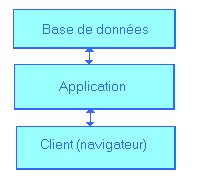


Figure : Architecture 3 tiers

#### 1.2 Modèle de spécification logicielle

On s’intéresse aux fonctionnalités du système technique en élaborant la spécification logicielle. Les cas d’utilisations représentent la pierre angulaire de cette étape. Ils vont être utilisés différemment que dans les besoins fonctionnels.

Dans ce cas l’acteur sera nommé d’exploitant, désigne celui qui bénéficie que des fonctionnalités techniques. En effet, cet acteur existe toujours, chaque utilisateur s’authentifie, bénéficie d’une certaine sécurité, rapidité de réponse ou même une consultation d’aide en ligne, ou autre. Parfois tout cela peut être utilisé sans même l’utilisateur se rendra compte.

- **Identification des cas d’utilisation technique**

Un cas d’utilisation technique est un cas destiné à l’exploitant. C’est une séquence d’actions produisant une valeur ajoutée opérationnelle ou purement technique. C’est-à-dire qu’il ne produit pas de valeur ajouté fonctionnelle.

L’acteur exploitant : est un acteur au sens d’UML, si ce n’est qu’il ne bénéficie que des fonctionnalités techniques du système. Il s’agit ici de l’utilisateur dans son sens le plus général, indépendamment des fonctions ou du métier qu’il réalise au travers de l’application.

Les cas d’utilisations techniques peuvent êtres recensées comme suit :

* Gestion de sécurité : sécurisation de l’application surtout au niveau de requêtes paramétrées.
* Le temps de réponse doit être optimal, qui reviens à optimiser le temps d’exécution des requêtes.
* Manipule des concepts métiers via des objets, d’où la nécessité d’établir un mécanisme de persistance, dans ce cas les fonctionnalités de ce mécanisme vont se restreindre à la lecture.
* Accessible via le web
* Disponibilité : l’application doit être disponible pour répondre au besoin métiers de l’utilisateur
* Evolutivité : l’application doit être conçue de sorte qu’elle soit ouverte à l’extension.  Réutilisabilité : la possibilité de réutiliser certains modules.
* Maintenance : Assurer une facilité de maintenance.
* Gestion des erreurs et des exceptions.

On constate que ces besoins, qui sont indépendants de la branche gauche (fonctionnelle), peuvent être liés à des problèmes récurrents.

* + - * **Organisation du modèle de spécification logicielle**

Le modèle de spécification logicielle décrit précédemment est superficiel, car en effet certains cas d’utilisations sont transverses à plusieurs responsabilités, de l’interface utilisateur à la base de données.

D’où, il est nécessaire d’organiser la spécification logicielle en couches selon la responsabilité pour prévoir le comportement technique de chacune.

On introduit alors une décomposition ou plutôt une organisation en couches logicielles, qui, chacune définit une responsabilité particulière.

* + - * **Organisation en couche logicielle**

Une couche logicielle représente un ensemble de spécifications ou de réalisations qui, expriment ou mettent en œuvre respectivement, un ensemble de responsabilités techniques et homogènes pour un système logiciel.

Les différentes couches sont empilées, de sorte que celles au niveau supérieur sollicitent un ou plusieurs services de celles qui sont en dessous.

Cette organisation permet de diviser le problème en sous-parties spécialisées.

Le schéma suivant liste les différentes couches logicielles d’une application, depuis l’interface graphique jusqu’aux entrepôts de stockage de données :

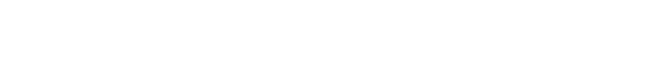
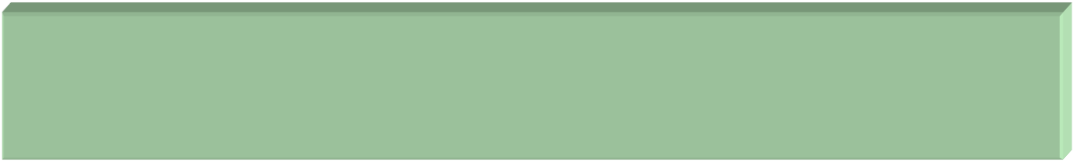


**Couche Métier**



-

Composants orientés service



-

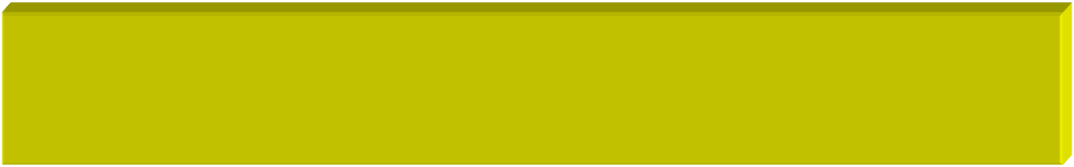
Client léger

-

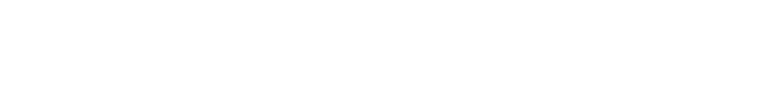
Client lourd



**Couche Client**



**Couche Présentation**

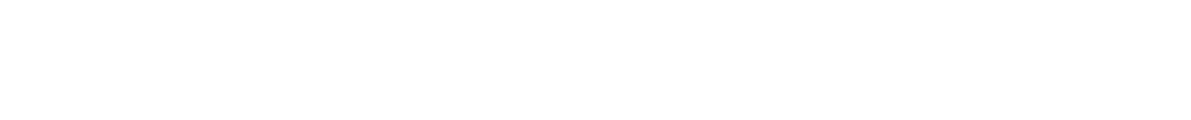


-

Contrôle de la navigation

-

Génération des pages



**Co**

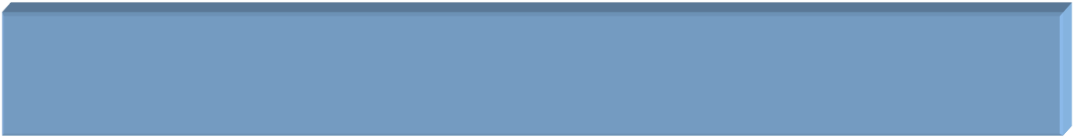
**uche**

**Intégration / persistance**



-

Objets techniques, Framework DAO, etc...



**Couche Ressources**



-

SGBD, ERP, CRM, Annuaire, MOM

Figure : SCHEMA DES DIFFERENTS COUCHE DE L’APPLICATION

Voici de manière résumée les responsabilités de chaque couche :

* + - **La couche Client :** représente le navigateur web.
    - **La couche Présentation :** contient les éléments qui contrôlent l’affichage, la restitution des écrans de l’application et le dialogue avec l’utilisateur
    - **La couche Métier :** est constituée des composants implémentant les cas d’utilisation métier. Cette couche a la responsabilité. Cette couche a la responsabilité :
      * De la gestion des validations et processus fonctionnels (cas d’utilisation).
      * De l’implémentation des règles de gestion métier (de contrôle, de calcul, de préparation des données à mettre à jour).
      * De l’intégrité transactionnelle (début, validation, annulation) des données applicatives (mises à jour des données préparées).
    - **La couche Intégration et Persistance :** représente les objets des sources de données et éventuellement de leurs relations.
    - **La couche Ressources :** drivers SGBD, Annuaire, MOM, etc…



### Principes généraux d’architecture

L’intérêt même d’un modèle d’architecture en couches réside dans l’indépendance des couches entre elles. Cela permet de disposer d’un système évolutif (changement de base de données ou d’outil de mapping objet / relationnel sans impacter les autres couches par exemple), et de pouvoir développer et tester séparément les couches.



Cela repose toutefois sur l’utilisation de Design Pattern éprouvés dans la structuration des couches et la communication inter-couches :

* + - MVC pour la Couche Présentation.

Modèle / Vue / Contrôleur (Couche Présentation)

* + - **La Vue** est (dans le cadre d’application web) le plus souvent implémentée par des pages JSP, afin de construire dynamiquement du HTML. Ces JSP peuvent inclure des taglibs (bibliothèques de tags) issues de Framework (Struts, DisplayTag) ou spécifiques.
    - **Le Modèle** est l’objet à qui est transmis la requête utilisateur. Le Modèle va traiter la requête, puis stocker les résultats de ces traitements, avant de les transmettre à la Vue. De plus, elle contient le résultat des saisies de l’utilisateur.
    - **Le Contrôleur** fait le lien entre la Vue et le Modèle, en gérant l’enchaînement de la

Navigation et le traitement des actions/événements d’un écran.

Cela est résumé dans le schéma suivant :

:

’

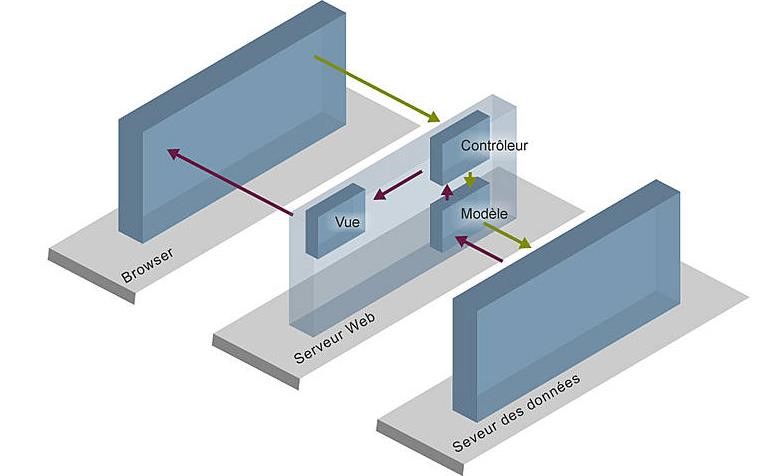


Figure : model MVC

La communication entre le contrôleur et le modèle peut se faire par l’intermédiaire d’un délégué (pattern Business Delegate).



**Cela permet de :**

Réduire le couplage entre la couche Présentation et la couche Métier.

Masquer à la couche Présentation la logique de localisation et d’instanciation des services de la couche Métier.

### Architecture applicative

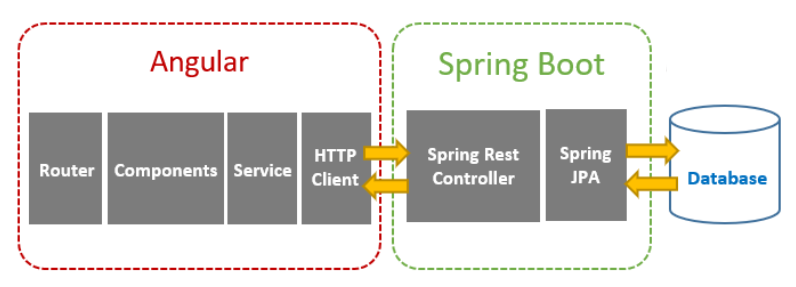


Figure : Architecture applicative

**IV. Conclusion**

Ce chapitre nous a permis d’apporter une vision globale de l'architecture de la solution ainsi qu’une description sur les technologies implémentées, le prochain chapitre va concrétiser tout ce qui précède et présenter la partie de la réalisation du projet.

# Chapitre 4 : Réalisation de la solution

Dans les chapitres précédents j’ai pu dégager les différentes fonctionnalités auxquelles doit répondre le système, ensuite j’ai formalisé ces fonctionnalités par des diagrammes UML et j’ai spécifié les différents choix techniques. Dans la suite je présente le travail réalisé à travers quelques exemples d’illustration.

### Inscription

* Accès participant qui l’invite à la création de son profil avec le renseignement des rubriques suivantes :
* Nom
* Prénom
* Niveau d’études : Bac, Bac+2, Bac+3, Bac+5 et plus
* Spécialité : A définir
* Expérience : Débutant, moins de 2 ans d’expérience, entre 2 et 5 ans d’expérience, ente 5 et 10 d’expérience, plus de 10 ans d’expérience
* Poste pour lequel vous postulez : Listes des postes

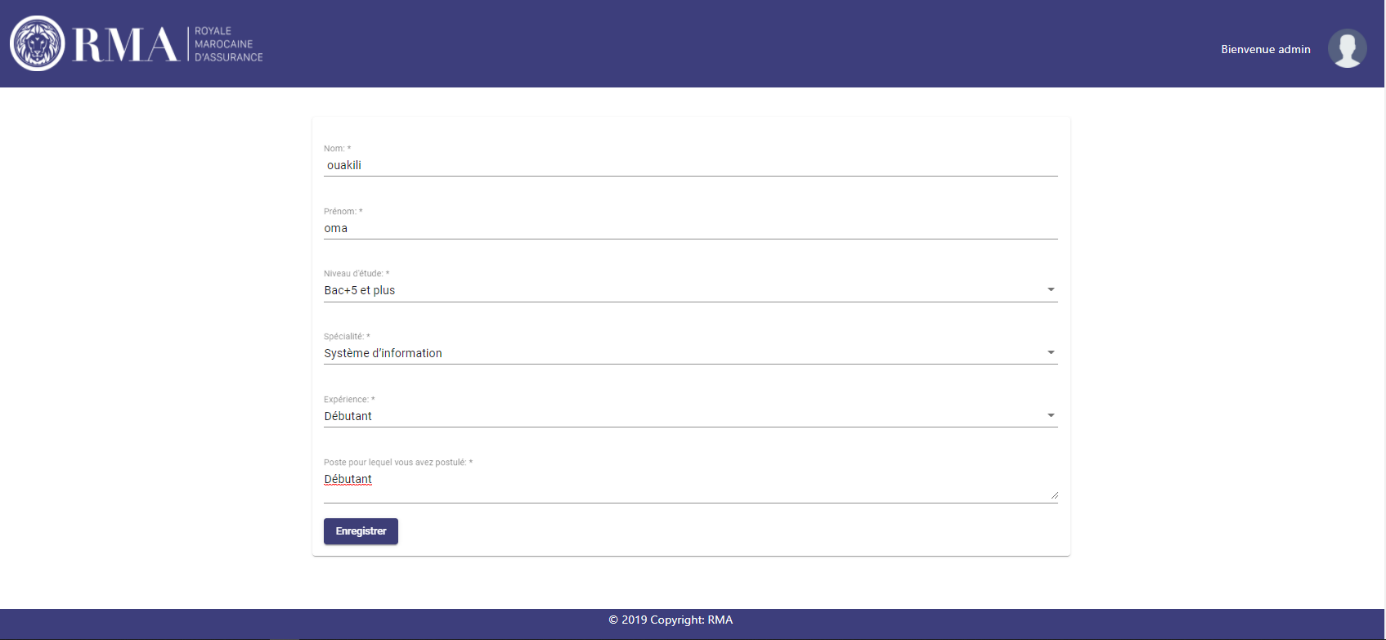


Figure : Inscription

### Explication sur le test

* Description textuelle comment le test se va passer

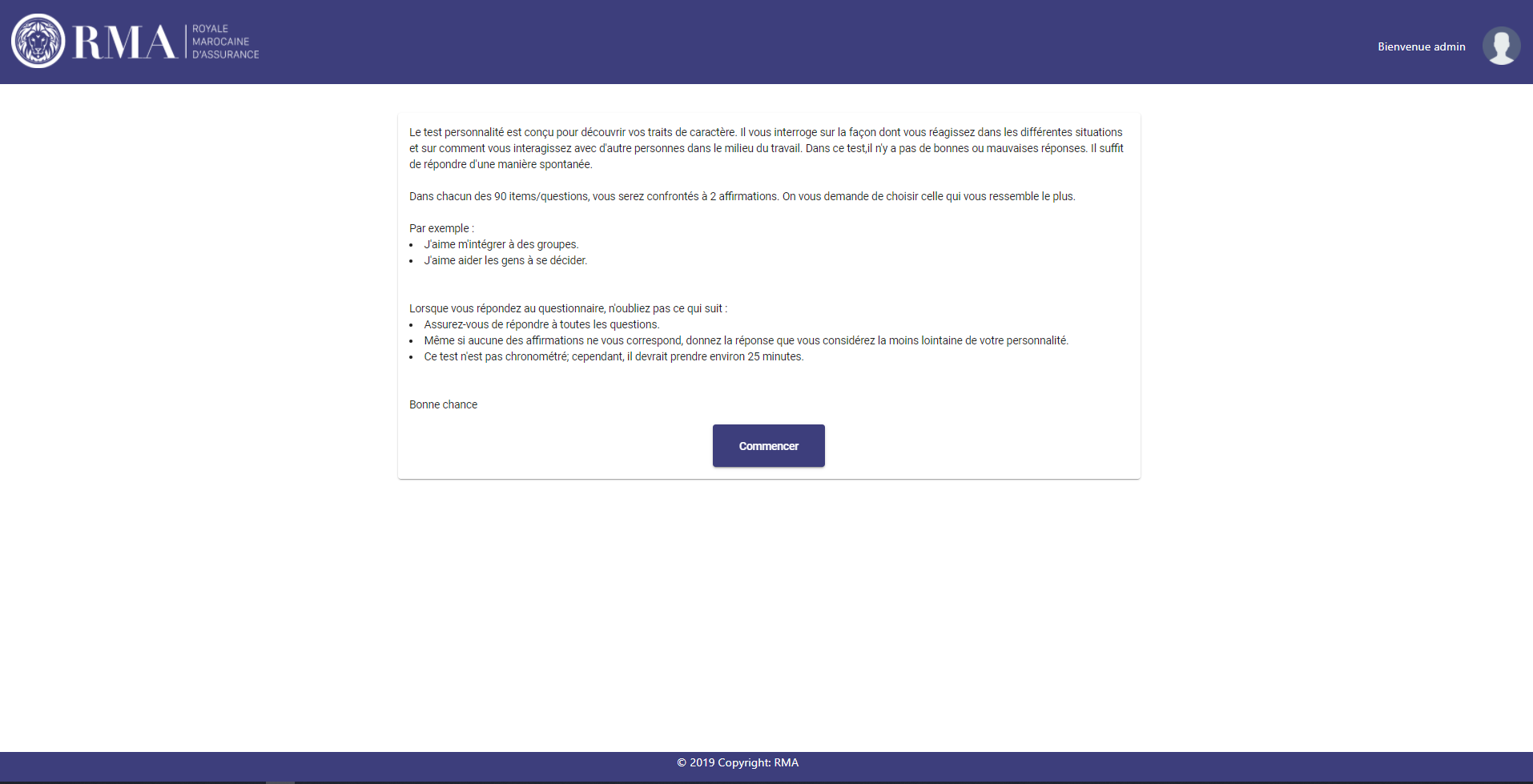


Figure : Explication sur le test

### Le test

* Cette figure représente l'interface où se déroule le test

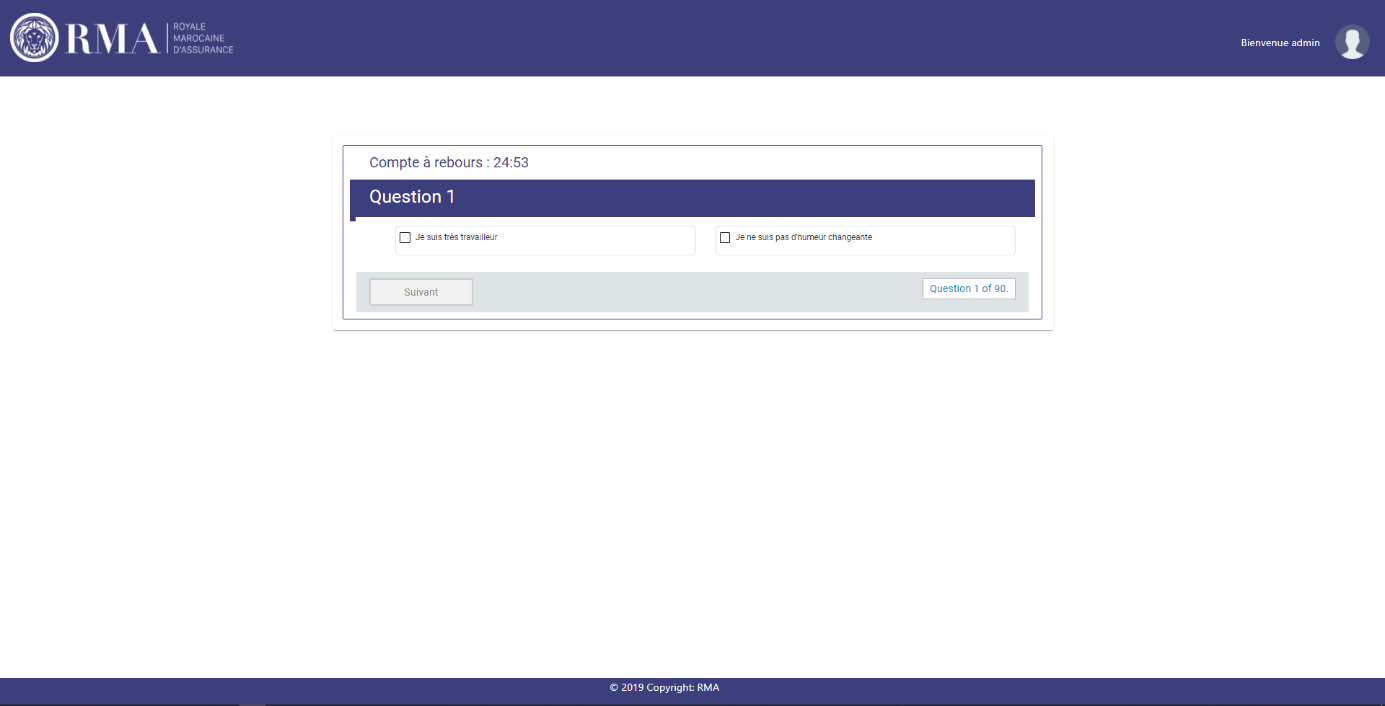


Figure : Début du test

### Remerciement

* Cette figure représente la fin du test



Figure : Remerciement

### Authentification admin

* Accès RH dont la finalité est la consultation et l’impression des résultats
* Nom d’utilisateur
* Mot de passe



Figure : Authentification admin

### List des candidats

* Cette figure représente l'interface la liste des candidats qui ont déjà passe le test avec l’accès à leurs résultats

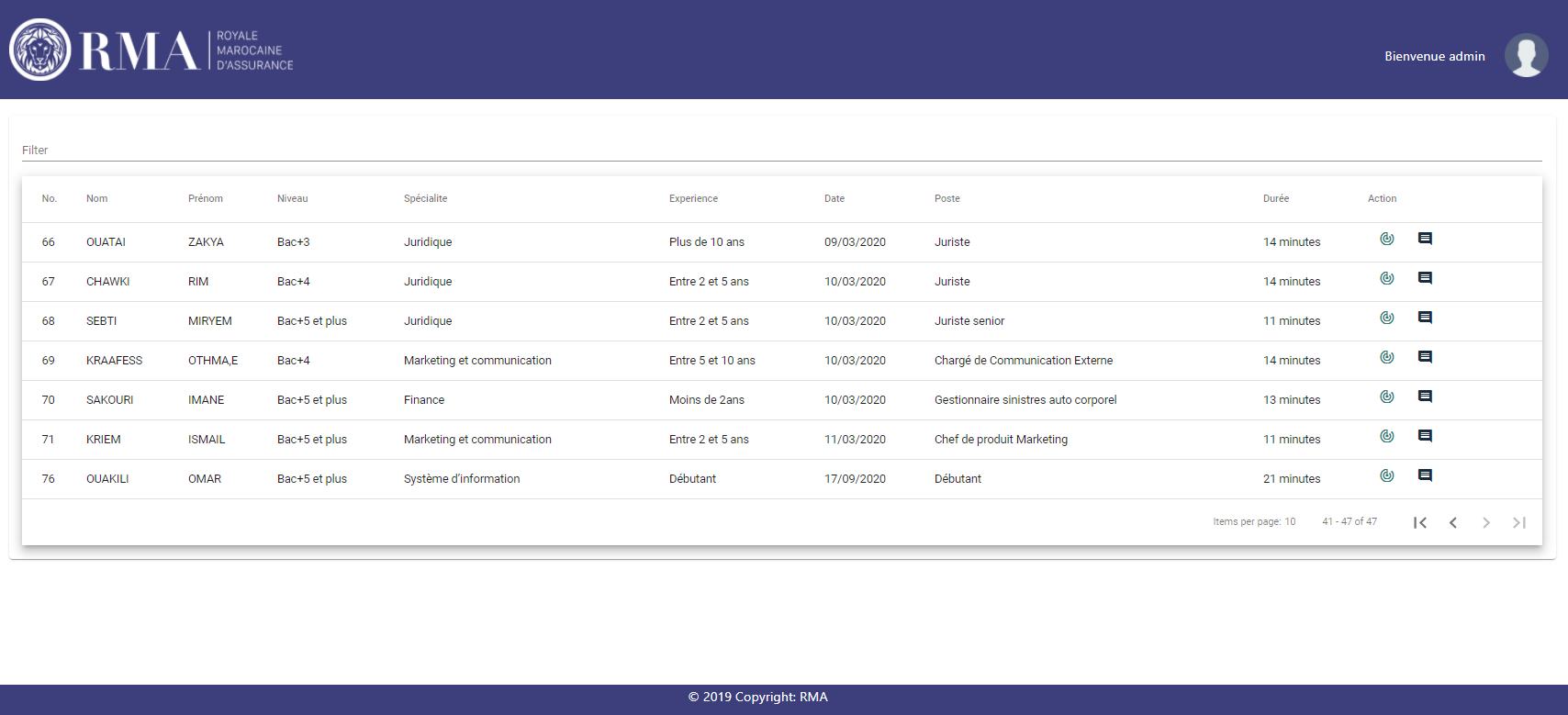


Figure : List des candidats

### Résultat

* Cette figure représente le résultat d’un candidat sous forme Rosace

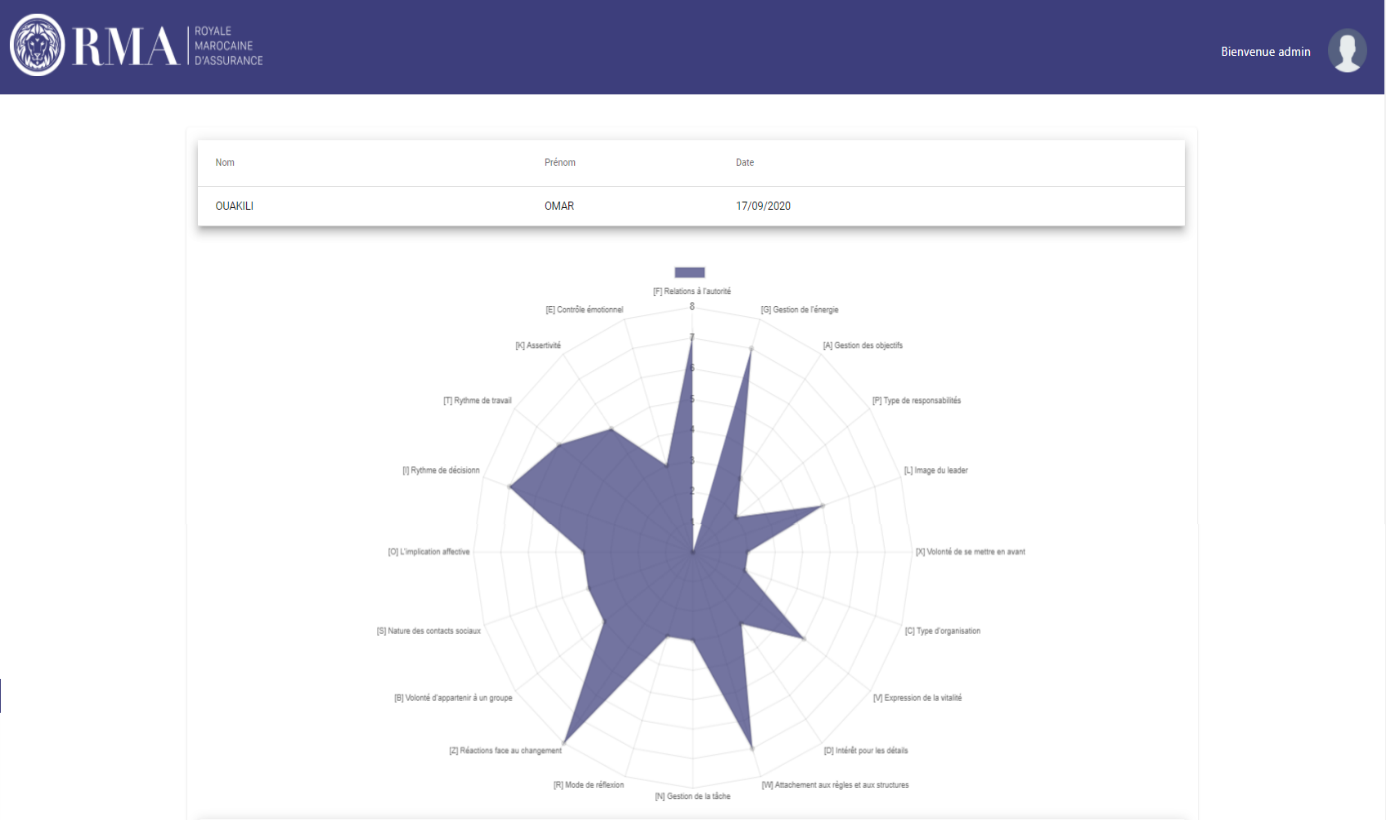


Figure : résultat

### La description des échelles

* Cette figure représente la description de chaque échelle de résultat

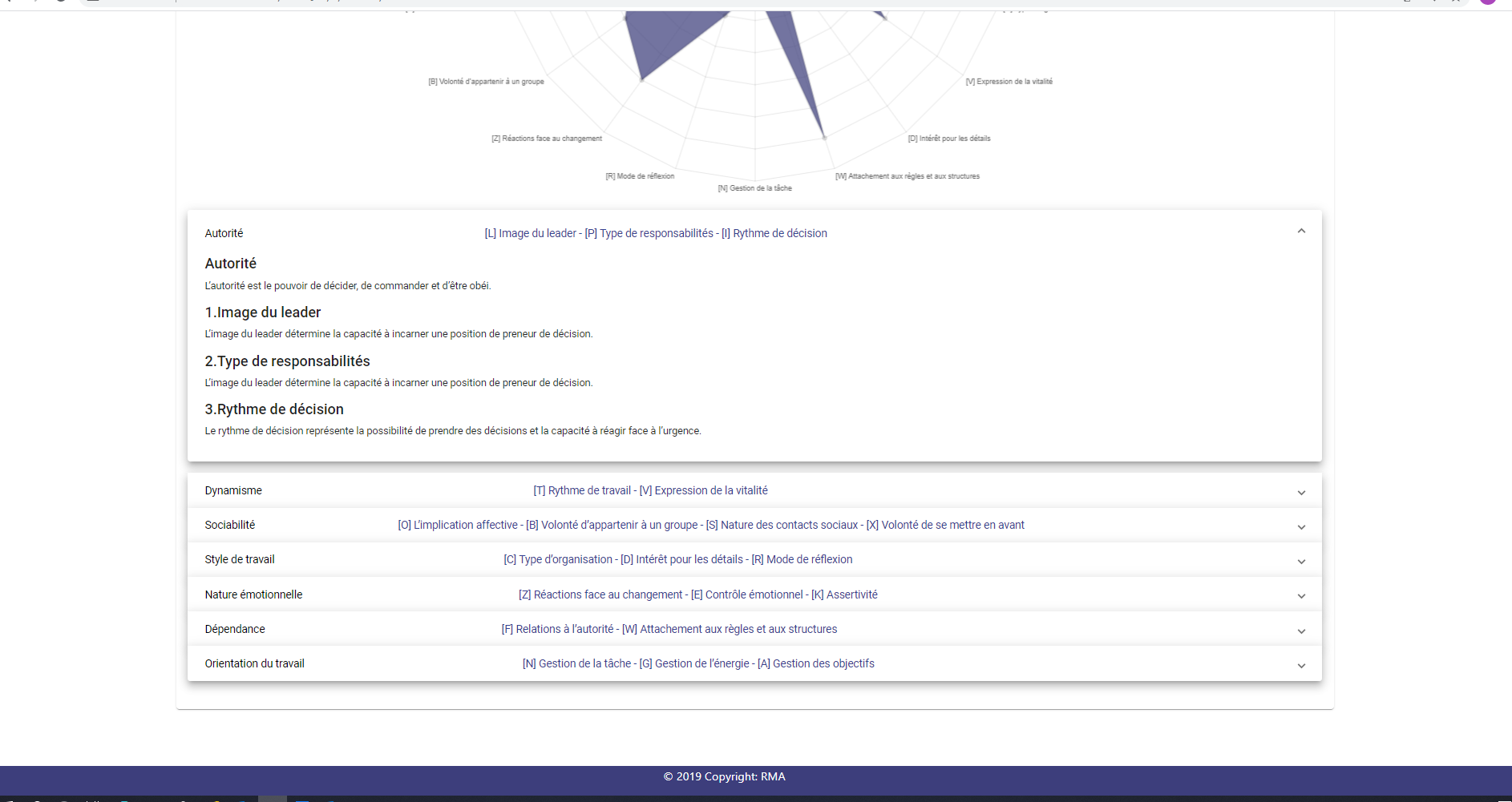


Figure : La description des échelles

### Ajouter des commentaires

* Cette figure représente l’interface qui permet au RH d’ajouté un commentaire sur le résultat du candidat

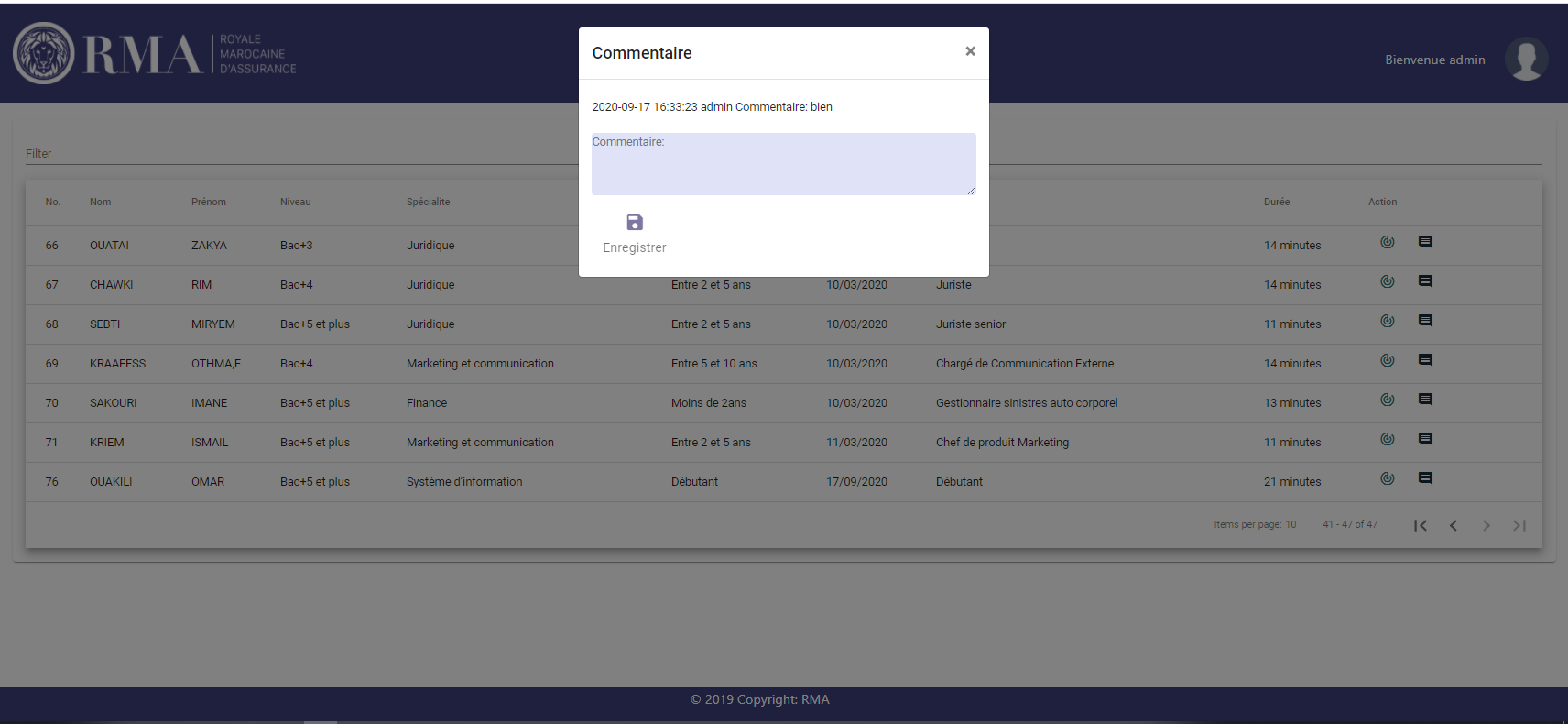


Figure : Ajouter des commentaire

**II. Conclusion**

Le dernier chapitre du projet a concrétisé tous les chapitres précédents, en présentant les fonctionnalités implémentées, ainsi qu’un aperçu des interfaces produites.

# Conclusion générale

J’ai présenté dans ce rapport le projet de fin d’étude réalisé au sein de la Royale Marocaine d’Assurance intitulé : « Mise en place d’une Solution Informatique de Test de Personnalité », ce stage s’est déroulé pendant la période du 2 Mars 2020 au 2 septembre 2020.

Afin de réaliser le travail demandé, j’ai essayé de respecter les bonnes pratiques du processus de développement logiciel. Il s’est avéré primordial de recenser d’abord les besoins fonctionnels et techniques de l’application. Cette étape nous a permis de passer à la conception de la solution en se basant sur le formalisme UML. La bonne gestion de la phase de réalisation et le respect des principes de la méthodologie cycle de vie en V , nous ont permis d’implémenter, tester et valider les différentes évolutions réalisées conformément aux spécifications et aux contraintes du projet définis dans le plan qualité projet.

L’architecture technique de la solution proposée est basée sur le principe de l’inversion de contrôle en se basant sur des Frameworks. Coté Backend, C’est l’architecture Java/JEE qui a été choisie en utilisant l’écosystème Spring Boot. Coté Frontend, c’est le Framework Angular qui a été choisi. Le choix de cette architecture permet aux développeurs de se concentrer sur les aspects métiers de l’application et déléguer aux Frameworks les aspects techniques tels que la sécurité et la gestion des performances.



Enfin les tests de l'application par le client sont concluants et répondent parfaitement aux besoins demandés.

# Bibliographie et webographie

* MVC ARCHITECTURE

[https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture)

[US/Apps/Fundamentals/Modern\_web\_app\_architecture/MVC\_architecture](https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture)

* SPRING SECURITY

<https://spring.io/guides/tutorials/spring-security-and-angular-js/>

* ANGULAR 7 FROM SCRATCH

<https://www.youtube.com/watch?v=f7mY4qEewVc&t=3199s>

* API REST Json Web Token (JWT)

[https://fr.slideshare.net/mohamedyoussfi9/scurit-des-applications-web-avec-json-web-token- jwt](https://fr.slideshare.net/mohamedyoussfi9/scurit-des-applications-web-avec-json-web-token-%20jwt)

* **Spring Boot and Angular**

[https://www.baeldung.com/spring-boot-angular-web](%20https://www.baeldung.com/spring-boot-angular-web)