CHAPITRE 2:sprint 0

1. Introduction :

Dans ce chapitre nous allons présenter le sprint zéro qui est définie comme étant le premier pas de réalisation de projet. En premièrement, nous allons identifier les acteurs, les besoins fonctionnels et les besoins non fonctionnels .en deuxièmement, ce sprint va contenir une description de la démarche de la méthode adoptée ainsi que les différents environnements de travail. Finalement, il résumera l’architecture générale du projet.

1. Spécification des besoins :

La spécification des besoins est la première phase de cycle de vie d’une application.

Par suite les besoins de notre application se repartissent comme suit :

1. Besoins fonctionnels :

Dans cette partie nous allons présenter les différents détails des fonctionnalités que notre application doit fournir aux acteurs qui sont les suivants :

* Créer un nouveau compte : la création de compte est une fonctionnalité dans lequel l’utilisateur peut se connecter rapidement.
* S’authentifier :cette tâche est obligatoire pour que l’utilisateur puisse accéder à l’application.
* Gérer le programme :le droit de l’ajout de programme détaillé de l’évènement en précisant le lieu et aussi l’accès total pour n’importe quelle modification, mise à jour et suppression.
* Gérer document : cette fonctionnalité permet à l’administrateur d’ajouter, consulter et supprimer le document.
* Gérer jury : dans ce cas l’administrateur pour ajouter, supprimer, modifier les jurys.
* Gérer speakers:dans ce cas l’administrateur pour ajouter, supprimer, modifier les speakers.
* Gérer lauréat : dans ce cas l’administrateur pour ajouter, supprimer, les lauréats.
* Gérer candidat : dans ce cas d’utilisation l’administrateur à le droit de supprimer et consulter la liste des candidats.
* Gérer exposant : dans ce cas d’utilisation l’administrateur à le droit de supprimer et consulter la liste des exposants.
* Gérer participant : dans ce cas d’utilisation l’administrateur à le droit de supprimer et consulter la listedes participants.
* Consulter programme :l’utilisateur peut consulter le programme de l’évènement et son localisation.
* Consulter jury : l’utilisateur de l’application peut consulter les jurys.
* Consulter speakers: l’utilisateur de l’application peut consulter les speakers.
* Consulter lauréat: l’utilisateur de l’application peut consulter les lauréats.
* Réserver espace exposant: les exposants ont le pouvoir de réserver et d’annuler la réservation d’une espace d’exposition.
* Remplir une enquête de satisfaction : les participants, les exposants, les candidats ont l’accès de remplir une enquête de satisfaction pour évaluer chaque meeting.
* Remplir un formulaire : chaque participant, exposant, candidat doit remplir un formulaire afin de pouvoir participer à l’évènement.
* Chatter : les candidats peuvent communiquer entre eux pour créer une bonne interactivité.
* Recevoir attestation : les participants à la fin de l’évènement auront des attestations de présence téléchargeable.

1. Besoin non fonctionnels :

Les besoins non fonctionnels sont liés aux contraintes qui peuvent être dans l’application pour les prendre en considération et mettre en place une solution bien adéquate aux attentes précitées. Notre application doit obligatoirement respecter les fonctionnalités suivantes :

* L’Extensibilité : l’application doit être extensible, c’est à dire qu’il pourra y avoir une possibilité d’ajouter, ou de modifier des nouvelles fonctionnalités.
* Authentification :l’application doit être hautement sécurisée car les informations ne doivent pas être accessibles à tous le monde.
* Les contrôles des champs :l’application doit avoir un contrôle sur les champs de saisis pour éviter l’introduction des informations qui ne correspondent pas aux types des champs.
* Convivialité : l’application doit offrir une interface conviviale, explicite, simple à utiliser.

1. Identification des acteurs :

* *Administrateur :*le rôle de l’administrateur est la gestion de toutes les fonctionnalités du système.
* *Exposant :*il réserve une espace d’exposition.
* *Participant :*il participe à l’évènement il peut assister aux conférences et aux workshops.
* *Candidat :*il participe à l’évènement entant que candidat afin de gagner un parmi les trophées proposé.
* *L’utilisateur :*personne qui a un espace privé qui a lui permet de s’informer généralement sur les détailles de l’évènement.

1. Pilotage de projet avec Scrum :
2. Les rôles Scrum :

L’équipe Scrum qui a participé à la réalisation de ce projet se définit comme suit :

* **Product Owner :** Mme Tritar Lamia c’est la directrice de l’agence et l’organisatrice de l’événement .Elle détermine ce que doit être réalisé.
* **Scrum Master :** MmeAyachi Ghannouchi Sonia c’est celle qui veille sur l’avancement du développement du projet. Elle organise les réunions et explique la méthode agile.
* **Scrum Team :** équipe formée par Moussa Sahar et Laajili Hela responsable de la définition des sprints ainsi que leur réalisation.

1. Product Backlog :

|  |  |  |  |
| --- | --- | --- | --- |
| *ID* | *User’s story* | *Priorité* | *Complexité* |
| *1* | *-En tant qu’administrateur, je veux s’authentifier à l’aide d’un login et d’un mot de passe afin de gérer le fonctionnement de l’application et protéger mon système.* | *1* | *Elevée* |
| *2* | *-En tant qu’administrateur, je veux gérer les membres des jurys.* | *2* | *Moyenne* |
| *3* | *-En tant qu’administrateur, je veux gérer les speakers.* | *2* | *Moyenne* |
| *4* | *-En tant qu’administrateur, je veux gérer le programme.* | *3* | *Elevée* |
| *5* | *-En tant qu’administrateur, je veux ajouter les documents dont ils ont besoin pendant l’évènement.* | *3* | *Elevée* |
| *6* | *-En tant qu’administrateur, je veux ajouter les lauréats des éditions précédentes.* | *2* | *Elevée* |
| *7* | *-En tant qu’administrateur, je veux gérer les participants.* | *1* | *Moyenne* |
| *8* | *-En tant qu’administrateur, je veux gérer les candidats.* | *1* | *Moyenne* |
| *9* | *-En tant qu’administrateur, je veux gérer les exposants.* | *1* | *Moyenne* |
| *10* | *-En tant que visiteur, je veux créer un compte.* | *1* | *Elevée* |
| *11* | *-En tant qu’utilisateur, je veux m’authentifier afin d’avoir plus d’informations.* | *1* | *Moyenne* |
| *12* | *-En tant qu’utilisateur, je veux voir les détails de l’évènement.* | *3* | *Faible* |
| *13* | *-En tant qu’utilisateur, je veux consulter la liste des lauréats,*  *jury et speakers.* | *2* | *Faible* |
| *14* | *-En tant qu’exposant, je veux réserver un stand pour l’exposition.* | *1* | *Moyenne* |
| *15* | *-En tant que candidat, je veux m’inscrire à une ou plusieurs compétitions selon les catégories que je le choisie.* | *1* | *Moyenne* |
| *16* | *-En tant qu’utilisateur, je veux chater avec les autres candidats.* | *3* | *Moyenne* |
| *18* | *-En tant que participant, je veux m’inscrire à l’évènement et sélectionner les conférences, et workshop pour y assister.* | *1* | *Moyenne* |
| *19* | *-En tant que participant, je veux recevoir mon attestation de présence en ligne.* | *3* | *Faible* |
| *20* | *-En tant que (participant, candidat, exposant), je veux recevoir un code QR après le payement.* | *3* | *Elevée* |
| *21* | *-En tant que (participant, candidat, exposant), je veux remplir une enquête de satisfaction après l’évènement afin de noter l’organisation et le bon déroulement de l’évènement.* | *3* | *Moyenne* |
| *22* | *-En tant que visiteur, je veux créer un compte.* | *2* | *Moyenne* |
| *23* | *-En tant qu’administrateur, je veux envoyer les attestations de présence par e-mail.* | *3* | *Moyenne* |
| *24* | *-En tant qu’administrateur, je veux envoyer un code QR après le payement.* | *3* | *Moyenne* |

1. Planification des sprints :
2. Environnement de travail :
3. Environnement matériel :

Pour la réalisation de projet, nous avons utilisé un pc portable avec les caractéristiques suivantes :

1. Environnement de développement :

\*Angular8 :



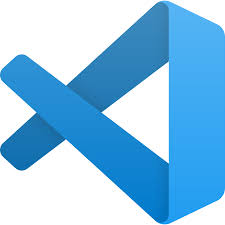
Angular est un **Framework** open source écrit en **JavaScript** qui permet la création d’applications Web accessibles via une page web unique.Angular 8 est sorti le 28 mai 2019

\*ionic3 :



Ionic est un framework open-source composé d’un mélange d’outils et de technos pour développer des applications mobiles hybrides rapidement et facilement ; créé en 2013 par Max Lynch, Ben Sperry, et Adam Bradley.

\*visual studio code :



**Visual Studio Code** est un [éditeur de code](https://fr.wikipedia.org/wiki/%C3%89diteur_de_texte) extensible développé par [Microsoft](https://fr.wikipedia.org/wiki/Microsoft) pour [Windows](https://fr.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://fr.wikipedia.org/wiki/Linux) et [macOS](https://fr.wikipedia.org/wiki/MacOS)[1](https://fr.wikipedia.org/wiki/Visual_Studio_Code#cite_note-TechCrunch-1).

\*PhpMyAdmin :



phpMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL.

\*MySQL :



MySQL est un système de gestion de bases de données relationnelles. Il est distribué sous une double licence GPL et propriétaire.

\*CSS :



Les **CSS**, Cascading Style Sheets (feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML

\*HTML :



**HTML5** (*HyperText Markup Language 5*) est la dernière révision majeure du HTML ([format de données](https://fr.wikipedia.org/wiki/Format_de_donn%C3%A9es) conçu pour représenter les [pages web](https://fr.wikipedia.org/wiki/Pages_web)). Cette version a été finalisée le 28octobre 2014.

1. Environnement Logiciel :

**\*Conception** :

**\***draw.io :



Draw.io est une application gratuite en ligne, accessible via son navigateur qui permet de dessiner des diagrammes.

**\***StarUML :



StarUML est un logiciel de modélisation UML, cédé comme open source par son éditeur

**\*gestion de projet** :

\*GitHub :



GitHub est un outil gratuit pour héberger du code open source, et propose également des plans payants pour les projets de code privés.

**\***Postman :



Une application moderne est construite sur des APIs. C'est sur cette phrase d'accroche que nous découvrons le site de [**Postman**](https://www.getpostman.com/), un outil qui permet de construire et de tester rapidement des requêtes HTTP.

1. Architecture généralede l’application :
2. Architecture physique

L'architecture à trois niveaux s’avère la mieux à adopter (aussi appelée architecture 3- tiers) caractérise les systèmes clients/serveurs dans lesquels le client demande une ressource et le serveur la lui fournit directement. En effet, l’architecture 3-tiers est l'extension du modèle client –serveur. L'architecture logique du système est divisée en trois niveaux ou couches : \*couche présentation

\*couche métier

\*couche accès aux données

Comme l’indique la figure ci-dessous, le client (couche présentation) accèdent au serveur pour demander une ressource. Ce dernier (couche métier) fait appel au serveur secondaire (serveur de base de données) pour livrer les données demandé aux clients.

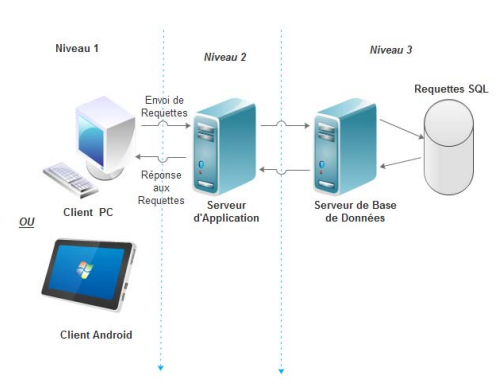


Figure :

1. Architecture logiciel :

Le MVC (« modèle vue contrôleur ») est une architecture de développement visant à séparer le code source en modules.

Cette architecture est le plus communément retrouvée au sein d’applications web mais existe également au niveau des applications lourdes.

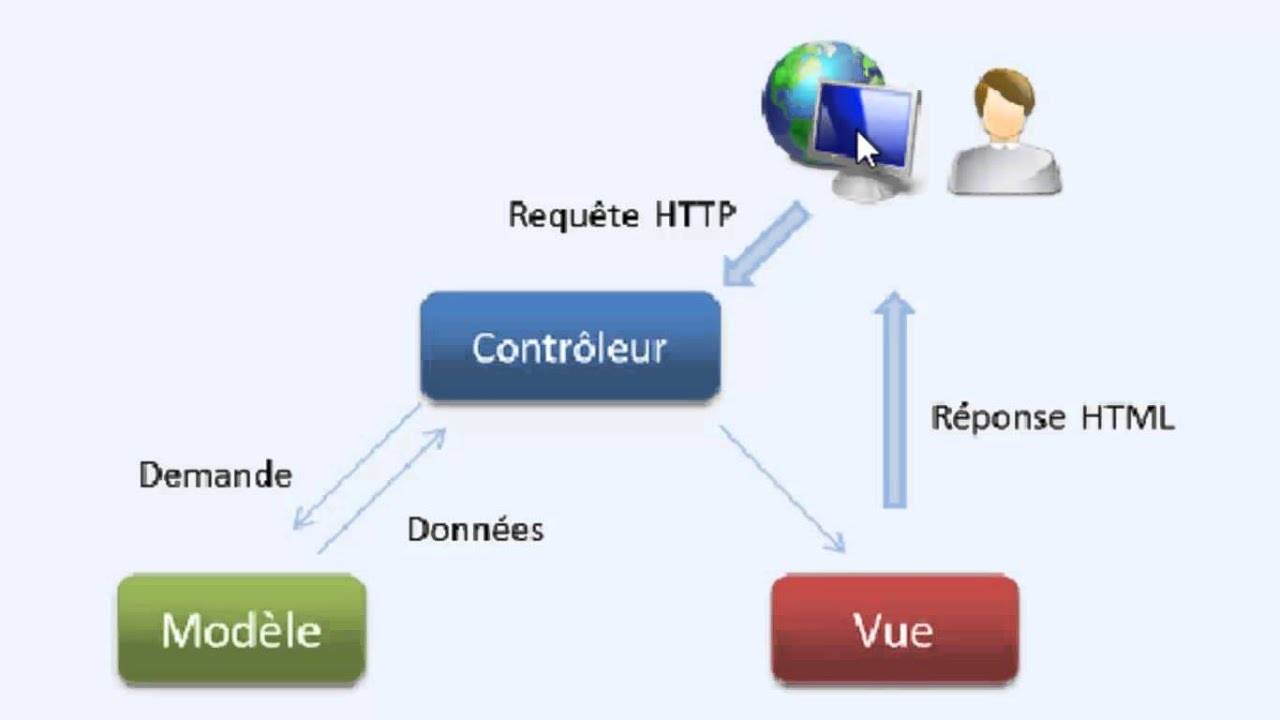
Ainsi, comme vous pouvez le voir, il y a 3 couches distinctes :

\* **Le modèle** définit les données utilisées par l’application. En effet, c’est ici que le lien se fera entre notre application et la base de données.

\* **La vue** définit la façon dont les informations seront affichées à l’écran (via des composants par exemple). Il s’agit de l’interface utilisateur.

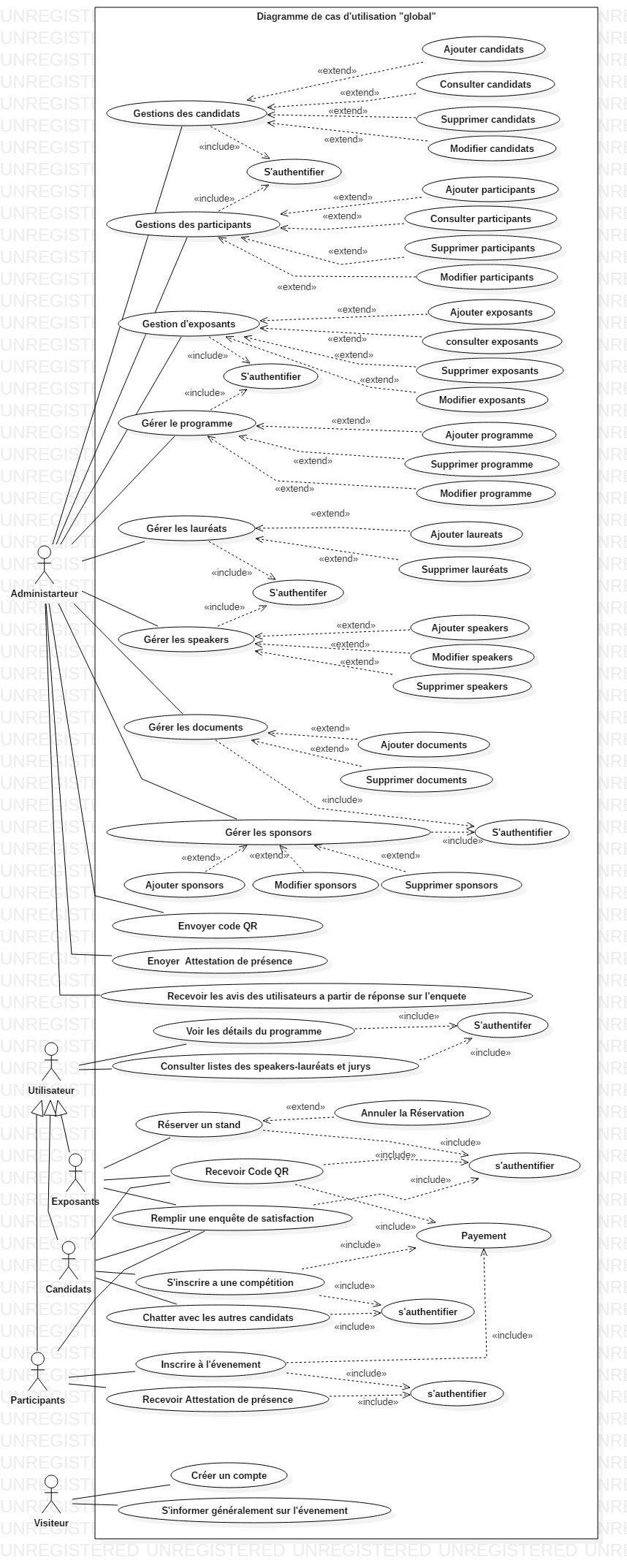
\* **le contrôleur**, nous retrouvons toute la logique métier. En effet, lorsque l’utilisateur interagit avec la vue, la requête est traitée par le contrôleur.

Ainsi, c’est le contrôleur qui définira la logique d’affichage, et affichera la vue suivante à l’écran.



1. Conception :
2. Diagramme de cas d’utilisation global :

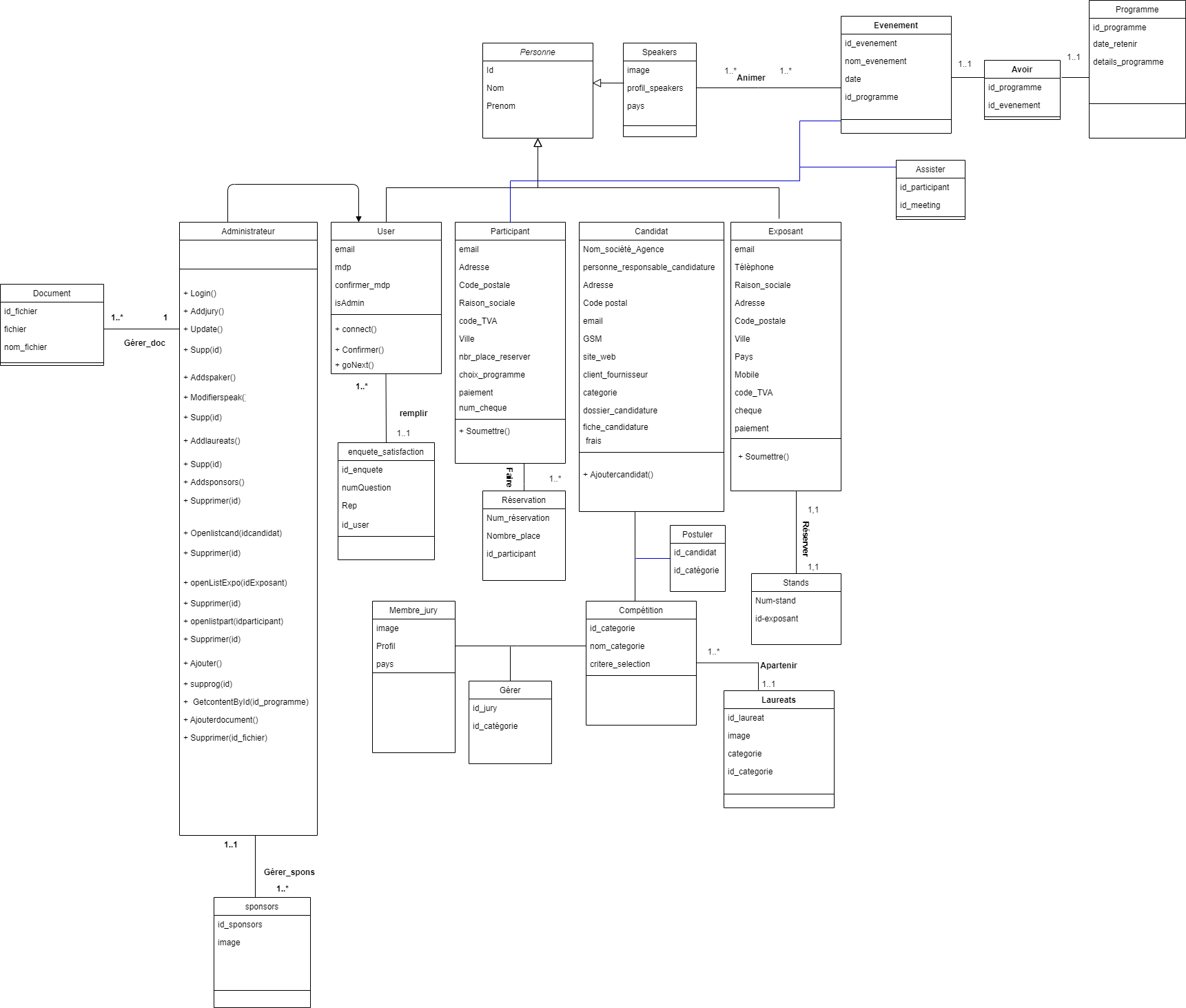
Les diagrammes de CU sont la représentation graphique des [interactions](https://fr.wikipedia.org/wiki/Unified_Modeling_Language) entre les [acteurs](https://fr.wikipedia.org/wiki/Acteur_(UML)) et le système selon un ordre chronologique dans la formulation [Unified Modeling Language](https://fr.wikipedia.org/wiki/Unified_Modeling_Language).



1. Diagramme de classe global :

Le diagramme de classes est un schéma utilisé en [génie logiciel](https://fr.wikipedia.org/wiki/G%C3%A9nie_logiciel) pour présenter les [classes](https://fr.wikipedia.org/wiki/Classe_(informatique)) et les [interfaces](https://fr.wikipedia.org/wiki/Interface_(informatique)) des systèmes ainsi que les différentes relations entre celles-ci.

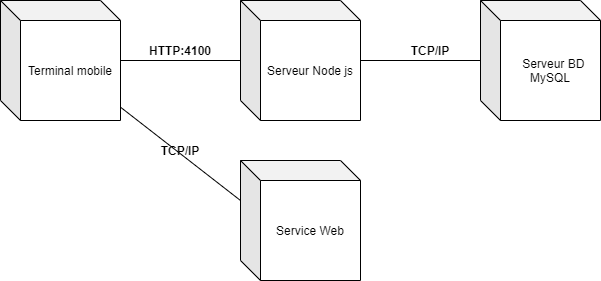
Une [classe](https://fr.wikipedia.org/wiki/Classe_(informatique)) est un ensemble de [fonctions](https://fr.wikipedia.org/wiki/Fonction_(informatique)) et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la [programmation orientée objet](https://fr.wikipedia.org/wiki/Programmation_objet). Elles permettent de modéliser un [programme](https://fr.wikipedia.org/wiki/Programme_informatique) et ainsi de découper une tâche complexe en plusieurs petits travaux simples.



1. Diagramme de déploiement :

Le diagramme de déploiement définit l’architecture matérielle de l’application. Il présente les périphériques utilisés et la répartition du système sur ces différents éléments.

Le diagramme de déploiement de notre application est représenté par le diagramme ci-après :



1. Conclusion :

Après avoir décrit les besoins fonctionnels et techniques attendus de notre application qui consistent à mettre en place une démarche de développement. Lors de cette dernière phase nous avons essayé d’exprimer le fonctionnement de notre système en se basant principalement sur le diagramme de cas d’utilisation global. Aussi après avoir présenté les différents acteurs, user story et l’environnement de travail nous pouvons ainsi entamer la prochaine étape qui consiste à lancer le sprint 1.