



## **Master 2 MBDS de l'Université de Nice Sophia Antipolis (UNS)**

### **Travail Pratique Transversal**

**Mise en place d'un site et d'une application mobile sous Android de pari sur l'esport Dota2. Et une application lourde pour administrer le site.**

Du 17 Mai 2021 au 16 Août 2021

Réalisé par :

- RAKONTONDRAZAKA Ny Toky Mahery N°27
- RALAIARISOA Tolotriniavo Hubert N°29
- RAZAKANDRAINY Tohaniaina N°57

Tuteur à l'université :

Jury :

## **Résumé**

« Il est courant pour les étudiants en informatique d'exercer des pratiques sur les connaissances acquises dans les études académiques. »

Ceci pour initier les étudiants pour le monde professionnel, accroître ses capacités mais aussi pour les évaluer. Il nous a été confié de mettre en place une plateforme de pari en ligne. Notre choix s'est porté sur l'e-sport Dota2 comme point central de travail ; cela doit-être réalisé en 3 mois et comportera un site web, une application Android et un client lourd. La mise en place de la plateforme assure l'utilisation des technologies étudiées en MBDS comme : Angular, Oracle, NFC, QR code et aussi les méthodes de travail adaptés en entreprise.

Les attentes sont d'ordre que tous les besoins exprimés doivent-être répondus conformes aux exigences que ce soient : techniques, méthodes et analyses. Ce travail aidera les étudiants à mettre la main à la pâte pour bien appliquer les acquis théoriques.

**Mots-clés :** E-sport, Dota, Angular, Android, Winform, Oracle, MongoDB, QR-Code, Pari en ligne.

## **Abstract**

« It is common for computer science students to practice on the knowledge acquired in academic studies. »

This is to initiate students for the professional world, increase its capabilities but also to evaluate them. We were entrusted with the task of setting up an online betting platform. We chose Dota2 e-sport as the focus of our work; this is to be completed in 3 months and will include a website, an Android application and a Winform client. The implementation of the platform ensures the use of technologies studied in MBDS such as: Angular, Oracle, NFC, QR code and also the methods of work adapted in company.

The expectations are that all the needs expressed must be met in accordance with the requirements that they are: technical, methods and analysis. This work will help the students to get their hands dirty to apply the theoretical knowledge.

**Keywords:** *E-sport, Dota, Angular, Android, Winform, Oracle, MongoDB, QR-Code, Online betting.*

## Liste des figures et des tableaux

Figure 1 Scénario normal Inscription et Connexion.....	page 18
Figure 2 Scénario normal parier .....	page 20
Figure 3 Scénario normal Rechercher problème .....	page 22
Figure 4 Architecture système .....	page 25
Figure 5 Architecture en 3 couches .....	page 27
Figure 6 Structuration de la couche vue Angular .....	page 28
Figure 7 Structuration de la couche vue Android .....	page 29
Figure 8 Structuration de la couche vue Winform.....	page 30
Figure 9 Structuration de la couche contrôleur Java.....	page 31
Figure 10 Structuration de la couche contrôleur Node .....	page 32
Figure 11 Structuration de la couche métier .....	page 33
Figure 12 Modèle Conceptuel de Données .....	page 34
Figure 13 Cas d'utilisation pour parier .....	page 35
Figure 14 Cas d'utilisation pour finaliser un pari.....	page 36
Figure 15 Composant web.....	page 37
Figure 16 Déploiement web .....	page 37
Figure 17 Composant Android.....	page 37
Figure 18 Déploiement Android .....	page 38
Figure 19 Composant Winform .....	page 38
Figure 20 Déploiement Winform .....	page 38
Tableau 1 Comparatif des solutions au vue des critères.....	page 09
Tableau 2 Contraintes et risques du projet .....	page 13

## Liste des acronymes

**API** : Application Programming Interface

**BDD** : Base de Données

**DB**: DataBase

**DOTA**: Defense of the Ancients

**DSDM**: Dynamic System Development Method

**E-sport**: Electronic Sport

**ID** : Identifier

**IHM** : Interface Homme Machine

**JS** : Javascript

**MBDS** : Mobiquité Base de Données et Intégration de Systèmes

**MOBA**: Multiplayer Online Battle Arena

**MVC**: Model View Controller

**QR Code** : Quick Response Code

**VoIP**: Voice Over Internet Protocol

**Winform**: Windows Forms

## **Plan du document**

<i>1.Introduction générale.....</i>	<i>page 07</i>
<i>2.Présentation du Travail Pratique Transversal .....</i>	<i>page 07</i>
<i>3.Etat de l’art sur le sujet traité.....</i>	<i>page 08</i>
<i>4.Etude de l’existant et solution envisagée .....</i>	<i>page 09</i>
<i>5.Démarche projet.....</i>	<i>page 10</i>
<i>6.Exigences réalisées dans le projet (vision externe) .....</i>	<i>page 17</i>
<i>7.Architecture système.....</i>	<i>page 25</i>
<i>8.Conception du système logiciel réalisée dans le projet .....</i>	<i>page 25</i>
<i>9.Tests du système logiciel .....</i>	<i>page 39</i>
<i>10.Conclusion générale .....</i>	<i>page 39</i>

## 1. Introduction générale :

« *Supporter son équipe c'est bon et normal, mais gagné avec son équipe c'est encore meilleur.* » Actuellement, en master 2 en MBDS pour devenir spécialiste que ce soit en : Mobiquité, BigData ou Intégration Système, .. Dota 2 est un pionnier que ce soit dans le MOBA ou l'e-sport dont il récompense les joueurs professionnels à plus de 40 millions de dollars pour cette année, non seulement cela mais notre choix est aussi justifié que les joueurs de Dota à Madagascar sont nombreux de façon exponentiels. Notre objectif est donc la réalisation d'une plateforme de pari en ligne sur Dota2. Mettre en œuvre tous les moyens techniques, les recherches et humaines pour achever l'objectif fixé. L'équipe se divisera les tâches en soit : assurer un travail conforme aux exigences donnés, l'échange entre client et équipe technique, exécuter les tâches selon le délai.

## 2. Présentation du Travail Pratique Transversal :

### 2.1. Présentation de l'équipe de développement

Notre équipe est composé de trois jeunes développeurs passionnés :

RAKOTONDRAZAKA Ny Toky Mahery

RALAIARISOA Tolotriniavo Hubert

RAZAKANDRAINY Tohaniaina

### 2.2. Présentation du sujet et objectifs du projet

Nous voulons réaliser une plateforme de paris en ligne sur Dota 2. Actuellement la majorité de pouvoir effectuer des paris sur le sport se fait en allant physiquement dans les endroits destinés à ce fait. Et que les sports proposés sont les classiques. Alors d'où notre projet de proposer cela en ligne sans se déplacer à tout moment ni la possibilité de perdre son ticket de pari et proposer de parier sur de l'e-sport qui est très émergent en ce moment. Les enjeux sont de proposer une solution en ligne et complète du traditionnel pari et attirer des nouveaux parieurs et faire changer d'habitude les autres.

### 3. Etat de l'art sur le sujet traité

Dans cette partie une étude sera donnée en comparaison des différentes solutions déjà proposées en rapport de la plateforme de pari en ligne que nous avons réalisée.

#### 3.1. Critères de comparaison

Les critères de comparaison pour effectuer dans nos études :

**Praticité, Efficacité, Simplicité, Ere numérique.**

#### 3.2. Etude de chaque solution au vu des critères

Les solutions proposées sur chaque critère :

Magasin de pari :

**Praticité**, aller dans un magasin de pari est assez pratique au vu du temps que cela doit dépenser, souvent rempli, déconseiller au vue de la pandémie.

**Efficacité**, la procédure de pari dans un magasin est souvent fastidieuse, prendre son tiquet, utiliser de l'argent liquide, attendre les matchs pour encaisser.

**Simplicité**, la compréhension des paris dans les magasins ne sont pas simple à cause de la manque d'assistance et explication.

**Ere numérique**, les magasins de pari ne suivent pas l'évolution du monde numérique et ce que internet peut offrir, ainsi bloquent les nouveaux potentiels parieurs

Plateforme de pari en ligne :

**Praticité**, la plateforme n'est pas contraint par le temps et l'espace, restez chez soi et pouvoir effectuer son pari

**Efficacité**, les écrans sont partout ainsi la plateforme est avec les parieurs où qu'ils soient. Que ce soit sur un smartphone ou un ordinateur. Pas de tiquet à prendre ou utilisation de l'argent liquide, tout se fait numériquement.

**Simplicité**, en quelques clics, effectuer votre pari, pas d'attente à la fin du match, le client sera notifié et son argent sera automatique versé.

**Ere numérique**, la plateforme propose une solution dans la tendance de l'internet, attractive pour des potentiels parieurs.



### 3.3. Tableau comparatif des solutions au vu des critères

Critères	Magasin de pari	Plateforme de pari en ligne
Praticité	Nécessite un déplacement	Non contraint par le temps et l'espace
Efficacité	Procédure fastidieuse	Tout se fait numériquement
Simplicité	Manque d'assistance	En quelques clics, parier et comprenez les résultats
Ere numérique	Moins de présence	Solution à l'ère du digitale

Tableau 1 : Comparatif des solutions au vu des critères

## 4. Etude de l'existant et solution envisagée

### 4.1. Etude de l'existant

#### 4.1.1. Description externe du système logiciel

Avant que le projet ne soit réalisé, aucun existant n'a été constaté sur la vision externe.

#### 4.1.2. Description interne du système logiciel

Avant que le projet ne soit réalisé, aucun existant n'a été constaté sur la vision interne.

### 4.2. Critique de l'existant

Sur la critique de l'existant nous allons prendre en référence les magasins de pari classiques.

Les points forts :

- Les magasins sont souvent tenus par les instituts de pari puissant et facile à reconnaître
- Des clients fidèles dans ces magasins
- Une bonne ambiance présente dans ces magasins

Les points faibles :

- Moins d'assistance
- Moins de présence sur internet et sur les évolutions
- En vue de la pandémie, la forte présence des personnes dans les magasins est totalement déconseillée.

### 4.3. Solutions envisagées

C'est dans cette optique que la solution en ligne s'est posée.

La plateforme de pari en ligne, apportera une solution simple et efficace aux clients. Sans se déplacer, effectuer son pari est possible. Que le client soit sur son ordinateur ou sa tablette. Ou tout simplement un futur client qui veut visiter ou regarder c'est quoi un site de pari ; le site web répondra à ses besoins.

Pour les clients fidèles qui veulent suivre à tout prix les paris, recevoir des notifications, ne rien rater, l'application Android est la solution qui convient à ces clients.

### 4.4. Objectifs principaux et livrables

Développer et réaliser la plateforme dans le délai tout en respectant les besoins exprimés.

Le site web hébergé sur Heroku

Le package de l'application Android

L'exécutable du client lourd Winform

## 5. Démarche projet :

### 5.1. Principes de la démarche projet

#### 5.1.1. Activités d'ingénierie logicielle

L'ingénierie logicielle couvre en particulier aux activités clés du cycle de vie du logiciel et dans l'ordre dans lequel nous avons effectués ces activités. Ces activités clés du cycle de vie d'un logiciel sont :

***L'analyse fonctionnelle, l'architecture, la programmation, les tests, la validation et la maintenance.***

**L'analyse des besoins :**

Récoltant des informations détaillées sur les fonctionnalités que le logiciel devra offrir. L'analyse fonctionnelle a été déjà fait en amont et présent dans le Cahier des Charges fournissant les travaux et les résultats attendus.

#### **L'architecture :**

Sur le plan conceptuel, étape auquel divers formalismes de modélisations, et de conception logiciel seront appliqués au travail.

Sur les modélisations le document de conception en parle plus en détails.

Sur les grandes lignes l'architecture se présente en trois parties :

***L'interface utilisateur, la couche métier et l'accès aux données.***

#### **La programmation :**

La programmation des codes sources suit les plans initialement établis sur l'architecture, la conception et la méthode DSDM qui offre une flexibilité dans la revue de la conception sans modifier l'architecture et partir de zéro

#### **Les tests et la validation :**

Une étape de vérification et test faites, des tests unitaires, par les développeurs servant à valider si les codes écrites répondent aux besoins, trouver les erreurs de programmation, les bugs et des résultats inattendus.

#### **Maintenance :**

Une suite d'opération de post-production, visant à établir des correctifs et des futurs améliorations que ce soit une vision interne ou externe.

### **5.1.2. Méthode de gestion de projet**

Le choix de l'équipe de développement sur la gestion du projet s'est porté sur la méthode DSDM (*Dynamic System Development Method*).

*Sur l'application de la méthode et la démarche de communication ;*

**Réunions :** Tous les débuts de la semaine et tous les 3 jours pour revoir les tâches à réaliser prochainement.

**Compte rendus :** émettre des comptes rendus soit dans une réunion, soit après un bloc de tâche réalisé, pour évoquer les erreurs, bugs trouvés, une difficulté ou une conception à revoir.

**Test** : analyser si l'implémentation répond aux demandes fonctionnelles et garantir le bon fonctionnement de l'application.

**Coopération** : une partie très important, l'implication des clients/utilisateurs durant le cycle de développement pour revoir les besoins et leur visions vis-à-vis des travaux en cours et déjà effectués.

### **5.1.3. Rôles et responsabilités**

Pour bien mener à terme les tâches à réaliser 3 parties prenantes sont prises en compte :

Client/Utilisateur : exprimant leurs besoins et visions en rapport de leurs attentes du projet.

Product Owner : assure et garantit la communication entre client/utilisateur que les demandes de ces derniers sont effectuées par l'équipe de développement.

Equipe de projet : effectue les tâches techniques pour réaliser le projet.

### **5.1.4. Outils**

Pour bien assurer les rôles et la communication, ces outils ont été choisis pour faciliter l'entente, la communication et la continuité des travaux.

Discord ; logiciel gratuit de VoIP et de messagerie instantanée pour les réunions ou visio-conférence entre le client et l'équipe de projet, ou entre équipe de projet.

Google Docs ; un workspace en ligne pour créer et modifier des documents, en collaboration avec le client et l'équipe de projet et en interne.

Trello ; outil de gestion de projet en ligne pour donner les tâches et avoir un suivi au cours du cycle de développement.

Github ; service de repository en ligne et gratuit, notamment pour gérer les versions, et l'hébergement des codes sources.

### **5.1.5. Gestion de la configuration**

L'organisation des livrables se fait en trois parties, présentant chacun la partie de la plateforme.

Le site web : hébergé sur Heroku et mise à jour par git.

Le package d'installation de l'application Android.

L'exécutable du client lourd Winform

La règle de nommage des fichiers se fait en référence des numéros de l'équipe de projet : ***project\_filenamesxxx\_272957***.

Sur les sauvegardes et versions, dès que l'équipe finit un bloc de tâche ou modification après une validation l'envoie sur github.

Les documents du projet ; se trouve sur google docs pouvoir modifier facilement.

## 5.2. Contraintes et risques sur le projet

N° du risque	Libellé du risque	Priorité	Importance	Facteur Contribuant	Solution proposée	Statuts
1	Ne pas pouvoir tester la partie accès donnée	Haute	Haute	1.1 L'activation du compte oracle cloud peut prendre plus de 72 heures	1.1 Créer le compte Oracle cloud très tôt	Réalisé : Choisir un serveur oracle avec moins de trafic
2	La différence des champs ID fournit pour les deux API	Moyen	Moyen	1.2 Utilisation de deux API différents.	1.1 On a insérer manuellement toutes les teams Dota professionnels.	Réalisé : Vu l'importance des teams Dota dans les paris.
3	Résultat de pari introuvable	Haute	Haute	1.2 L'API n'est pas géré par l'équipe de projet	1.1 Le système renvoie un mail automatiquement vers le BackOffice/Admin. 1.2 L'Administrateur insère manuellement les résultats des paris.	Réalisé : Vu la haute importance des résultats des paris

Tableau 2 : Contraintes et risques du projet

### 5.3. Démarche projet mise en œuvre

Dans le cadre du choix DSDM comme méthode de gestion de projet, voici les étapes en 9 parties :

**Implication des utilisateurs** -> ils doivent être présent tout au long du développement du projet

**Autonomie** -> L'équipe est autonome sur le développement et doit avoir un droit de décision imparable sur l'évolution des besoins.

**Visibilité du résultat** -> Livraison fréquente de l'application afin d'obtenir des feedbacks fréquents.

**Adéquation** -> L'application livrée doit être en adéquation avec le besoin des utilisateurs.

**Développement itératif et incrémental** -> Le produit doit constamment s'améliorer avec le feedback des utilisateurs.

**Réversibilité** -> En cas de besoin, tout développement doit pouvoir revenir en arrière.

**Synthèse** -> Un schéma directeur fixe le projet dans son ensemble que ce soit au niveau vision ou au niveau objectifs

**Tests** -> Automatisation de tests fréquent afin de garantir une non-régression au sein de l'application.

**Coopération** -> L'ensemble de l'équipe ou parties prenantes sur le projet doivent accepter d'éventuelles modifications des fonctionnalités demandées.

#### **5.4. Planification**

Le Diagramme de GANTT est fournit dans un fichier annexe nommé  
***Diagramme de Gantt TPT.xls***

## **5.5. Budget du projet**

Le coût de projet vis-à-vis à l'équipe de développement :

Locaux : 500.000 Ar/mois sur 3 mois

Indemnités : 300.000 Ar/mois sur 3 mois

Machines : 2.000.000Ar

Frais de connexion internet : 1.500.000 Ar

Travaux et recherches : 200.000 Ar



## **6. Exigences réalisées dans le projet (vision externe/utilisateur)**

### **6.1. Exigences fonctionnelles**

#### **6.1.1. Cas d'utilisation Inscription et Connexion**

Ce cas d'utilisation décrit comment un utilisateur s'enregistre et se connecte sur le site web.

Préconditions : le client ayant un ordinateur connecté à internet et un navigateur web et l'url du site.

Descriptions cas normal :

Le cas d'utilisation commence lorsqu'un client consulte le site. Un utilisateur non connecté sur le site peut consulter les informations sur les prochains matches mais ne peut pas effectuer des paris.

L'utilisateur clique sur le bouton SIGNUP dans la barre de navigation, puis le site lui retourne la page d'inscription avec les champs de saisis : Nom, Prénom, Pseudo, Mot de Passe et un bouton INSCRIPTION.

L'utilisateur remplit les champs de saisies puis appuie sur le bouton INSCRIPTION, l'utilisateur reçoit une notification de son inscription.

Le site web lui renvoie sur la page LOGIN avec les champs de saisis : Pseudo, Mot de Passe et un bouton CONNEXION.

L'utilisateur remplit les champs de saisis sur son Pseudo et son Mot de Passe et appuie sur le bouton CONNEXION.

Le site web renvoie l'utilisateur sur une page d'accueil personnalisé.

Déroulements alternatifs :

L'utilisateur ne remplit pas tous les champs de saisis lors de l'inscription, le système lui demande de tout remplir.

Le pseudo fourni par l'utilisateur est déjà pris par un autre utilisateur, le système lui demande un autre pseudo.

Lors du Login, l'utilisateur saisit un mauvais pseudo ou un mauvais mot de passe le système lui demande de les remplir correctement.

### Scénario normal Inscription et Connexion

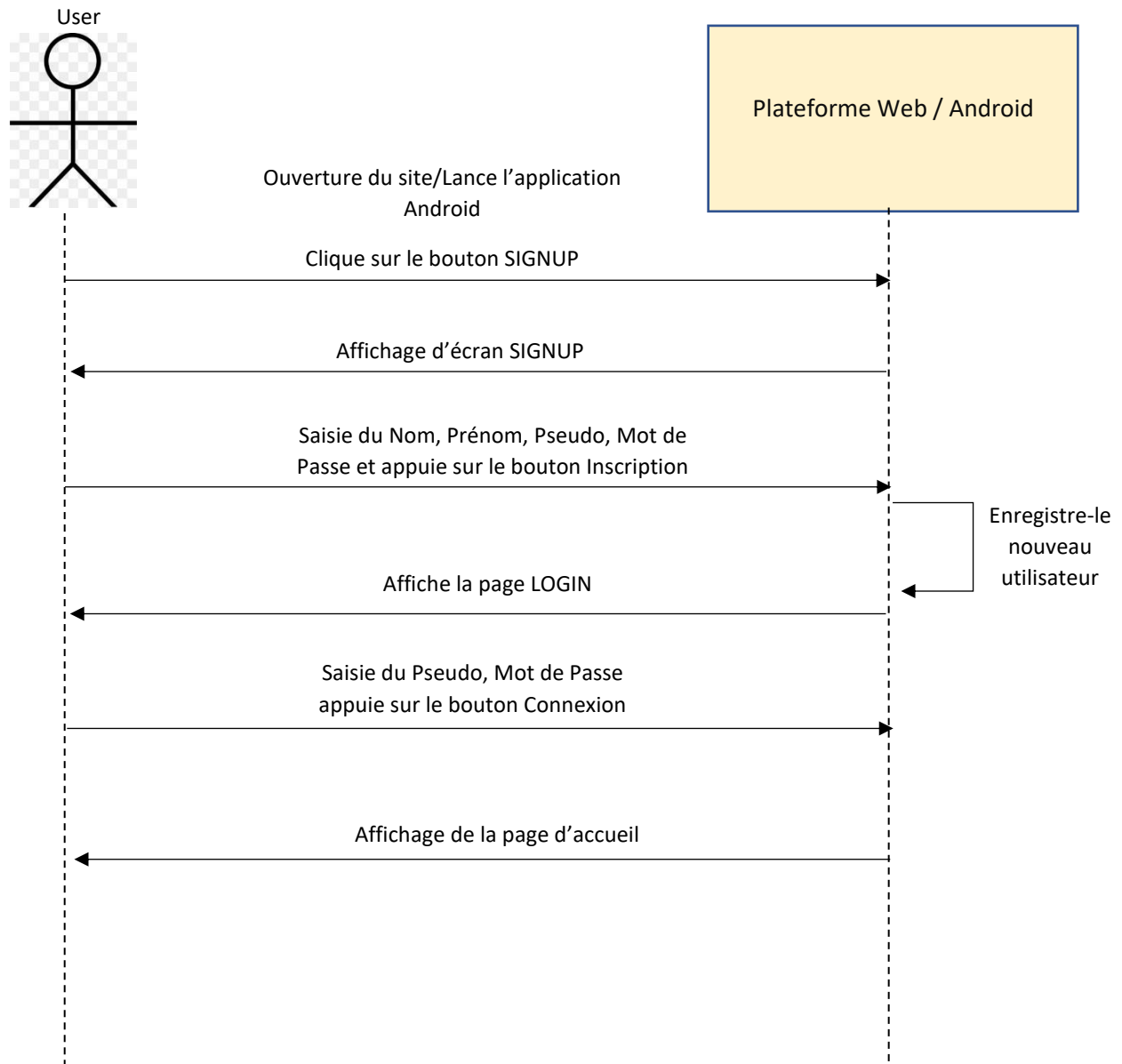


Figure 1 : Scénario normal Inscription et Connexion

### 6.1.2. Cas d'utilisation de Parier

Ce cas d'utilisation décrit comment un utilisateur effectue un pari sur le site web.

Préconditions : le client ayant un ordinateur connecté à internet et un navigateur web, l'url du site et un compte enregistré sur le système.

#### Descriptions cas normal :

L'utilisateur clique sur le bouton LOGIN dans la barre de navigation, puis le site lui retourne la page LOGIN avec les champs de saisis : Pseudo, Mot de Passe et un bouton LOGIN.

L'utilisateur remplit les champs de saisies puis appuie sur le bouton CONNEXION.

Le site web lui renvoie sur la page d'accueil personnalisé.

L'utilisateur clique sur le bouton Parier, le site lui renvoie sur la page paris contenant les prochains matches à venir, que l'utilisateur peut effectuer un pari.

L'utilisateur clique et choisit un match, le site lui renvoie une page sur les détails du match.

L'utilisateur choisit de faire un pari *MapOverAll*, un pop-up s'affiche demandant à l'utilisateur de saisir un montant pour son pari.

L'utilisateur saisit un montant et valide le pari.

La page lui notifie que son pari a été bien enregistré.

#### Déroulements alternatifs :

Lors du pari, l'utilisateur ne remplit pas le champ de saisi du montant pour valider le pari alors impossible pour l'utilisateur de parier.

Lors du pari, l'utilisateur remplit le champ de saisi du montant a une valeur supérieure à son solde le système lui renvoie solde insuffisant.

### Scénario normal parier

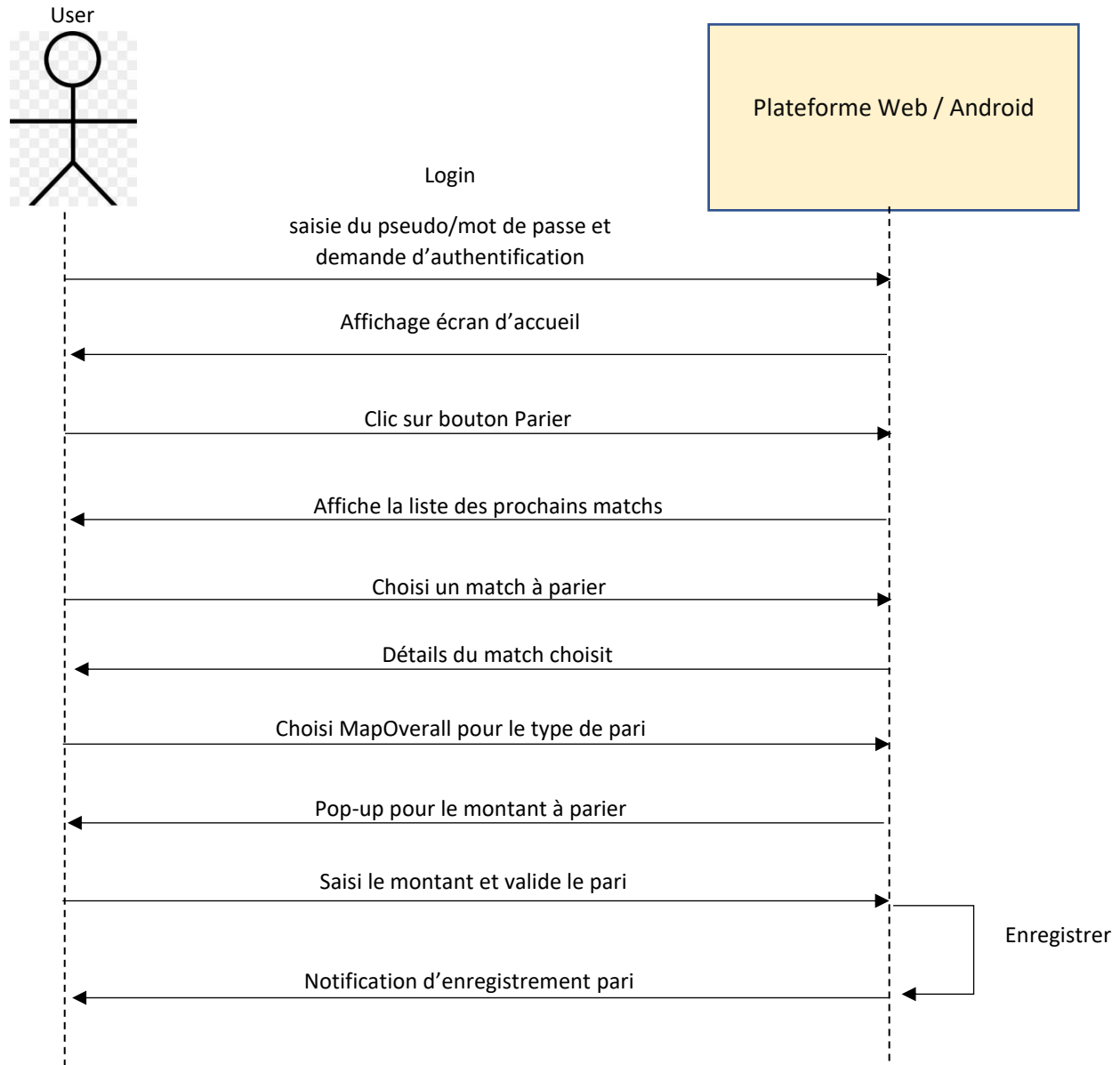


Figure 2 : Scénario normal parier

### 6.1.3. Cas d'utilisation Recherche de problème Backoffice

Ce cas d'utilisation décrit comment du côté administrateur on résout les problèmes des matches introuvables dans l'API.

Préconditions : l'administrateur ayant un ordinateur connecté à internet et/ou le client Winform sur son ordinateur, un navigateur web et l'accès administrateur sur la plateforme.

Descriptions cas normal :

L'administrateur consulte la partie réservée/ ou l'application winform.

Le système lui renvoie la page d'accueil.

L'administrateur clique sur le bouton *Liste des problèmes*.

Le système lui renvoie la liste des problèmes.

L'administrateur choisit un problème et effectue si un pari est gagnant ou perdant.

Le système traite le changement sur le compte du client, son solde, qui a parié sans résultat.

Ensuite le système notifie l'administrateur sur le traitement avec succès.

Déroulements alternatifs :

L'utilisateur ne remplit pas tous les champs de saisis lors de l'inscription, le système lui demande de tout remplir.

Le pseudo fournit par l'utilisateur est déjà pris par un autre utilisateur, le système lui demande un autre pseudo.

Lors du Login, l'utilisateur saisit un mauvais pseudo ou un mauvais mot de passe le système lui demande de les remplir correctement.

### Scénario normal recherche problème dans BackOffice

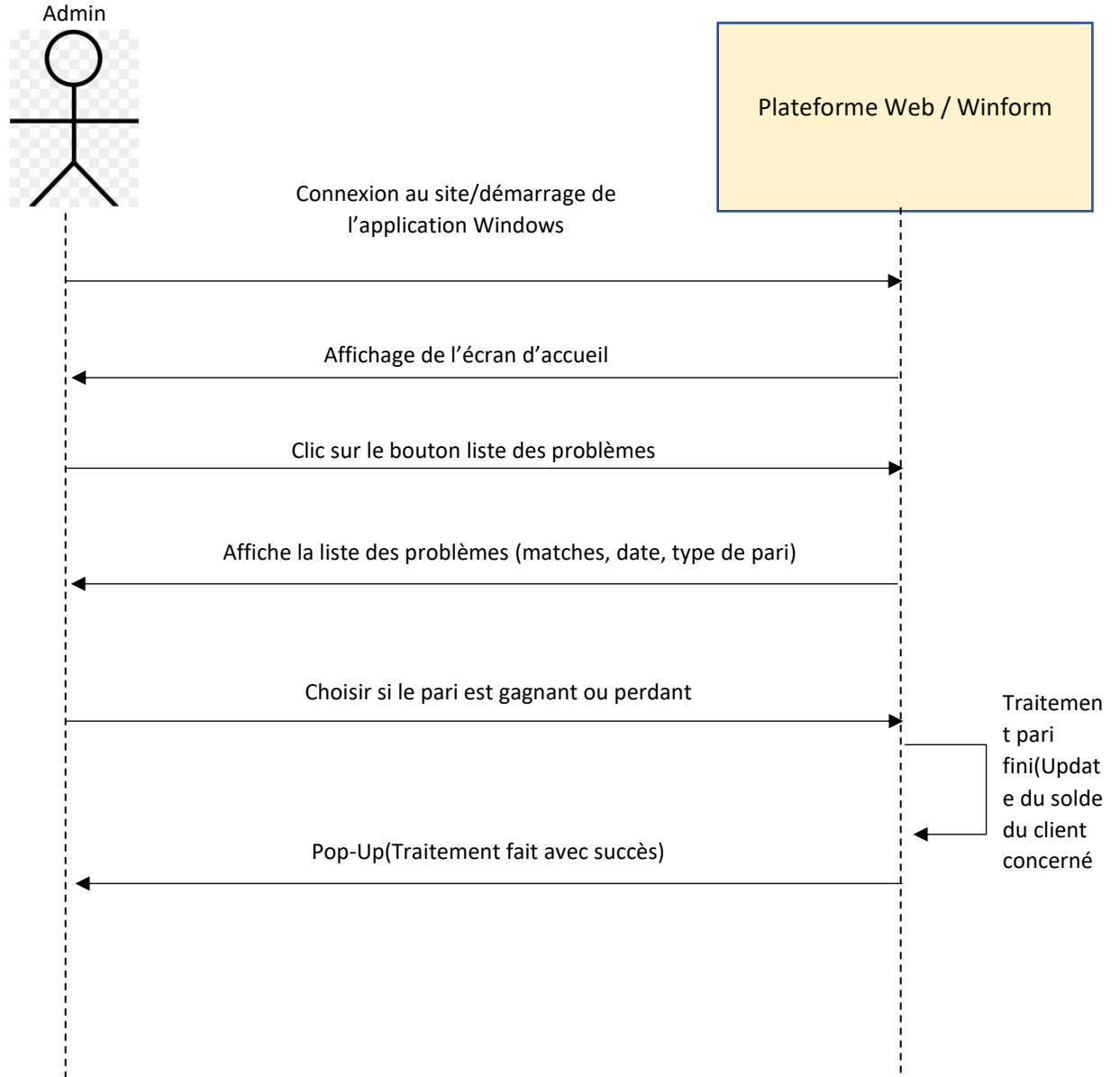


Figure 3 : Scénario normal Rechercher problème dans BackOffice

## 6.2. Exigences non-fonctionnelles transverses

### Utilisabilité :

Sur les chartes graphiques, la plateforme est bien agencée, couleurs vives, aucune information inutile, les boutons sont facilement trouvables et clair. La prise en main est très facile, avec une interface de façon Flat Design et suivant le thème Dota2.

### Performances :

La plateforme a été optimisée dans toutes ses demandes, que ce soit demande de l'utilisateur ou traitement du système.

Pendant le développement l'équipe a pu optimiser la plateforme de 70% de vitesse sur le temps de réponse en se basant sur les bonnes pratiques pour optimiser Oracle Cloud.

### Robustesse :

La plateforme a été conçue avec une architecture qui tolère les pannes.

Rendre disponible aux utilisateurs 24h/24

Selon nos études, 87% des pannes qui peuvent survenir proviendraient des serveurs dont l'équipe est indépendante.

### Sécurité :

Les données des utilisateurs sont chiffrées, aucune de ses données n'est vendue à des tiers.

### Support :

La conception et l'architecture en amont permettra une maintenance dans le plus bref délai et une possibilité d'évolution.

### **6.3. Interfaces détaillées**

#### **6.3.1. IHM**

Les interfaces hommes machines sont à détailler dans le Cahier des charges fourni à l'équipe de développement.

#### **6.3.2. Interfaces avec d'autres systèmes**

Le système logiciel interagit notamment sur deux API Dota différents :

Open Dota : une API open source et gratuit qui permet d'accéder aux données des matches de Dota Professionnel, et les statistiques des joueurs.

Open Dota est utilisée par la plateforme pour avoir les résultats des matchs et les informations des équipes.

Rivalry API : une API gratuit qui permet de calculer les côtes des prochains matchs et les détails des matchs professionnels.

Rivalry est utilisé par la plateforme pour offrir les côtes aux utilisateurs et recevoir les ID des matchs en cours.



## 7. Architecture système

Présentation de l'architecture logicielle appliquée au projet :

Une architecture trois tiers :

- **L'interface utilisateur concerne l'expérience utilisateur de l'application**
- **La logique métier permet de fournir l'accès aux données, aux services dont dépend l'application**
- **L'accès données, garantit le stockage des données.**

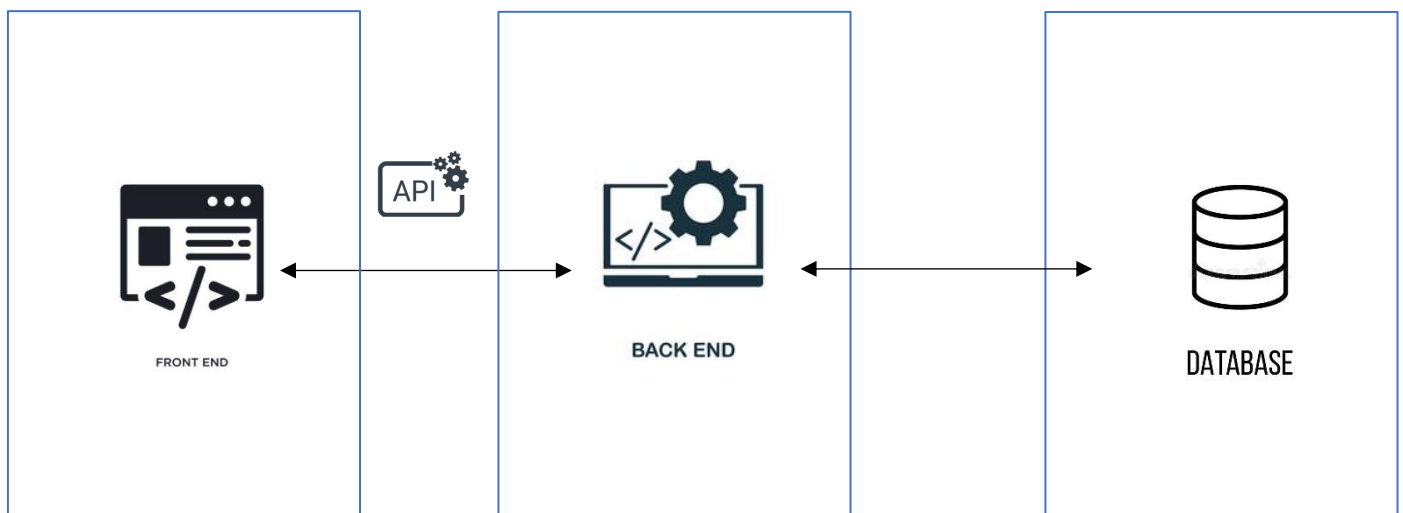


Figure 4 : Architecture système

## 8. Conception du système logiciel réalisée dans le projet

### 8.1. Plateforme technique

- Architecture matérielle : 1 serveur d'application Java – Spring / 1 serveur Javascript – Node.Js
- Framework logiciel : technos Angular/ Android Native/ C# .Net– Oracle et Mongo comme BDD

### 8.2. Conception du logiciel développé

#### 8.2.1. Conception du code source

Les choix sur la structuration des unités de code :

Définition de l'architecture du code en pattern en couches MVC

La conception des attributs et les associations entre classes

La conception des traitements.

Le choix d'affiner la structuration du code :

- *Affiner la structuration en packages*
- *Factoriser du code avec la généralisation*
- *Communication entre classes et dépendances*
- *Gérer les états*

La conception du code source se définit par les objectifs voulus mais aussi assurer une qualité ( : réponse aux besoins demandées, maintenable et évolutive) et une productivité ( : sur le moyen/long terme ; facile à maintenir, possibilité de changer d'architecture sans repartir à zéro).

*Sur les bonnes pratiques de conception :*

\* Séparation des préoccupations en unités de code :

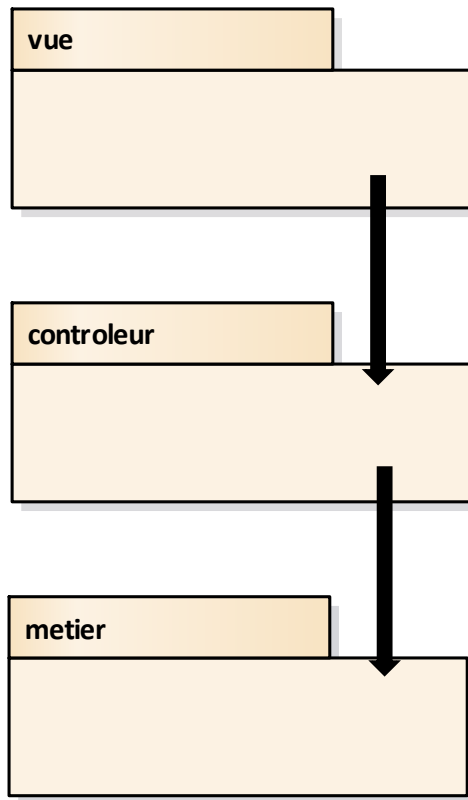
- Chaque unité de code possède une responsabilité claire, reflétée par son nom, et non partagée.
- Les données et les traitements associés sont définis dans la même unité, qui est responsable de l'intégrité de ses données.
- Séparer les préoccupations d'affichage des préoccupations de gestion des données métier

Sur les règles de nommage :

- Le nom de l'unité doit être toujours clair et reflète son attribut ou ses fonctions.

### **8.2.2. Le code source – vue statique**

Pour assurer une bonne maintenabilité et évolution à notre code source, une architecture en 3 couches :



*Figure 5 : Architecture en 3 couches*

## Structuration de la couche vue

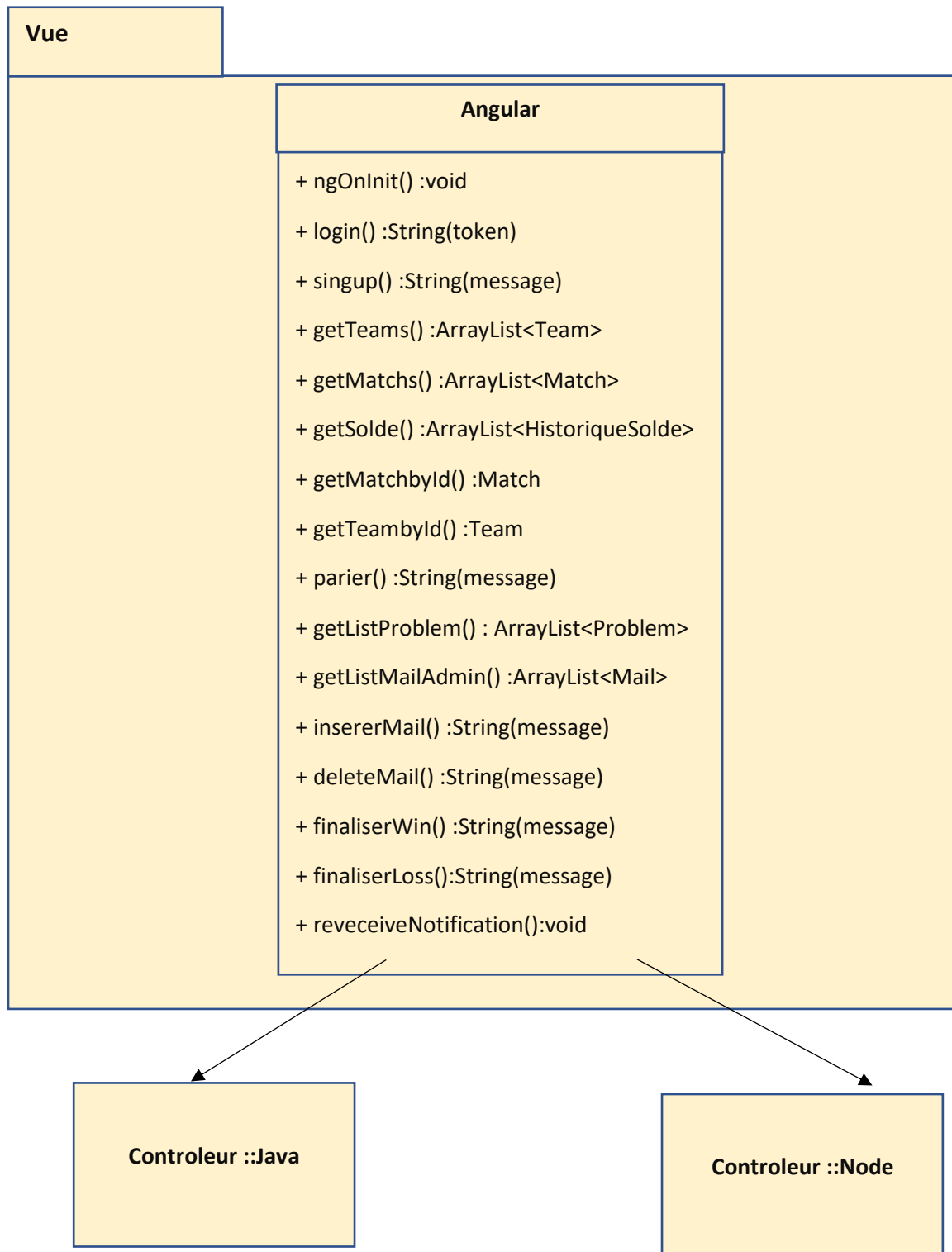


Figure 6 : Structuration de la couche vue Angular

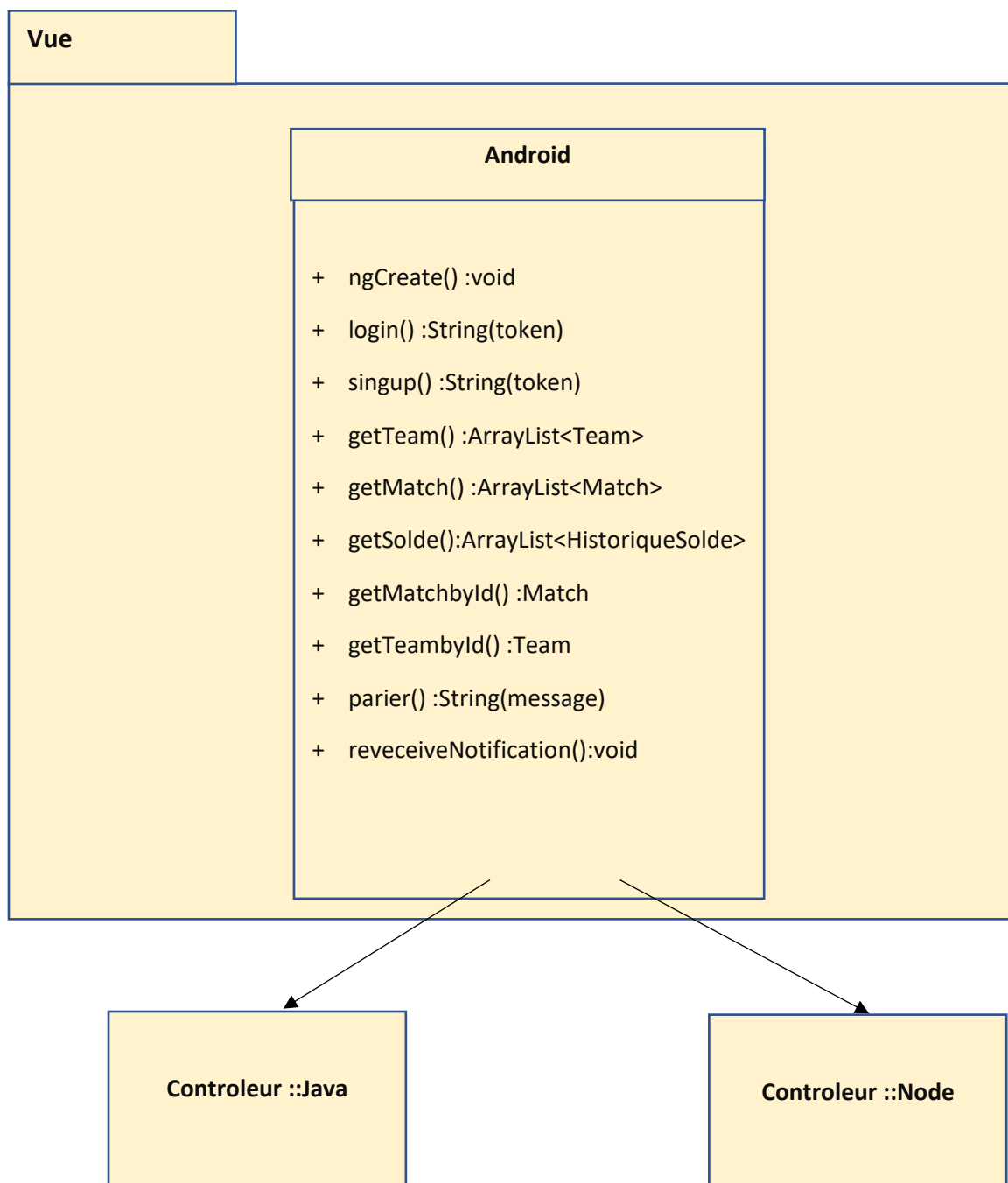


Figure 7 : Structuration de la couche vue Android

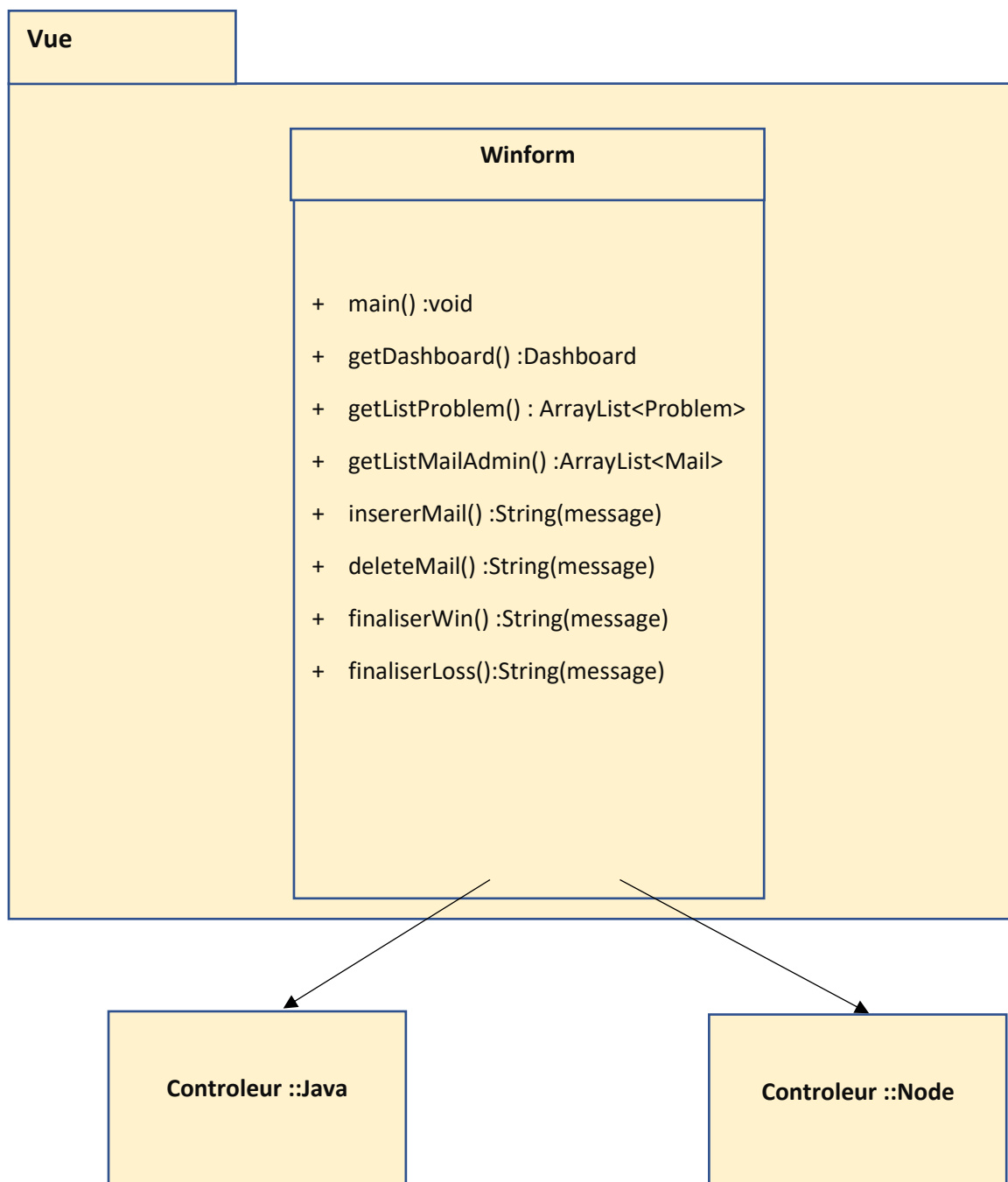


Figure 8 : Structuration de la couche vue Winform

## Structuration de la couche contrôleur

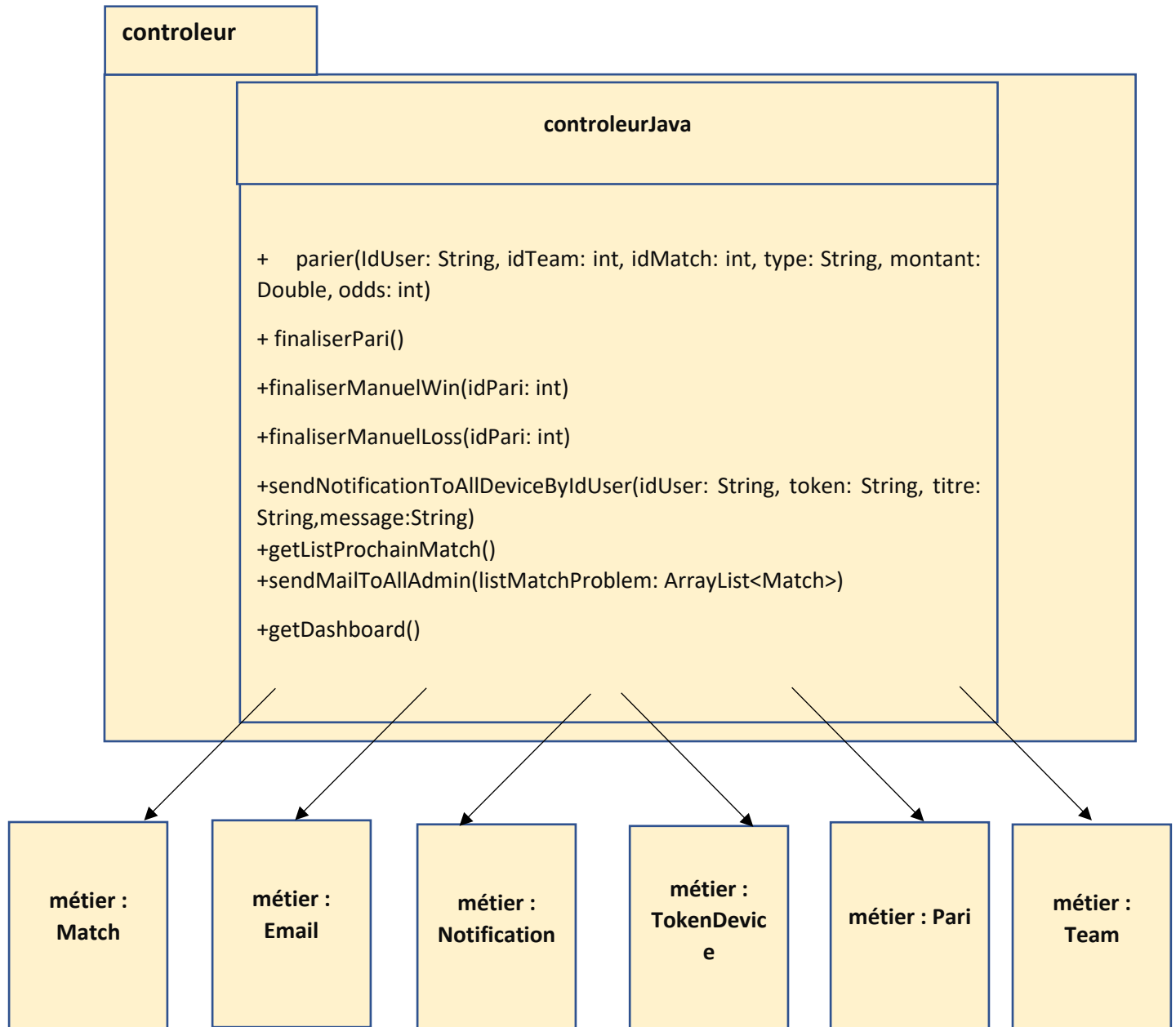


Figure 9 : Structuration de la couche contrôleur Java

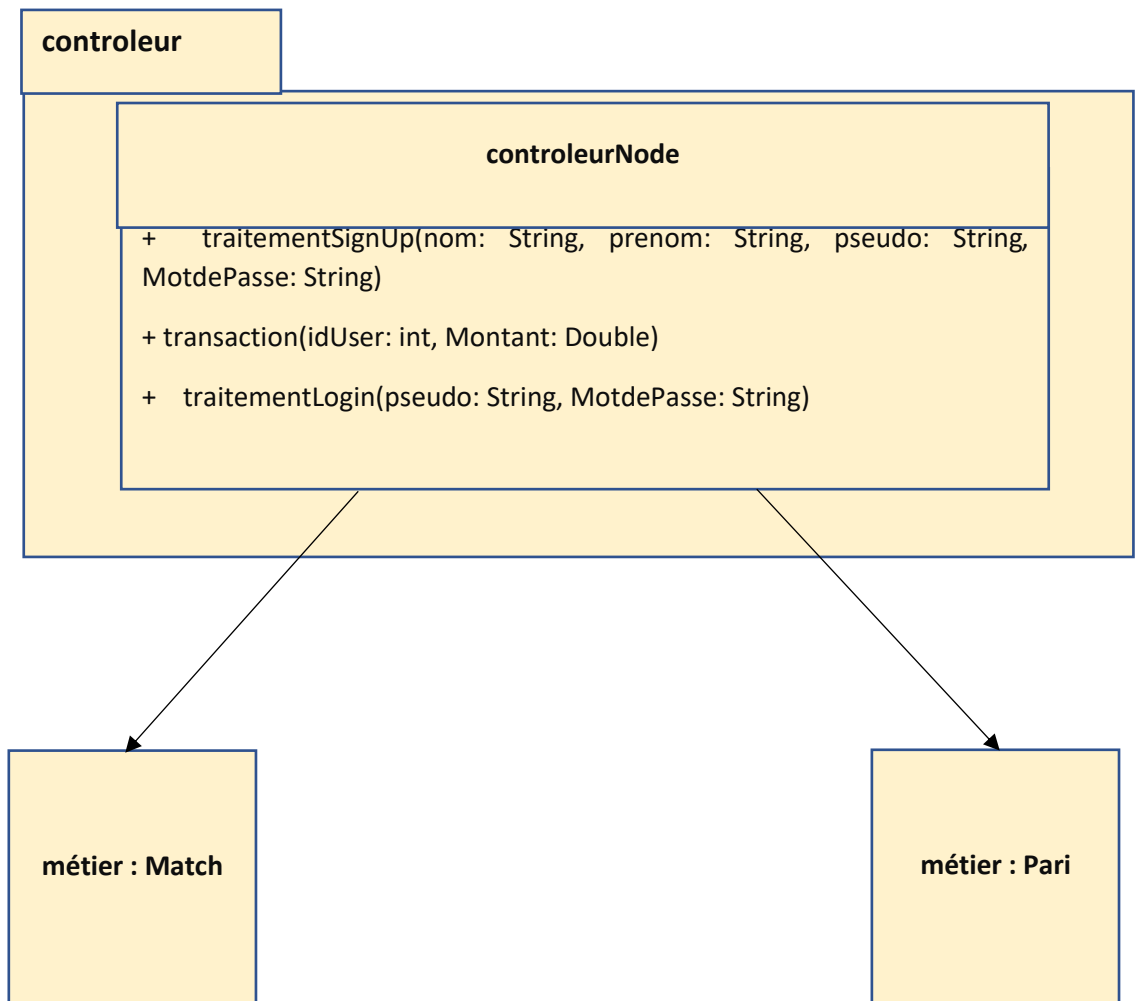


Figure 10 : Structuration de la couche contrôleur Node



## Structuration de la couche métier



Figure 11 : Structuration de la couche métier

### 8.2.3. Modélisation de données

Modèle Conceptuel de données

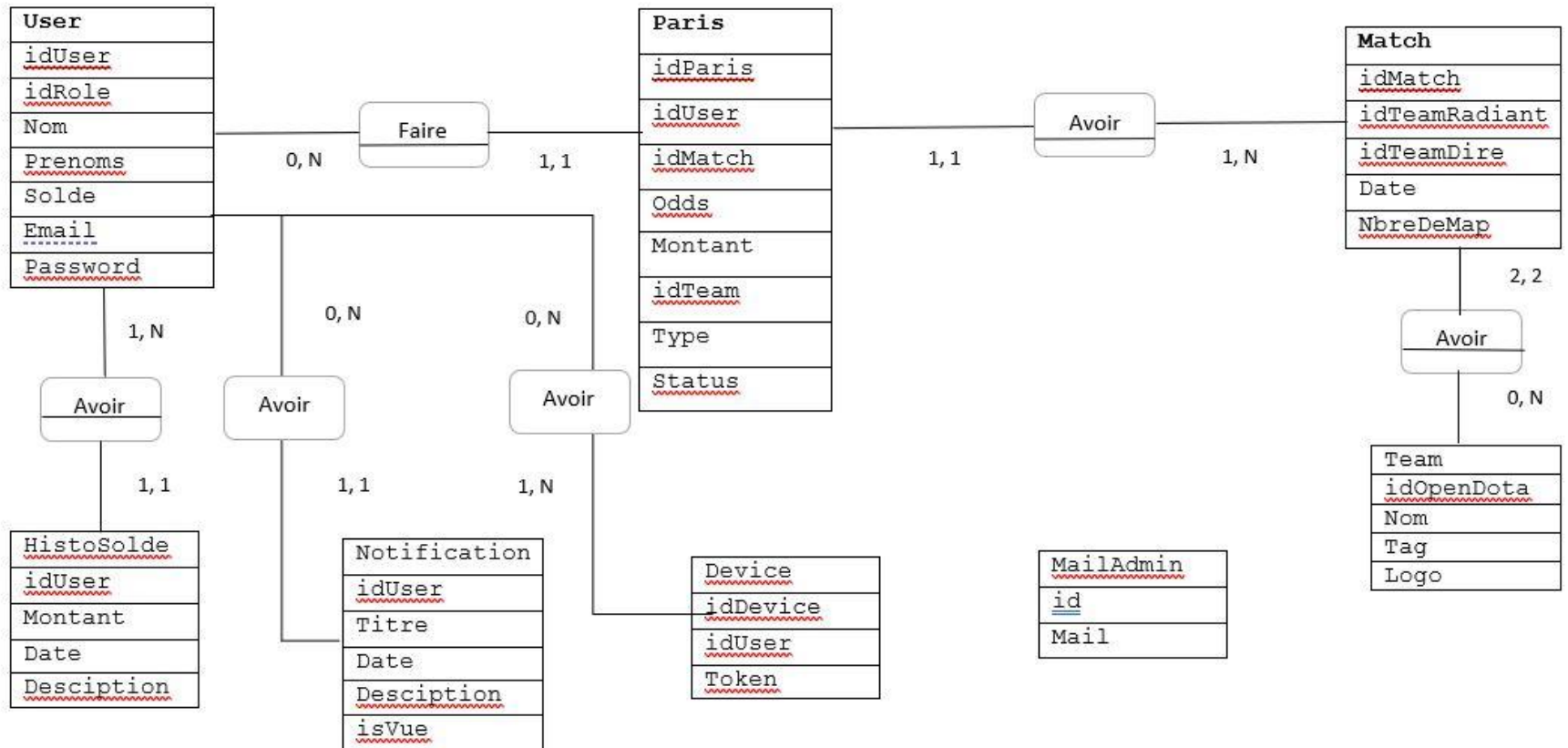


Figure 12 : Modèle Conceptuel de données

## 8.2.4. Réalisation des cas d'utilisations

### 8.2.4.1. Cas d'utilisation pour parier

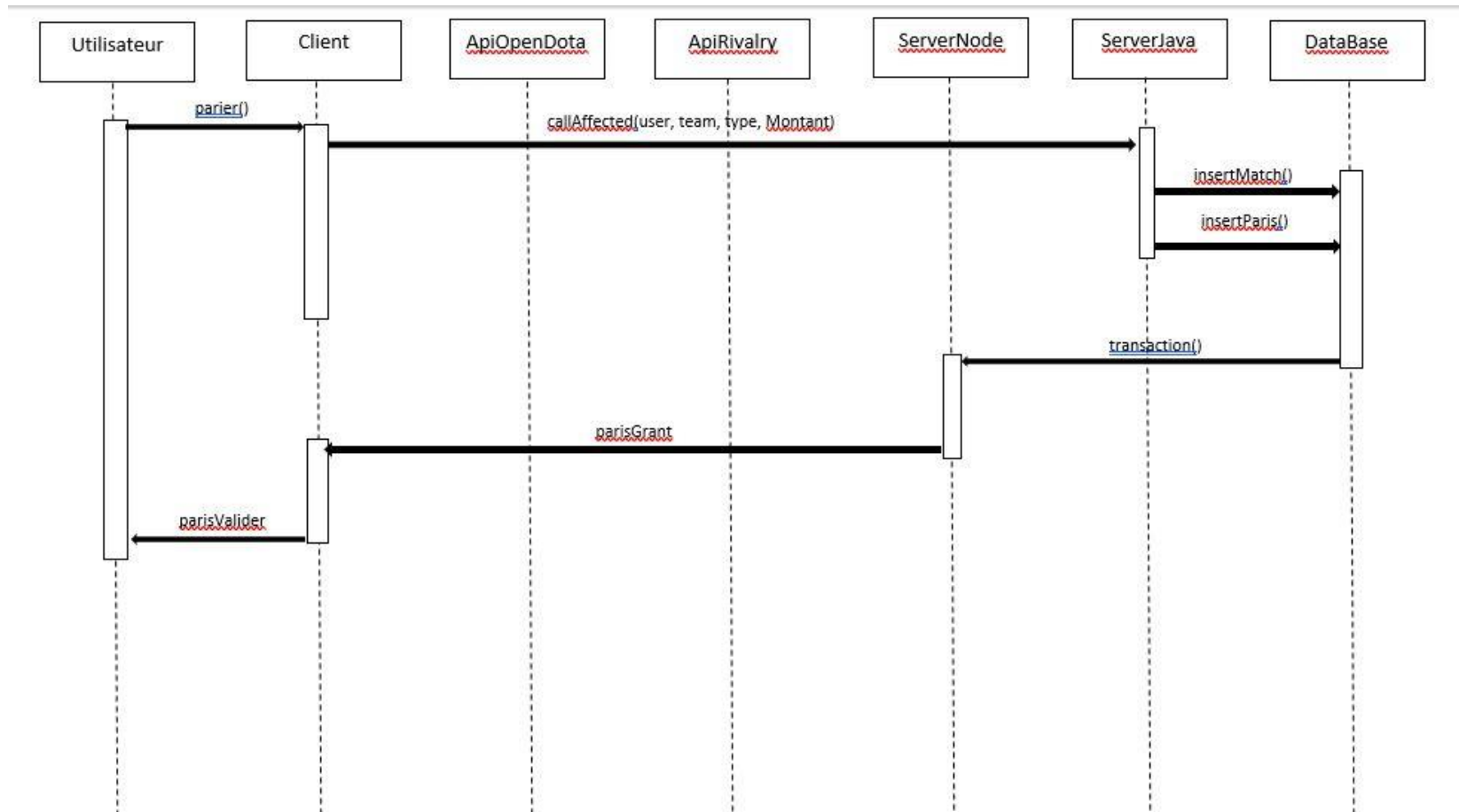


Figure 13 : Cas d'utilisation pour parier

## 8.2.4.2. Cas d'utilisation pour finaliser un pari

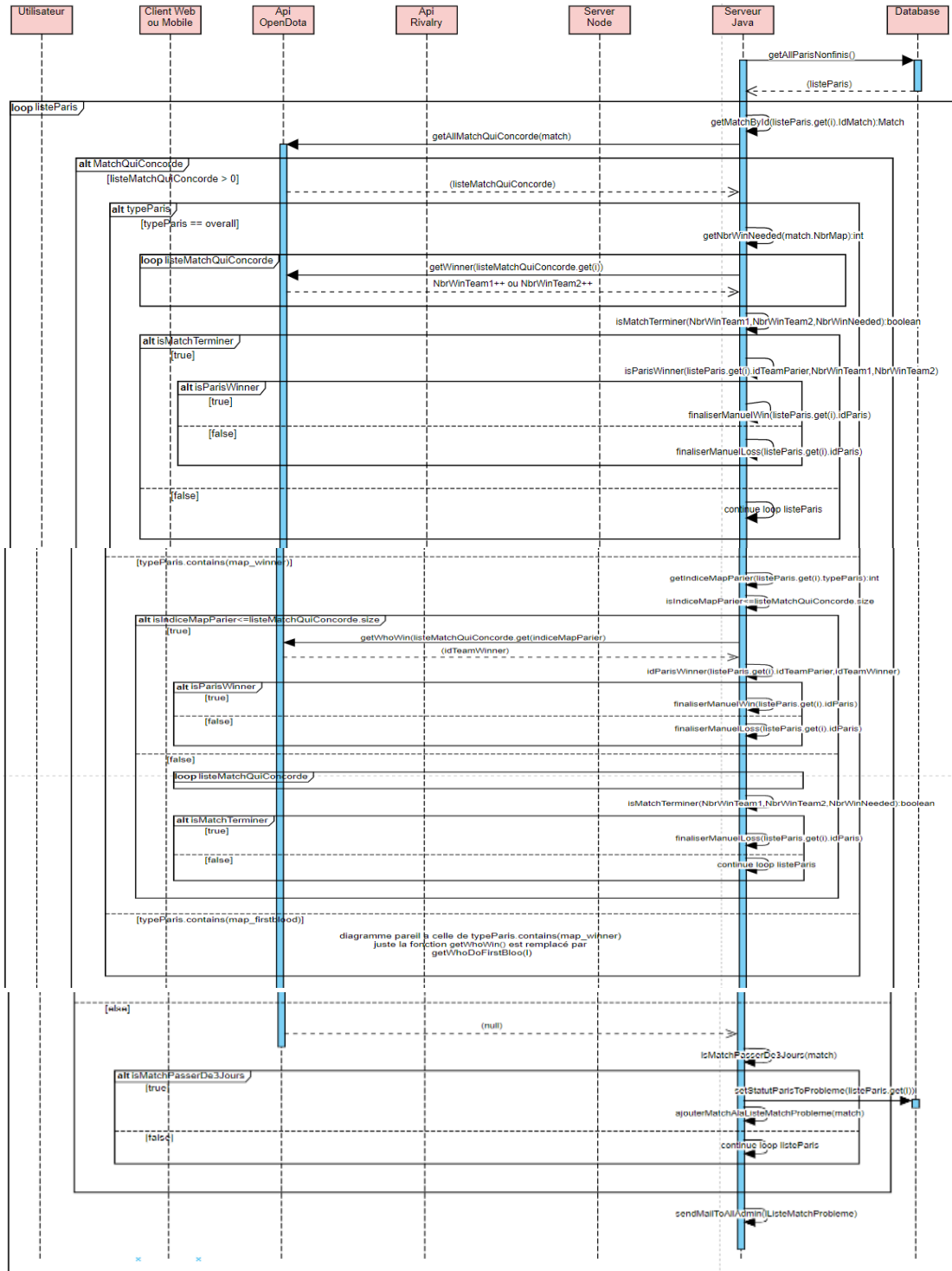


Figure 14 : Cas d'utilisation pour finaliser un pari

### 8.2.5. Les composants et leur déploiement

Les composants logiciel sont subdivisés en trois parties selon les couches :

Le site web en Angular nommé **frontend-angular-mbds272957**

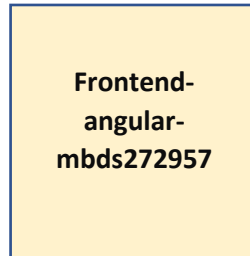


Figure 15 : Composant web

Il est déployé sur un service d'hébergement Heroku

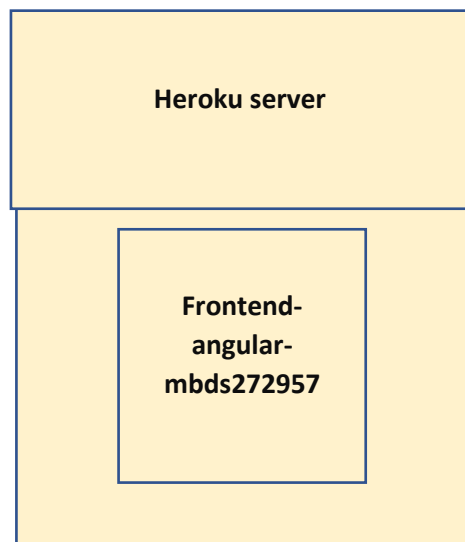


Figure 16 : Déploiement web

L'application Android est produite sous la forme d'un seul package pour Android mobile nommé : **Dota2bet.apk**

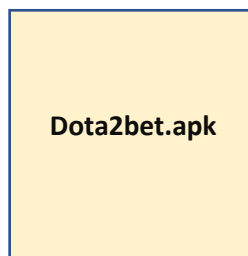
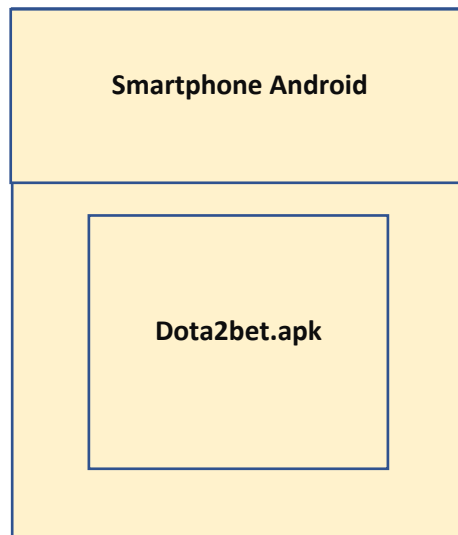


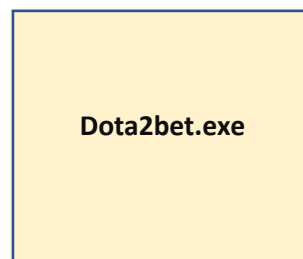
Figure 17 : Composant Android

L'application Android peut être installée dans les smartphones Android ayant Lollipop, Android 5.0 minimum



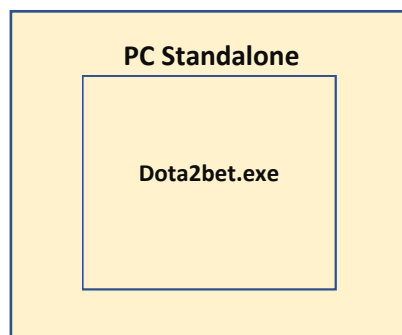
*Figure 18 : Déploiement Android*

Le client lourd Winform est produit sous la forme d'un seul composant exécutable nommé : **Dota2bet.exe**



*Figure 19 : Composant Winform*

Le client winform peut être déployé sur un PC Standalone :



*Figure 20 : Déploiement Winform*

Le nommage des composants est en rapport avec le nom de la plateforme et les numéros de l'équipe du projet.

## 9. Tests du système logiciel

Sur les tests du système logiciel, on va définir un champ de test pour chaque partie de la plateforme.

Test sur le site web :

- Tout d'abord le client lance son navigateur web
- Il va saisir l'adresse url du site web sur la barre de recherche **frontend-angular-mbds272957.herokuapp.com**
- Une fois sur la page d'accueil du site, le client va cliquer sur LOGIN pour se connecter à son compte
- Il va remplir les champs Pseudo et Mot de passe par Benett et ProBetBeluga puis appuyer sur le bouton CONNEXION
- Une fois connecter, il appuie sur Parier
- Sur la page parier, il va choisir un prochain match Fnatic vs SMG pour parier.
- Il va choisir SMG avec un type de pari First Team to Get First Blood
- Il va remplir le montant à parier 3000 Ar pour SMG, puis appuyer sur Parier
- Il va ensuite recevoir une notification, ' Votre pari a été bien enregistré '
- Après la fin du match, il a reçu les résultats de son pari et voir les changements dans son solde utilisateur.

## 10. Conclusion générale

Réaliser cette plateforme de pari en ligne a été énormément riche, et existant sur la mise en place d'un service complet.

Nous avons réalisé 5 blocs de programmes en total : une application en Angular, une application Android, un client lourd en winform. Deux serveurs d'application, le premier en Java et le deuxième en NodeJs. Et deux CloudBD distinctes ; Oracle cloud et MongoDB

La plateforme est livrée à 100% en se référant sur le cahier des charges qui nous a été fourni.

L'équipe a notamment rencontré les premières difficultés pour la demande de deux serveurs d'application distinctes et 2 serveurs de base de données ; mais avec des longues sessions de réunion, d'analyses et une bonne pratique de

conception. L'équipe a pu résoudre le problème.

Oracle cloud, a posé un souci sur la location de serveur et la création de compte développeur. Sur ce l'équipe a dû se mobiliser pour payer et louer le serveur pendant 30 jours.

L'implémentation du technologie QR Code nous a enthousiasmé par le fait d'explorer et exploiter cette technologie.

Les perspectives du projet selon l'équipe se voit d'abord dans l'optimisation à rendre plus fluide la montée en charge du serveur actuel. Nous prévoyons d'ajouter d'autres titres d'e-sport dans le catalogue de pari, des lives et des actualités autour de l'esport. Une sortie d'une version iOS de l'application est prévue pour 2022.

Ce projet nous a fait ressortir nos capacités techniques et humaines dans la gestion de projet, notamment un beau travail d'équipe, que malgré nos difficultés sur un obstacle notre devise reste toujours **" Sambatra izay mino" ... " Fa minoa fotsiny ihany"** qui nous a fortifié durant ces périodes.