****

****

****

**Conception et Développement d’une solution**

**Web pour la gestion des clubs pour l'ISIMM**

****

**Réalisé par :**

**Maryem Aloui Ines Chebbi**

**Haroun Hassen Aziz Ferchichi**

**Louay Ben Ali Aziz Ben Tanfous**

**Encadré par Mme Aljia Bouzidi**

**Année Universitaire : 2023-2024**

**Remerciements**

Nous tenons à exprimer notre profonde gratitude à tous les membres de notre équipe pour leur dévouement, leur travail acharné et leur engagement tout au long de cette année. Votre contribution a été inestimable et a grandement contribué au succès de notre projet. Nous sommes fiers de nous-mêmes et de notre travail.

Nous voudrions également remercier chaleureusement notre encadrant et professeur, Dr Aljia Bouzidi, pour son soutien, ses conseils précieux et son expertise. Son engagement envers notre projet et ses efforts pour nous guider ont été inestimables.

TABLE DE MATIÈRES

Introduction générale7

Chapitre Ⅰ : Contexte et objectifs du projet8

Introduction9

1. Contexte général9

1.1. Cadre général 9

1.2. Présentation de l’organisme de l’accueil9  
 1.3. Objectifs 9

2. Etude de l’existant10

2.1. Solutions similaires10

2.2. Critique de l’existant13  
 2.3. Solution retenue13

3. Méthodologie de travail adoptée Scrum ………14

Conclusion15

Chapitre II : Sprint 0 : phase de préparation16

Introduction17

1. Analyse et spécification des besoins17

1.1. Identification des acteurs……………………………………………………………….

1.2. Capture des besoins…………………………………………………………………

1.2.1. Besoins fonctionnels ……………………………………………………………………………………………

1.2.2. Besoins non fonctionnels……………………………………………………………………………………

1.3. Spécification des besoins fonctionnels………………………………………………

2. Etude technique17 2.1. Environnement du travail17

2.1.1. Environnement de développement Front-End……………………………………………….

2.1.1. Environnement de développement Back-End…………………………………………………

3. Pilotage avec Scrum21

3.1. Définition des rôles21

3.2. Backlog du produit23

3.3. Planification des sprints23

Conclusion…………………………………………………………………………….

Chapitre III : Sprint 1 & Sprint 2 36

Introduction…………………………………………………………………………….

1. Itération du sprint 137

1.1. Spécification des besoins du sprint 137

1.1.1. Backlog du produit………………………………………………………………………………………………

1.1.2. Diagramme du cas d’utilisation du sprint 1………………………………………………………….

1.2. Analyse du sprint 138  
 1.2.1. Analyse de la réalisation du sprint 139

1.2.2. Diagramme de classe du sprint 1

1.3. Conception du sprint 1

1.3.1. Vue statique du sprint 1

Diagramme de classe………………………………………………………………………………………………………

1.3.2. Vue dynamique du sprint1…………………………………………………………………………………………

Diagramme de séquence…………………………………………………………………………………………………

2. Itération du sprint 237

2.1. Spécification des besoins du sprint 237

2 .1.1. Backlog du produit ………………………………………………………………………………………….

2.1.2. Diagramme du cas d’utilisation du sprint 2 …………………………………………………………

2.2. Analyse du sprint 238  
 2.2.1. Analyse de la réalisation du sprint 239

2.2.2. Diagramme de classe du sprint 2

2.3. Conception du sprint 2

2.3.1. Vue statique du sprint 2

Diagramme de classe …………………………………………………………………………………………………….

2.3.2. Vue dynamique du sprint2 ………………………………………………………………………………………

Diagramme de séquence……………………………………………………………………………………………….

Conclusion……………………………………………………………………………….

Chapitre IV : Sprint 3 & Sprint 436

Introduction…………………………………………………………………

1. Itération du sprint 337

1.1. Spécification des besoins du sprint 337

1.1.1. Backlog du produit………………………………………………………………………………………………

1.1.2. Diagramme du cas d’utilisation du sprint 3………………………………………………………….

1.2. Analyse du sprint 338  
 1.2.1. Analyse de la réalisation du sprint 339

1.2.2. Diagramme de classe du sprint 3

1.3. Conception du sprint 3

1.3.1. Vue statique du sprint 3

Diagramme de classe…………………………………………………………………………………………………….

1.3.2. Vue dynamique du sprint 3…………………………………………………………………………………….

Diagramme de séquence………………………………………………………………………………………………

2. Itération du sprint 437

2.1. Spécification des besoins du sprint 437

2 .1.1. Backlog du produit…………………………………………………………………………………………….

2.1.2. Diagramme du cas d’utilisation du sprint 4……………………………………………………….

2.2. Analyse du sprint 438

2.2.1 Analyse de la réalisation du sprint 439

2.2.2. Diagramme de classe du sprint 4

2.3. Conception du sprint 4

2.3.1. Vue statique du sprint 4

Diagramme de classe……………………………………………………………………………………………………

2.3.2. Vue dynamique du sprint4…………………………………………………………………………………….

Diagramme de séquence………………………………………………………………………………………………

Conclusion……………………………………………………………………………….

Conclusion générale…………………………………………………………………………….

LISTE DES FIGURES

**Figure1 : Logo de l’ISIMM9**

**Figure2 : Capture de plateforme « ClubExpress » 10**

**Figure3 : Capture de plateforme « WildApricot »11**

**Figure4 : Capture de plateforme « MembreClicks »12**

**Figure5 : Modèle Scrum15**

**Figure6 : Frontend & Backend 18**

**Figure7 : Popularité des Framework JavaScript au fil du temps22**

**Figure8 : Logo de React 23**

**Figure9 : Logo de MongoDB 24**

**Figure10 : Logo de nodeJS 25**

**Figure11 : Logo de ExpressJS 26**

**Figure12 : Diagramme de cas d’utilisation du sprint 1…...………………… 28**

**Figure13 : Diagramme de classe d’analyse du sprint 130**

**Figure14 : Diagramme de classe de sprint 132**

**Figure 15 : Diagramme de séquence du sprint 135**

**Figure16 : Diagramme de cas d’utilisation du sprint 241**

**Figure17 : Diagramme de classe d’analyse du sprint 245**

**Figure18 : Diagramme de classe de sprint 246**

**Figure19 : Diagramme de séquence du sprint 247**

**Figure20 : Diagramme de cas d’utilisation du sprint 348**

**Figure21 : Diagramme de classe d’analyse du sprint 349**

**Figure22 : Diagramme de classe de sprint 350**

**Figure23 : Diagramme de séquence du sprint 347**

**Figure24 : Diagramme de cas d’utilisation du sprint 448**

**Figure25 : Diagramme de classe d’analyse du sprint 449**

**Figure26 : Diagramme de classe de sprint 450**

**Figure27 : Diagramme de séquence du sprint 449**

**Introduction générale**

Dans un monde en constante évolution, la gestion des clubs est devenue un élément crucial pour le développement des instituts d'enseignement supérieur tel que l'ISIMM. Cependant, les outils traditionnels de gestion des clubs peuvent être inefficaces et peu pratiques. Pour répondre à ce défi, il a été proposé de mettre en place une solution web pour la gestion des clubs de l'ISIMM.

Ce rapport détaillera les différentes phases par lesquelles nous sommes passés pour concevoir et développer cette solution. Dans un premier temps, nous présenterons le cadre général du projet, en introduisant l'ISIMM et les notions liées au projet demandé. Dans le deuxième chapitre, nous ferons un état de l'art des solutions existantes sur le marché pour la gestion des clubs. Enfin, dans le troisième chapitre, nous proposerons une solution adaptée aux besoins spécifiques de l'ISIMM.

**Chapitre Ⅰ : Contexte et objectifs du projet**

**Introduction :**

Ce rapport donne le coup d’envoi à notre projet de plateforme de gestion de clubs. Son objectif est clair : simplifier la vie des organisateurs et membres de clubs. En définissant clairement nos besoins et attentes, nous jetons les bases d’une solution numérique qui facilitera la gestion quotidienne des clubs, favorise l’interaction entre les membres et rendra l’expérience globale plus agréable.

**1. Contexte générale :**

**1.1. Cadre général :**

Ce projet de gestion de clubs de l’ISIMM, intégré dans le cadre des projets fédérés, vise à concevoir et mettre en œuvre une plateforme complète répondant aux besoins administratifs et de communication des clubs au sein de l’université.

**1.2. Présentation de l’organisme de l’accueil :**

Notre organisme d’accueil est l’institut supérieur d’informatique et de mathématiques de Monastir. Qui est fondée en 2002.L’ISIMM offre des programmes académiques diversifiés tels que des licences des mastères et des doctorats. L’ISIMM joue un rôle clé dans le développement des compétences techniques et scientifiques avec un engagement envers l’excellence académique et la recherche.



**Figure 1 : Logo de l’ISIMM[1]**

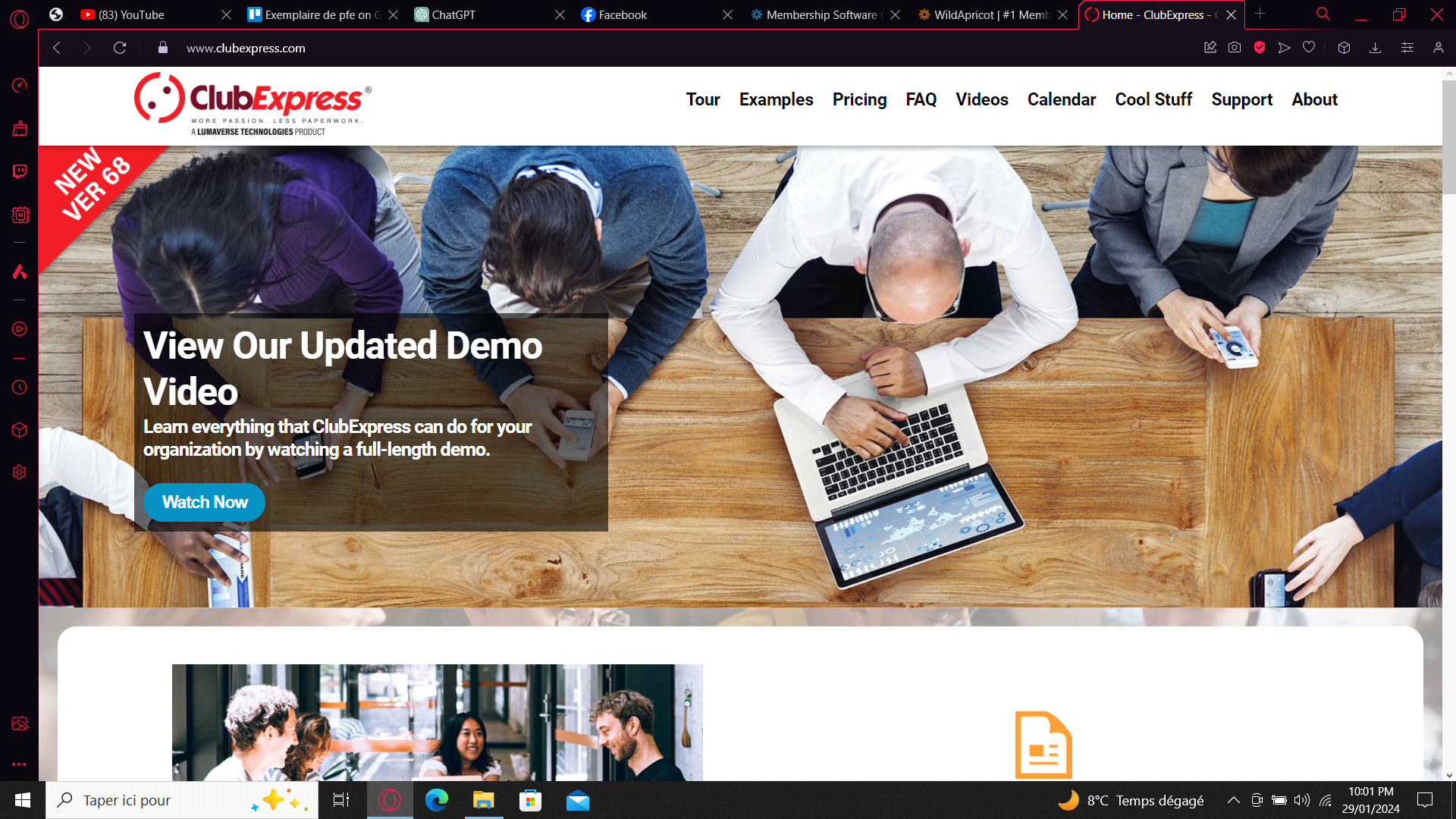
**1.3. Objectifs :**

* Préciser les objectifs à atteindre.
* L'amélioration de l'efficacité administrative des clubs.
* La facilitation de la communication entre les membres.
* Faciliter la communication entre les clubs.
* Faciliter la communication entre les clubs et l’administration.

**2. Etude de l’existant :**

**2.1. Solutions similaires :**

**Club Express :**

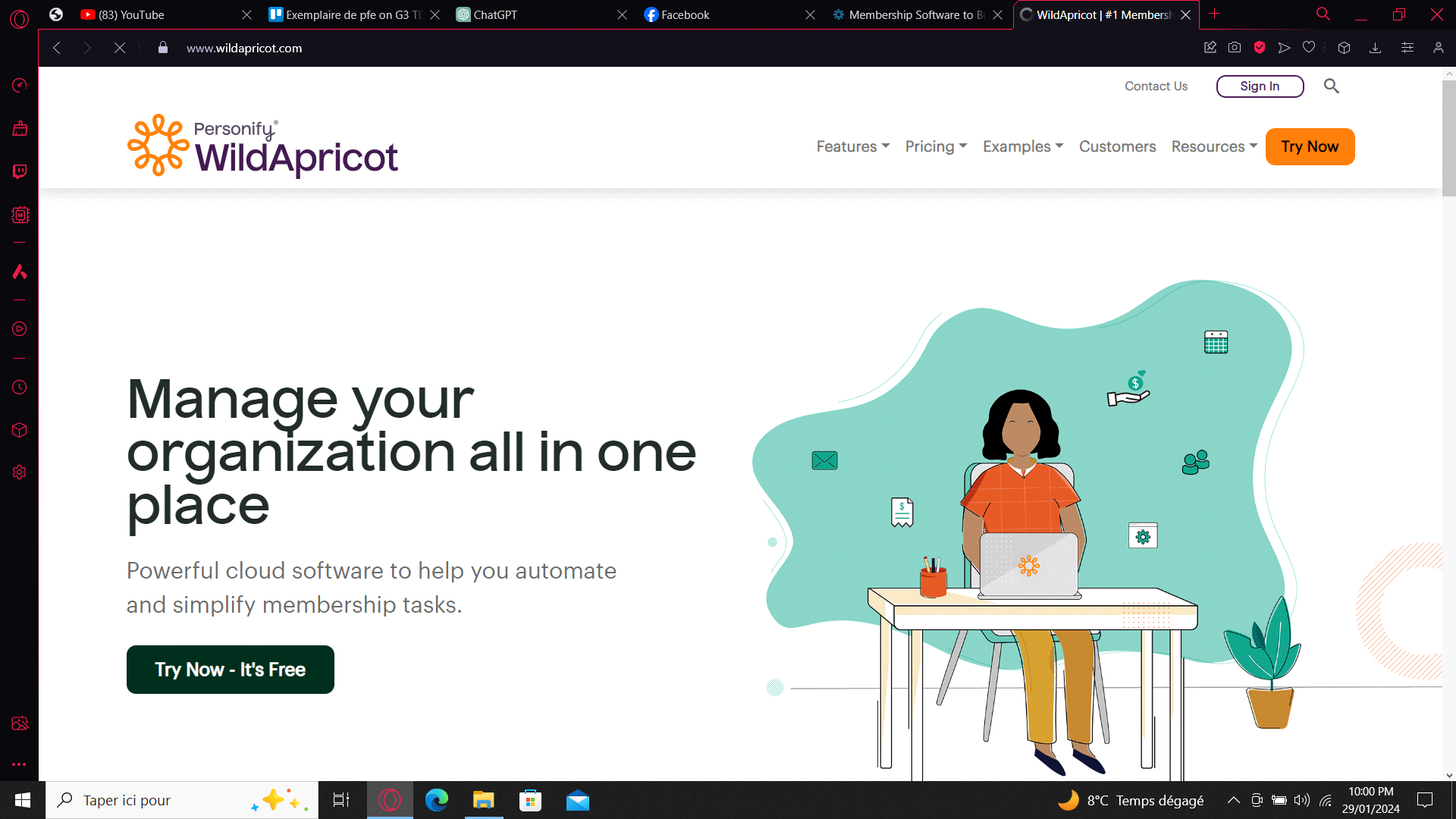


**Figure 2 : Capture du plateforme « Club Express »[2]**

La plateforme Club Express est une solution de gestion des clubs qui permet de simplifier et d'automatiser les tâches administratives, de communication et de marketing. Elle est destinée aux clubs de toutes tailles, qu'ils soient sportifs, culturels ou associatifs. La plateforme est accessible via un site web ou une application mobile.

* Gestion des membres : permet de gérer les adhésions, les cotisations, les coordonnées des membres.
* Gestion des finances : permet de gérer les recettes, les dépenses, les budgets.
* Gestion des événements : permet de gérer les événements du club, les invitations, les réservations.
* Gestion du marketing : permet de créer des campagnes de marketing, de suivre les résultats.

**WildApricot :**

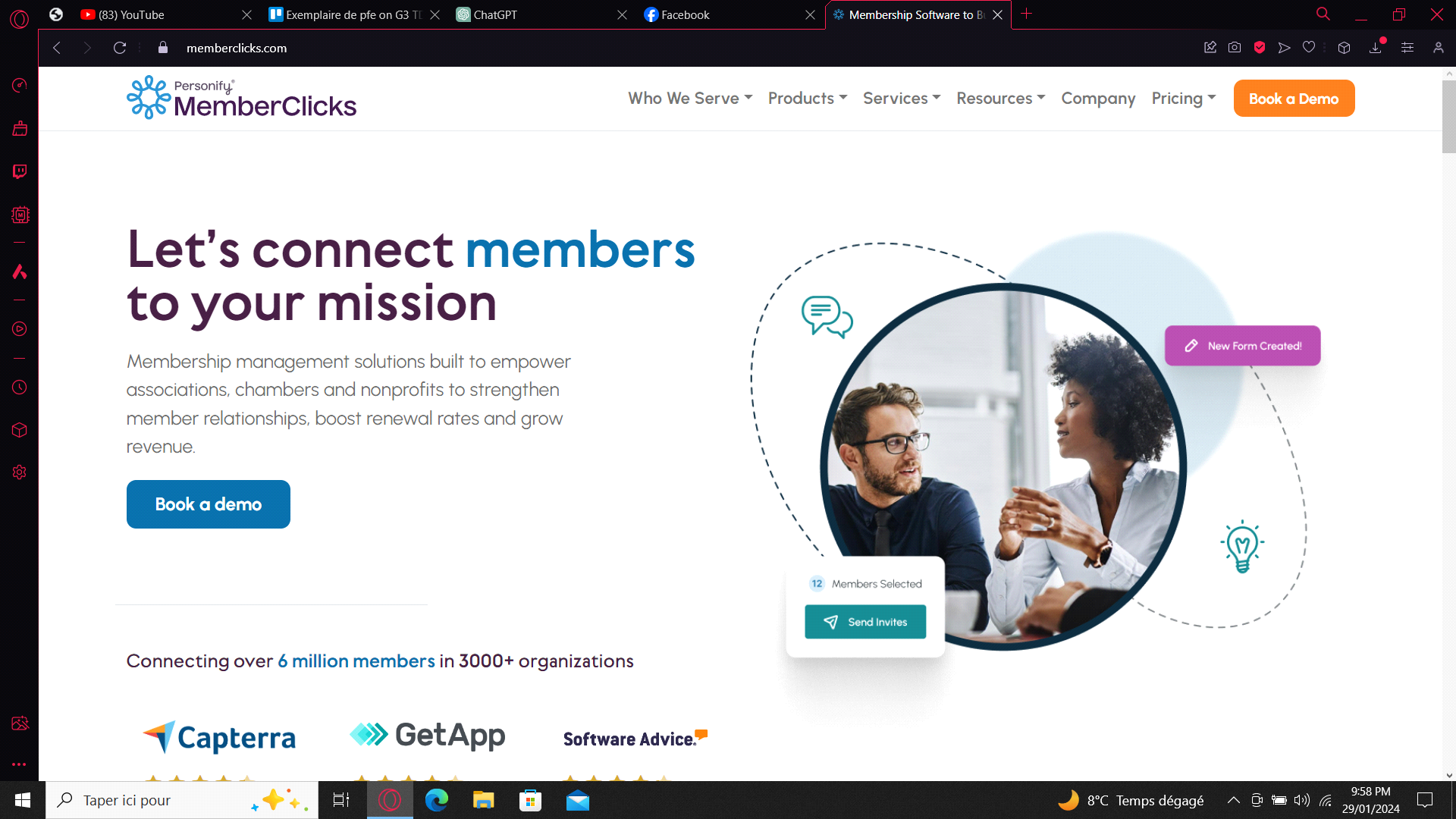


**Figure 3 : Capture du plateforme « WildApricot »[3]**

WildApricot est une plateforme de gestion de clubs complète et facile à utiliser qui aide les clubs de toutes tailles à simplifier et à automatiser leurs tâches administratives, de communication et de marketing.  
La plateforme offre une large gamme de fonctionnalités, notamment :

• Gestion des membres : permet de gérer les adhésions, les cotisations, les coordonnées des membres  
• Gestion des finances : permet de gérer les recettes, les dépenses, les budgets.  
• Gestion des événements : permet de gérer les événements du club, les invitations, les réservations, etc.  
• Gestion de la communication : permet de communiquer avec les membres, de diffuser des informations, de créer des événements.  
• Gestion du marketing : permet de créer des campagnes de marketing, de suivre les résultats.

**MemberClicks :**



**Figure 4 : Capture du plateforme « MemberClicks »[4]**

MemberClicks est une plateforme de gestion de clubs complète et facile à utiliser qui aide les clubs de toutes tailles à simplifier et à automatiser leurs tâches administratives, de communication et de marketing.  
  
La plateforme offre une large gamme de fonctionnalités, notamment :

* Gestion des membres : permet de gérer les adhésions, les cotisations, les coordonnées des membres, etc.
* Gestion des activités : permet de gérer les activités du club, les inscriptions, les horaires, etc.
* Gestion des finances : permet de gérer les recettes, les dépenses, les budgets, etc.
* Gestion des événements : permet de gérer les événements du club, les invitations, les réservations, etc.
* Gestion de la communication : permet de communiquer avec les membres, de diffuser des informations, de créer des événements, etc.
* Gestion du marketing : permet de créer des campagnes de marketing, de suivre les résultats, etc.

**2.2. Critique de l’existant :**

|  | Wild Apricot | Club Express | Membre Clicks |
| --- | --- | --- | --- |
| Gestion des membres | Oui | Oui | Oui |
| Gestion des évènements | Oui | Oui | Oui |
| Payements en lignes | Oui | Non | Non |
| Gestion des adhésions | Non | Oui | Non |
| Communication | Non | Oui | Oui |
| Tableau de bord | Non | Non | Oui |

**Tableau1: Tableau du critique de l’existant**

Suite à cette étude comparative entre les différentes solutions existantes, il s’est avéré qu’aucune de ces solutions-là ne peux pas satisfaire les utilisateurs. Pour cela nous allons développer une plateforme web pour mieux gérer les clubs.

**2.3. Solution retenue :**

Dans cette optique, il devient essentiel de présenter une solution novatrice et pratique visant à informatiser la gestion des clubs. Ainsi, les fonctionnalités clés que notre plateforme doit offrir incluent :

Notre vision pour ce nouveau projet consiste à tirer parti des points forts de différentes plateformes existantes afin de créer une solution complète et performante. Inspiré par les réussites de Wild Apricot en matière de gestion des membres, gestion des événements et paiement en ligne, nous souhaitons intégrer ces fonctionnalités fondamentales dans notre plateforme. Nous nous tournons également vers Club Express pour ses atouts en matière de gestion des adhésions, de communication efficace, et l'ajout d'un tableau de bord à la manière de MemberClicks pour offrir une expérience utilisateur holistique. En combinant ces caractéristiques, notre objectif est de proposer une plateforme unique, alliant les meilleurs aspects de chaque solution pour répondre de manière exhaustive aux besoins des clubs, simplifiant ainsi la gestion globale et améliorant la communication au sein de la communauté.

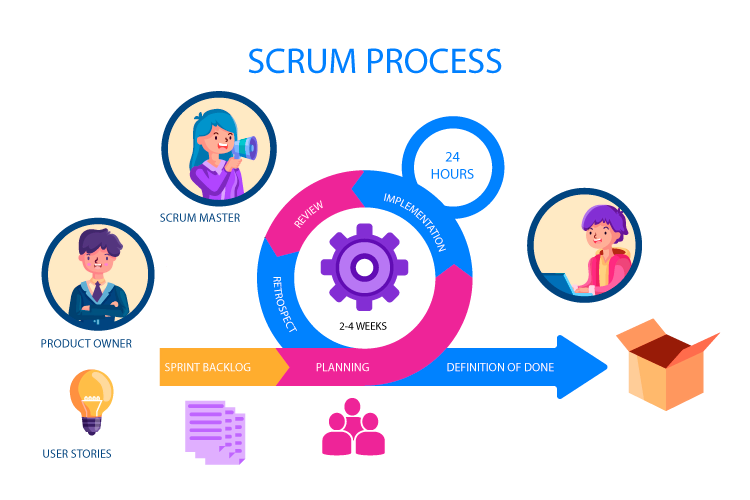
**3. Méthodologie de travail adoptée Scrum :**

**Méthode Scrum :**

La méthodologie Scrum est un cadre de gestion de projet itératif et incrémental utilisé principalement dans le développement de logiciels, mais également dans d'autres domaines où la complexité et l'incertitude sont élevées. Scrum est conçu pour permettre aux équipes de livrer régulièrement des produits de haute qualité en encourageant la flexibilité, la collaboration et l'adaptation continue aux changements. Voici les principaux éléments de la méthodologie Scrum :

1. **Équipes auto-organisées** : Les équipes Scrum sont autonomes et auto-organisées. Elles sont responsables de la planification, de l'exécution et de la livraison des fonctionnalités du produit.
2. **Sprints** : Les projets Scrum sont découpés en itérations appelées "sprints" qui durent généralement de deux à quatre semaines. Chaque sprint a un objectif clair et produit un incrément de travail potentiellement livrable.
3. **Backlog du produit** : Le backlog du produit est une liste priorisée de toutes les fonctionnalités, améliorations et correctifs à apporter au produit. Il est géré par le propriétaire du produit et peut être ajusté à tout moment pour refléter les changements de priorités ou les nouvelles informations.
4. **Backlog du sprint** : Avant le début de chaque sprint, l'équipe sélectionne un ensemble de tâches du backlog du produit pour réaliser pendant le sprint. Ce sous-ensemble est appelé le backlog du sprint.
5. **Réunions régulières** : Scrum définit plusieurs types de réunions régulières pour faciliter la communication et la collaboration. Cela inclut la planification du sprint, la revue de sprint et la rétrospective de sprint.
6. **Propriétaire du produit** : Le propriétaire du produit est responsable de maximiser la valeur du produit en gérant le backlog du produit, en définissant les exigences et en prenant des décisions concernant les fonctionnalités à développer.
7. **Scrum Master** : Le Scrum Master est responsable de veiller à ce que l'équipe suive les principes et les pratiques de Scrum. Il facilite les réunions, élimine les obstacles et aide l'équipe à s'améliorer continuellement.
8. **Transparence et adaptation** : Scrum favorise la transparence à travers des artefacts tels que les burndown charts et les kanban boards, ce qui permet à toutes les parties prenantes de suivre le progrès du projet. De plus, Scrum encourage l'adaptation continue en permettant aux équipes de réviser et d'ajuster leur planification à la fin de chaque sprint.

En résumé, Scrum offre une approche structurée et itérative pour la gestion de projets, en mettant l'accent sur la collaboration, la transparence et l'adaptation aux changements.



**Figure 5 : Modèle Scrum[5]**

**Conclusion :**

Au cours de ce chapitre, nous avons présenté la méthodologie 2TUP que nous allons adopter pour le développement de notre projet. Le chapitre suivant sera consacré au chapitre des besoins.

**Chapitre II :**

**Sprint 0 : Phase de de préparation**

**Introduction :**

Le choix et la mise en pratique de la méthode Scrum qu’un on a adopté nous amène au deuxième chapitre qui nous permet d’identifier toutes les fonctionnalités de notre système. Dans un premier temps, nous introduisons une capture initiale (c’est l’étude préliminaire) qui donne une version globale sur nos besoins. Dans un second temps, nous commençons à détailler les besoins de notre module

**1. Analyse et spécification des besoins :**

**1.1. Identification des acteurs :**

* **Président du club :** C’est un membre initialement responsable du club, il permet l’organisation et la classification des membres et permet l'accès des membres à la plateforme**.**
* **Étudiant :** c’est un simple visiteur de la plateforme. Qui est capable juste de consulter le tableau de bord des activités et d’y participer.
* **Administrateur :** Un administrateur est un utilisateur avec des autorisations étendues sur une plateforme. L'accès à l'espace administratif lui permet de gérer les utilisateurs, modifier les paramètres, et veiller au bon fonctionnement du système. Il est capable d’ajouter les gestionnaires et fait la mise à jour de liste des gestionnaires.
* **Membre :** C’est un acteur hérite des étudiants mais il a plus des options qui est capable de s’inscrire à la plateforme et ajouter les actualités.

**1.2 Capture des besoins**

La capture initiale des besoins (l’étude préliminaire) étape du processus . Elle consiste à effectuer un premier repérage des besoins fonctionnels de branche gauche (Branche fonctionnelle) et des besoins techniques de Branche droite (Branche technique).

**1.2.1 Besoins fonctionnels :**

Les besoins fonctionnels sont les fonctionnalités que notre système doit accomplir pour satisfaire les attentes de l’acteur de la plateforme.

* **S’authentifier :**

Acteur : Étudiant, Membre, Président du club, Admin

Description : Permet de vérifier chaque acteur sur la plateforme.

* **S'inscrire :**

Acteur : Étudiant

Description : Permet aux étudiants de créer un profil pour devenir membre du club.

* **Consulter activités :**

Acteur : Membre

Description : Autorise les membres à consulter le tableau de bord pour obtenir des informations sur les activités à venir.

* **Ajouter les Présidents :**

Acteur : Administrateur

Description : Donne à l'administrateur la possibilité d'ajouter des membres en tant que président du club.

* **Ajouter des activités :**

Acteur : Président du Club

Description : Autorise le président du club à ajouter des actualités ou des annonces importantes pour les membres.

* **Classifier les Membres :**

Acteur : Président du club

Description : Permet au président du club de gérer l'accès des membres en fonction de leur position ou de leur rôle dans le club.

* **Gérer le Tableau de Bord :**

Acteur : Président du Club

Description : Donne aux administrateurs et aux présidents du club la possibilité de gérer et d'ajouter des événements au tableau de bord.

* **Supprimer des Membres :**

Acteur : Administrateur

Description : Permet à l'administrateur de supprimer des membres inactifs ou indésirables.

* **Gérer Droits d'Accès :**

Acteur : Administrateur

Description : Permet à l'administrateur de définir les niveaux d'autorisation pour différents membres (administrateurs, présidents, membres normaux).

* **Gérer des activités** :

Acteur : Président du Club

Description : Autorise les administrateurs et les présidents du club à créer et planifier des événements.

* **Inscription aux activités :**

Acteur : Membre

Description : Permet aux membres de s'inscrire aux activités à venir.

* **Suivre les Présences :**

Acteur : Administrateur

Description : Donne à l'administrateur la possibilité de suivre la présence des membres aux événements.

* **Évaluer des Événements :**

Acteur : Membre

Description : Autorise les membres à laisser des commentaires ou des évaluations sur les événements.

* **Notifications par E-mail :**

Acteur : Membre

Description : Envoie des notifications par e-mail aux membres pour les événements à venir, les mises à jour importantes, etc.

* **Sessions de Formation :**

Acteur : Président du Club

Description : Autorise le président du club à organiser des sessions de formation pour les membres.

**1.2.2. Besoins non fonctionnels :**

Les besoins non fonctionnels sont également appelés les critères de qualité décrivent les caractéristiques qui ne sont pas liées directement aux fonctionnalités spécifiques d’un système, mais qui ont un impact sur sa qualité et son efficacité. Notre plateforme doit répondre aux critères suivants :

**1-Performance :**

* + Notre plateforme doit être capable de gérer efficacement un grand nombre d'utilisateurs simultanés.
  + Les temps de réponse doivent être rapides pour assurer une expérience utilisateur fluide.

**2-Sécurité :**

* + Notre plateforme doit garantir la confidentialité des données des membres du club.
  + Les mécanismes d'authentification et d'autorisation doivent être robustes pour empêcher l'accès non autorisé.

**3-Extensibilité :**

* + Notre plateforme doit pouvoir évoluer facilement pour prendre en charge de nouveaux clubs ou fonctionnalités.
  + L'architecture doit permettre l'intégration avec d'autres systèmes ou services externes.

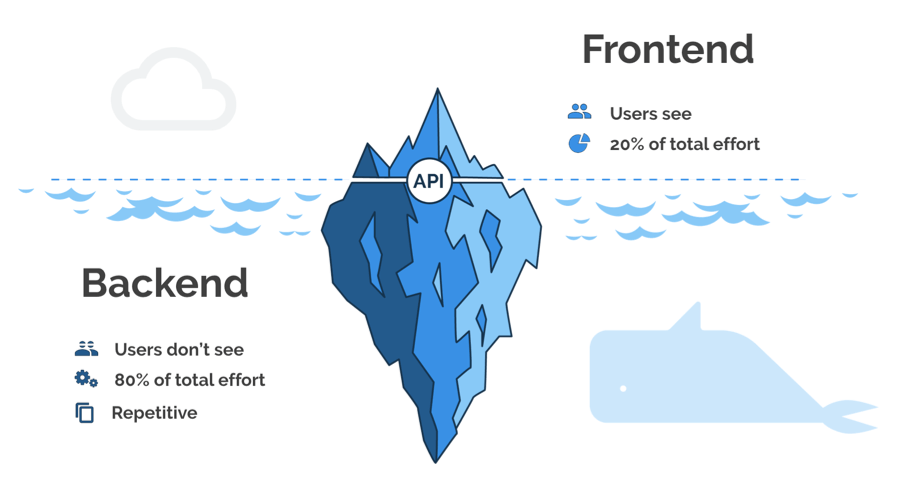
**1.3. Spécification des besoins fonctionnels :**

Le capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y. Dans cette partie nous présentons les besoins fonctionnels 24 précédente par un ensemble de graphe pour mieux comprendre les fonctionnalités qui offre note module ces sont les cas d’utilisation L'identification des cas d’utilisation une première fois, nous donne un aperçu des fonctionnalités futures que doit implémenter notre plateforme.

**2.Etude technique :**

**2.1. Environnement de travail :**

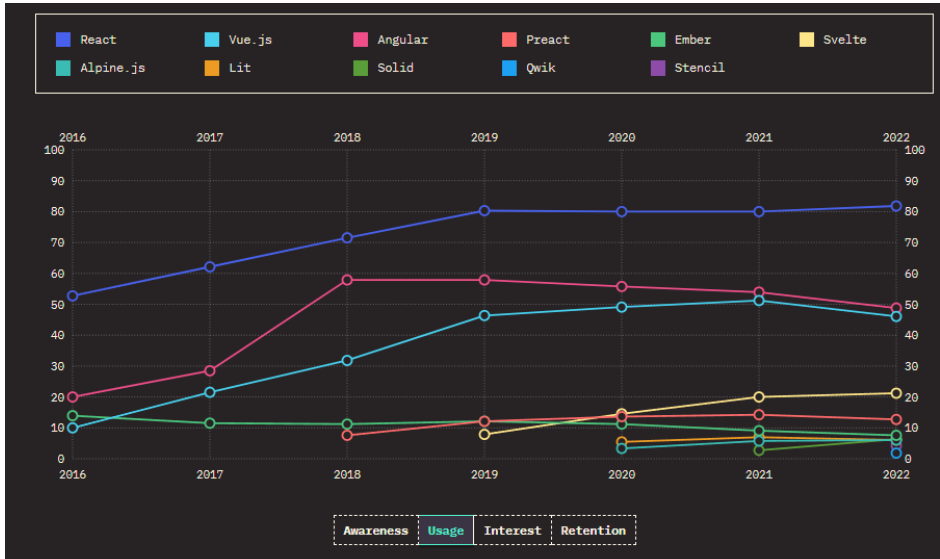
Dans cette section, nous rapporterons sur l'étude des différents langages et Framework utilisés dans la partie back-end et front-end afin de faire un choix judicieux pour la réalisation de notre projet.



**Figure6 : Frontend & Backend[6]**

**2.1.1 Environnement de développement Front-end :**

Lorsque l'on parle de Front-End, il s'agit finalement des éléments du site que l'on voit à l'écran et avec lesquels on peut interagir. Coté Front nous allons créer une application web pour cela nous commençons par présenter les différents frameworks respectif destiné à notre application.



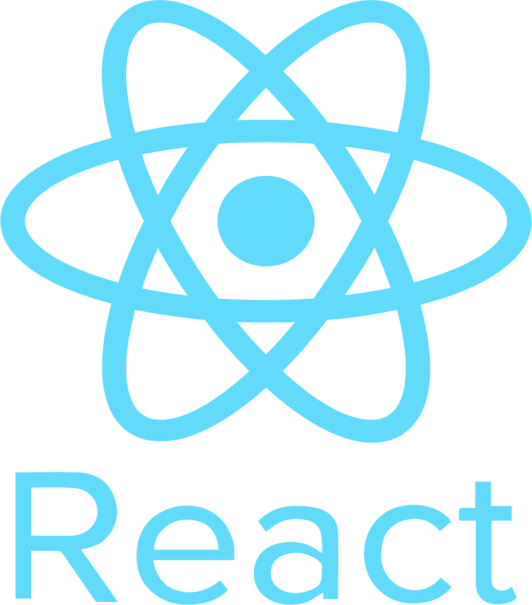
**Figure7 :** **Popularité des frameworks JavaScript au fil du temps [7]**

**Choix :**

Afin d'être en mesure de réaliser notre application web nous allons choisir le Framework React *.*React (aussi appelé React.js ou ReactJS) est une [bibliothèque](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_Logicielle) [JavaScript](https://fr.wikipedia.org/wiki/JavaScript) [libre](https://fr.wikipedia.org/wiki/Logiciel_libre). Elle est maintenue par [Meta](https://fr.wikipedia.org/wiki/Meta_(entreprise)) (anciennement [Facebook](https://fr.wikipedia.org/wiki/Facebook)) ainsi que par une communauté de développeurs individuels et d'entreprises depuis [2013](https://fr.wikipedia.org/wiki/2013_en_informatique).

Le but principal de cette bibliothèque est de faciliter la création d'[application web monopage](https://fr.wikipedia.org/wiki/Application_web_monopage), via la création de composants dépendant d'un état et générant une page (ou portion) [HTML](https://fr.wikipedia.org/wiki/HTML) à chaque changement d'état.

React est une bibliothèque qui ne gère que l'interface de l'application, cette interface étant considérée comme la vue dans le modèle [MVC](https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur). Elle peut ainsi être utilisée avec une autre bibliothèque ou un [framework](https://fr.wikipedia.org/wiki/Framework) MVC comme [AngularJS](https://fr.wikipedia.org/wiki/AngularJS). La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un [DOM](https://fr.wikipedia.org/wiki/Document_Object_Model) virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité.



**Figure8 : Logo de React[8]**

**Mapping entre MySQL et NoSQL:**

Migration de données :

Il s'agit du processus de déplacer des données d'une base de données relationnelle à une base de données NoSQL.

Peut nécessiter la transformation de données pour s'adapter au format NoSQL. Par exemple, convertir des tables en documents ou en paires clé-valeur.

Utilise des outils spécifiques pour faciliter la migration, tels que "Apache NiFi", "Talend", ou des scripts personnalisés.

Outils de synchronisation :

Ces outils permettent de synchroniser des données entre des bases de données MySQL et NoSQL, souvent utilisés pour des scénarios de réplication ou de sauvegarde.

Certains outils permettent des transformations pendant la synchronisation, facilitant ainsi le mapping entre des structures différentes.

Exemples de transformations courantes :

Table en documents : Convertir des lignes de table MySQL en documents dans une base de données NoSQL comme MongoDB. Les champs de la table deviennent des champs du document.

Clés étrangères : En NoSQL, il n'y a pas de relation de clés étrangères comme en SQL. Pour conserver les relations, vous pouvez utiliser des références ou des identifiants pour relier des documents.

Transactions : Si vous migrez de MySQL à NoSQL, gardez à l'esprit que NoSQL ne prend pas en charge les transactions de la même manière. Planifiez le design des données en conséquence.

**2.1.1 Environnement de développement Back-end:**

MongoDB est une base de données NoSQL orientée document pour le stockage de grandes quantités de données. MongoDB utilise des collections et des documents au lieu de tableaux et de lignes, comme dans les bases de données relationnelles traditionnelles.

Les documents sont constitués de paires clé-valeur, qui constituent l'unité de données de base de MongoDB. Les collections sont l'équivalent des tables des bases de données relationnelles, car elles comprennent des ensembles de documents et de fonctions.

Il fonctionne sur plusieurs systèmes d'exploitation, notamment Windows, Linux, OS X, Solaris et FreeBSD.

En tant que programme de base de données orienté document, il offre aux utilisateurs une flexibilité extraordinaire lors de l'ajout ou de la récupération de grandes quantités de données non structurées dans une base de données.

Les utilisateurs peuvent choisir entre trois versions de MongoDB :

* Cloud (base de données en tant que service)
* On-Premises (MongoDB en local)
* Realm (développement d'applications pour le Web, Android et iOS).
* Les utilisateurs peuvent également choisir entre les options sans serveur, serveur dédié et serveur partagé.

### **Quelles sont les caractéristiques de MongoDB ?**

MongoDB possède plusieurs fonctionnalités essentielles qui en font un excellent SGBD pour de nombreuses entreprises. Certaines de ces fonctionnalités comprennent :

* La réplication des données qui améliore la stabilité et la disponibilité.
* Équilibrage optimal de la charge qui s'adapte à l'augmentation des demandes des clients
* Le sharding intégré pour une évolutivité rapide
* Des schémas flexibles orientés documents qui permettent des analyses en temps réel
* Une meilleure indexation qui améliore l'exécution des requêtes

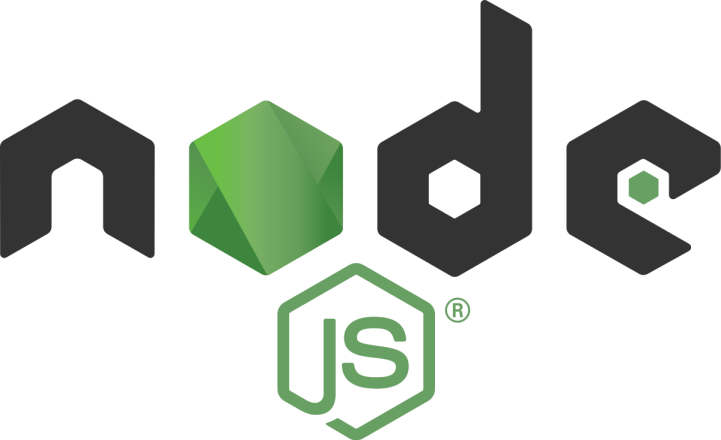


**Figure9 : Logo de MongoDB[9]**

Choix :

Afin de faire le bon choix nous allons étudier la plateforme node js. Node Js est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseaux qui doivent pouvoir monter en charge. Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur HTTP.

Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur. La figure 15 représente le logo de Node.



**Figure10 : Logo de nodeJS[10]**

Express.js : Express est le cadre applicatif actuellement le plus populaire dans Node.

Elle est la librairie sous-jacente pour un grand nombre d'autres cadres applicatifs web pour Node.



**Figure11 : Logo de ExpressJS[11]**

**3. Pilotage avec scrum :**

Dans cette partie, nous allons décrire et réaliser le Backlog de produit ainsi que la formalisation de diagramme global de cas d’utilisation et la planification de nos sprints.

**3.1. Définition des rôles :**

Ci-dessous nous, allons citer les acteurs de notre projet Scrum.

**Product owner :** Mme Aljia bouzidi

**Scrum Master :** Mme Aljia Bouzidi

**Les développeurs :** Ines Chebbi, Maryem Aloui,Louay Elghali,Haroun Hassen,Aziz Ferchichi,Aziz ben Tanfous

**3.2. Backlog de produit :**

Le Backlog est une liste des tâches proposées définissant les caractéristiques d’un produit. Il s’agit de l’outil de travail principal du Product Owner qui se charge de recueillir les besoins auprès des parties prenantes et de les transformer en liste de fonctionnalités prêtes à être développées par l’équipe de développement. Communément nous pouvons formaliser ces items sous forme d’User Stories ordonnées selon la priorité des besoins. Pour faire cela, il faut, tout d’abord, se rappeler de concepts clés suivants :

● Rang : C’est l’identifiant de chaque User Story.

● Thème : Il Représente le thème du User Story.

● User Story : C’est l’histoire utilisateur décrivant un besoin fonctionnel désiré par le client.

● Priorité : Elle est représentée par la méthode de priorisation ‘’MoSCoW’’ dans laquelle chaque User Story est priorisé comme l’un des suivants :

Must : La fonctionnalité doit être faite.

Should : La fonctionnalité devrait être faite dans la mesure du possible.

Could : La fonctionnalité pourrait être faite si elle n’a pas d’impact sur les autres tâches.

Won ’t : La fonctionnalité ne sera pas faite cette fois mais sera faite plus tard.

Le Backlog du produit présenté dans le tableau comprend les champs suivants :

| **Id** | **Thème** | **Rang** | **User Story** | **Priorité** |
| --- | --- | --- | --- | --- |
| 1 | **Gestion d'accès** | 1.1 | En tant que membre, président du club et admin, je veux m’authentifier | Must |
| 1.2 | En tant qu’étudiant, membre et président du club je veux m’inscrire sur la plateforme | Must |
| 2 | **Inscription** | 2.1 | En tant que membre, président du club, je peux m’inscrire dans un club ou dans une activité | Should |
| 3 | **Consultation** | 3.1 | En tant que étudiant, membre et président du club , je veux consulter tableau de bord | Should |
| 3.2 | En tant que membre, président du club, je veux consulter notifications et tâches | Should |
| 4 | **Communiquer** | 4.1 | En tant que membre, président du club et admin, je veux communiquer avec les autres acteurs. | Could |
| 5 | **Gestion des membres** | 5.1 | En tant que président du club et admin, je veux gérer les membres | Should |
| 6 | **Gestion des activités** | 6.1 | En tant que président du club je veux gérer lea activités et les ajouter sur le tableau de bord | Should |
| 7 | **Participer aux activités** | 7.1 | En tant qu’étudiant, membre et président du club, je veux participer aux activités | Should |
| 8 | **Gestion droits d'accès** | 8.1 | En tant que admin je veux gérer les droits d'accès des autres acteurs | Should |

**Tableau2 : Backlog du produit**

**3.3 Planification des sprints :**

Le pilotage de projet se fait en divisant nos User Stories sur des périodes pour avoir un travail bien organisé et guidé. Pour cela, nous avons divisé nos tâches sur des :

• Release : Une release correspond à la livraison d'une version de l’application. C’est la période qui s’écoule entre le début du travail sur cette version et sa livraison et qui passe par une série de sprints successifs.

• Sprint : C'est le terme utilisé dans Scrum pour itération, c’est un bloc de temps dont le but est de créer un incrément du produit potentiellement livrable. Ce but doit être défini en terme métier et non en terme technique pour qu'il soit compréhensible par les 42 membres ne faisant pas partie de l’équipe.

| **Release** | **Sprint** | **Tâche** | **Durée** |
| --- | --- | --- | --- |
| Release 1 : | Sprint 1 | Gestion d'accès | 3 |
| Inscription | 2 |
| Release2 : | Sprint 2 | Gestion des membres | 4 |
| Release 3 : | Sprint 3 | Gestion activités | 2 |
| Release 4 : | Sprint 4 | Communication | 2 |

**Tableau3 : Tableau de planification des sprints**

**Conclusion :**

Dans ce chapitre, nous avons fait une analyse des besoins nous a permis d'avoir une vision plus claire sur notre sujet et nous avons mis en relief l'exposition des différents Sprints de notre projet ainsi que les technologies utilisées. Le chapitre suivant sera consacré à l'analyse détaillée de premier release.

**Chapitre III :**

**Sprint1 & sprint2**

**Introduction**

Dans le chapitre précédent, nous avons choisi d'adopter la méthodologie « Scrum » pour la conception de notre futur système. En fait, la première phase de cette dernière et la phase de planification et architecture.

Cette étape est la plus importante dans le cycle de développement « Scrum » puisqu'elle influence directement la réussite des sprints. Les travaux réalisés dans cette période de temps conduisent à construire une bonne vision du produit, identifier les rôles des utilisateurs et dégager les fonctionnalités principales afin de produire le Backlog initial ainsi qu'une première planification des sprints

A ce niveau, nous détaillerons notre deux premiers releases composés de deux sprints et ayant pour mission la gestion d’accès et l’inscription pour le premier release et gestion des membres pour le deuxième . D’abord, nous commençons par la planification illustrée par le Backlog du produit de ces deux sprints. Ensuite, nous élaborerons la partie de l’analyse et nous exposerons notre étude conceptuelle.

**1. Itération du sprint 1 :**

**1.1. Spécification des besoins du sprint 1 :**

**1.1.1. Backlog du produit :**

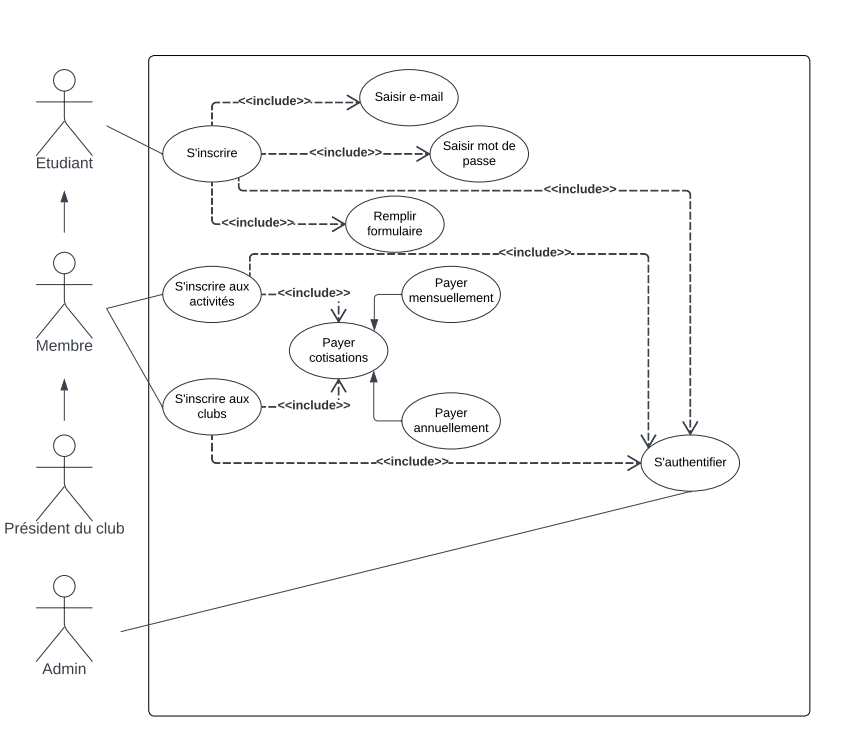
Notre échelle de travail est organisée sur une base de 3 jours par semaine, avec une durée de travail de 4 heures par jour. Ainsi, chaque semaine comprendra un total de 12 heures de travail. Le Backlog de ce release, qui est présenté dans le tableau 4 , détaille une liste des tâches qui ont été identifiées et qui doivent être réalisées avant la fin du sprint 1.

| **Id** | **Thème** | **User story** | **Estimation (durée par heure)** |
| --- | --- | --- | --- |
| **1** | **Gestion d'accès** | En tant que membre, admin ou président du club je veux avoir mon compte personnel | 8h |
| En tant que membre, admin ou président du club je veux me connecter. | 5h |
| En tant que membre, admin ou président du club je veux réinitialiser mon mot de passe quand je l’oubli | 4h |
| En tant que membre , admin ou président du club je veux me déconnecter | 4h |
| **2** | **Inscription** | En tant que étudiant je veux m’inscrire sur la plateforme | 5h |
| En tant que membre, président du club, je veux m’inscrire dans un club | 7h |
| En tant que membre, président du club, je veux m’inscrire dans une activité | 8h |

**Tableau4 : Backlog du sprint1**

**1.1.2. Diagramme de cas d’utilisation du sprint 1 :**

La figure si dessous représente le diagramme de cas d’utilisation du sprint 1.

****

**Figure12 : Diagramme de cas d’utilisation du sprint1**

**1.2. Analyse de sprint 1 :**

Dans cette partie, nous présenterons l’analyse du domaine du Sprint 1 et l’analyse de la réalisation des cas d’utilisation.

**1.2.1. Analyse de la réalisation de sprint 1 :**

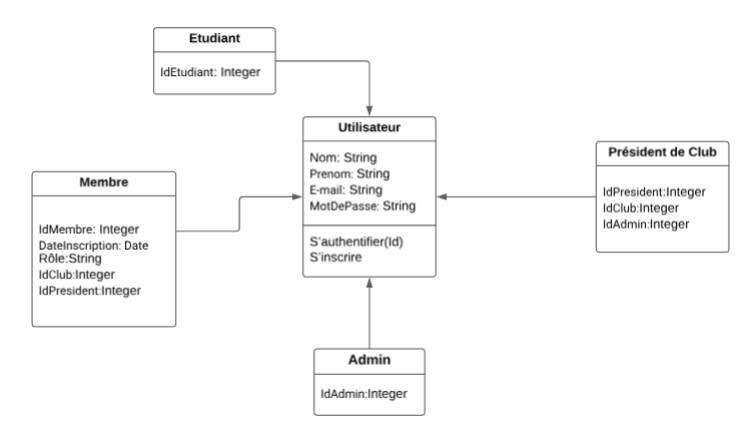
Dans cette partie, nous recensons pour chaque thème les différentes classes qui interviennent à sa réalisation dans le but de les faire intervenir dans les modélisations statique et dynamique. Le tableau 5 résume pour chaque thème du Sprint 1 les classes qui participent à sa réalisation.

| **Classe** | **Description** | **Attributs et types** |
| --- | --- | --- |
| **Utilisateur** | Elle encapsule les informations principales  Relatives aux utilisateurs de l’application | Nom: String  Prénom: String  E-mail: String  MotDePasse: String  S’authentifier (Id)  S’inscrire |
| **Etudiant** | Elle hérite de la classe « Utilisateur » et contient les attributs relatifs d'étudiant. | IdEtudiant: Integer |
| **Membre** | Elle hérite de la classe « Utilisateur » et contient les attributs supplémentaires relatifs à membre. | IdMembre: Integer  DateInscription: Date  Rôle: String  IdClub: Integer  IdPresident: Integer |
| **Président de Club** | Elle hérite de la classe « Utilisateur » et contient les attributs supplémentaires relatives à Président de Club. | IdPresident: Integer  IdClub: Integer  IdAdmin: Integer |
| **Admin** | Elle hérite la classe « Utilisateur » et contient  Les attributs supplémentaires relatives à  l’admin. | IdAdmin: Integer |

**Tableau5: Analyse du domaine du sprint 1**

**1.2.2. Diagramme de classe d’analyse du Sprint 1 :**

La figure 13 représente le diagramme de classe d’analyse du premier sprint.

****

**Figure13 : Diagramme de classe d’analyse du sprint1**

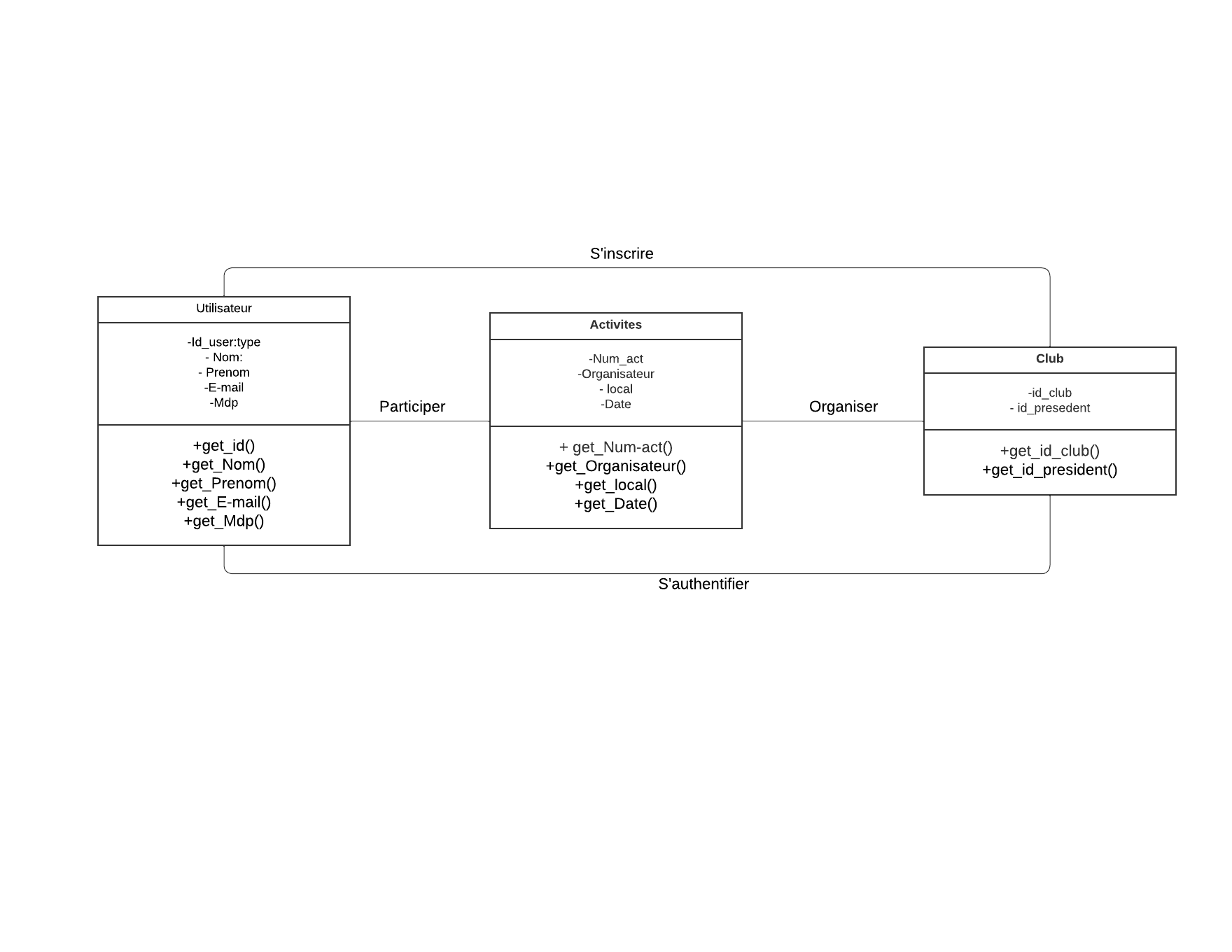
**1.3 Conception du Sprint 1 :**

**1.3.1 vue statique du Sprint 1 :**

Dans cette section nous allons représenter la vue statique du Sprint 1 en utilisant le diagramme des classes qui sera conceptualisé selon le patron MVC.

**Diagramme de classe :**

La figure si dessous représente le diagramme de classe de sprint 1.



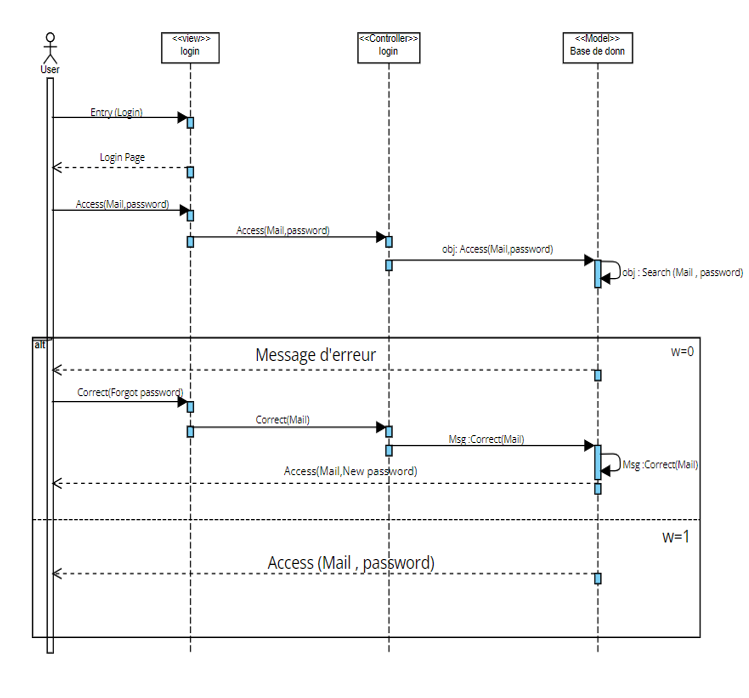
**Figure 14 : Diagramme de classe de sprint 1**

**1.3.2 Vue dynamique du Sprint 1 :**

Dans cette section, nous allons examiner de près la vue dynamique de la version 1 en utilisant le diagramme de séquence objet, lequel sera structuré selon le modèle MVC.

**Diagramme de séquence :**

La figure si dessous représente le diagramme de séquence de sprint 1.

****

**Figure 15 : Diagramme de séquence du sprint 1**

**2. Itération du sprint2 :**

**2.1. Spécification des besoins du sprint 2 :**

**2.1.1. Backlog du produit :**

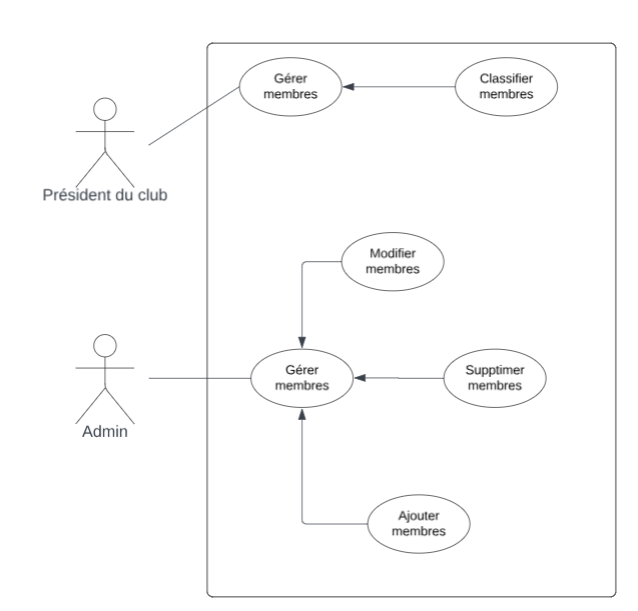
Le Backlog du deuxième sprint présenté par le tableau 6 contient une liste des tâches identifiées qui devront être réalisées avant la fin du sprint.

| **Id** | **Thème** | **User story** | **Estimation (durée par heures)** |
| --- | --- | --- | --- |
| **1** | **Gestion des membres** | En tant que admin je peux ajouter membre | 5h |
| En tant que président du club je peux classifier membre. | 5h |
| En tant que admin je peux supprimer membre. | 6h |
| En tant que admin je peux modifier membre. | 6h |

**Tableau 6 : Backlog du sprint 1**

**2.1.2. Diagramme de cas d’utilisation du sprint 2 :**

La figure si dessous représente le diagramme de cas d’utilisation du sprint 2.

****

**Figure 16 : Diagramme de cas d’utilisation du sprint 2**

**2.2. Analyse de sprint 2 :**

Dans cette partie, nous présenterons l’analyse du domaine du Sprint 2 et l’analyse de la réalisation des cas d’utilisation.

**2.2.1. Analyse de la réalisation de sprint 2 :**

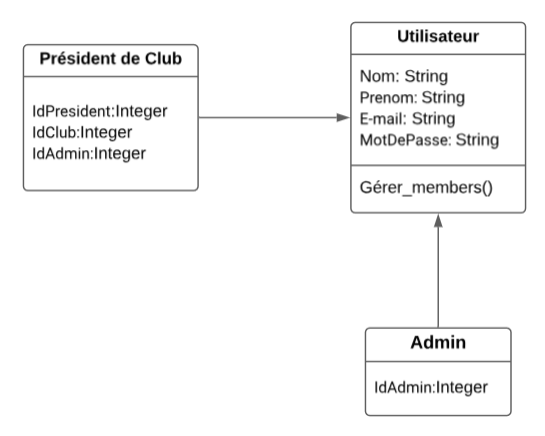
Le tableau 7 résume pour chaque thème du Sprint 2 les classes qui participent à sa réalisation.

| **Classe** | **Description** | **Attributs et types** |
| --- | --- | --- |
| **Utilisateur** | Elle encapsule les informations principales relatives aux utilisateurs de l’application | Nom: String  Prénom: String  E-mail: String  MotDePasse: String  Gérer\_Membre() |
| **Président de Club** | Elle hérite la classe « Utilisateur » et contient les attributs supplémentaires relatives à Président de Club. | IdPresident: Integer  IdClub: Integer  IdAdmin: Integer |
| **Admin** | Elle hérite la classe « Utilisateur » et contient les attributs supplémentaires relatives à  l’admin. | IdAdmin: Integer |

**Tableau7 : Analyse du domaine du sprint 2**

**2.2.2. Diagramme de classe d’analyse du Sprint 2:**

La Figure 17 représente le diagramme de classe d’analyse du deuxième sprint.

****

**2.2. Conception du Sprint 2 :**

**2.2.1. Vue statique du Sprint 2 :**

Dans cette partie, nous allons décrire la vue statique du Sprint 2 en utilisant le diagramme de classes, lequel sera conçu en suivant le modèle MVC.

**Diagramme de classe :**

La figure si dessous représente le diagramme de classe de sprint 2.

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

Description générée automatiquement**

**Figure 18 : Diagramme de classe du sprint 2**

**2.2.2. Vue dynamique du Sprint 2 :**

Dans cette partie, nous allons détailler la vue dynamique de sprint 2 en utilisant le diagramme de séquence objet qui sera conceptualisé selon le patron MVC.

**Diagramme de séquence :**

Une image contenant texte, nombre, Parallèle, reçu

Description générée automatiquementLa figure si dessous représente le diagramme de séquence du sprint 2.

**Figure 19 : Diagramme de séquence du sprint 2**

**Conclusion :**

Ce chapitre vient de présenter les deux premières releases du projet où nous avons planifié, détaillé et montré les interfaces qui lui correspondent. Dans le chapitre qui suit, notre effort sera consacré à produire une nouvelle release.

**Chapitre IV :**

**Sprint 3 & Sprint 4**

**Introduction :**

Dans ce chapitre, nous détaillerons nos deux dernières releases composées de deux sprints et ayant pour mission la gestion des activités pour le premier release et communication pour le deuxième. D’abord, nous commençons par la planification illustrée par le Backlog du produit de ces deux sprints. Ensuite, nous élaborerons la partie de l’analyse et nous exposerons notre étude conceptuelle.

**1. Itération du sprint3 :**

**1.1. Spécification des besoins du sprint 3 :**

**1.1.1. Backlog du produit :**

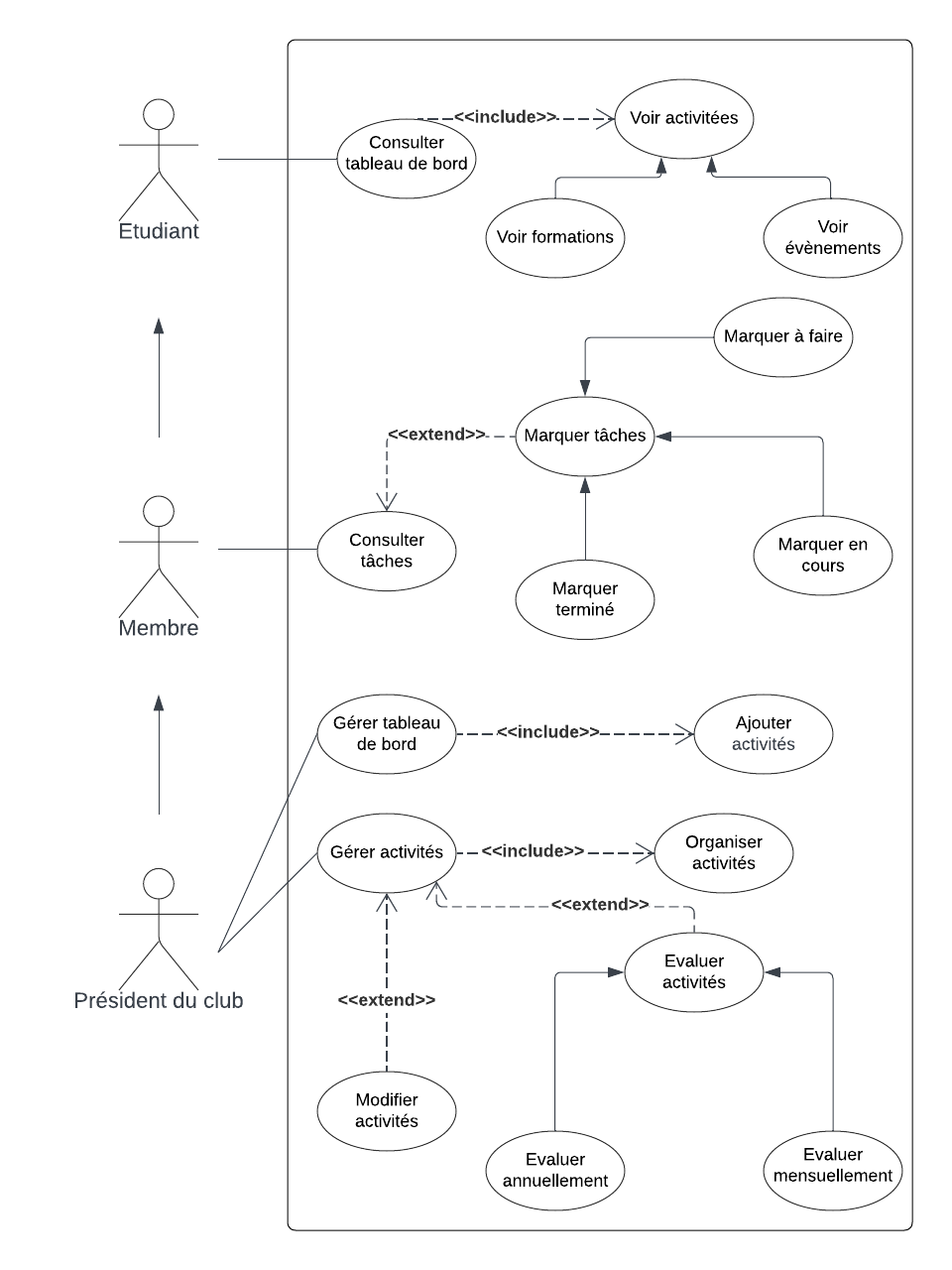
Le Backlog du troisième sprint présenté par le tableau 8 contient une liste des tâches identifiées qui devront être réalisées avant la fin du sprint.

| **Id** | **Thème** | **User story** | **Estimation (durée par heures)** |
| --- | --- | --- | --- |
| 1 | **Gestion des activités** | En tant que président je peux organiser les activités | 5h |
| En tant que président je peux évaluer les activités (mensuellement ou annuellement) | 6h |
| En tant que président je peux modifier les activités | 7h |
| En tant que membre et président du club je peux consulter mes tâches et je peux les marquer comme terminés, en cours ou à faire. | 6h |
| En tant qu'étudiant, membre et président du club je peux participer aux activités (formations, événements) | 5h |
| 2 | **Gestion de tableau de bord** | En tant que étudiant et membre, président du club, je peux consulter tableau de bord càd voir les activités: formations ou évènements | 6h |
| En tant que président du club je peux ajouter les activités au tableau de bord | 5h |

**Tableau8 : Backlog du sprint 3**

**1.1.2. Diagramme de cas d’utilisation du sprint 3 :**

La figure si dessous représente le diagramme de cas d’utilisation du sprint 3.

****

**Figure20 : Diagramme de cas d’utilisation du sprint 3**

**1.2. Analyse de sprint 3 :**

Dans cette partie, nous présenterons l’analyse du domaine du Sprint 3 et l’analyse de la réalisation des cas d’utilisation.

**1.2.1 Analyse de la réalisation de sprint 3 :**

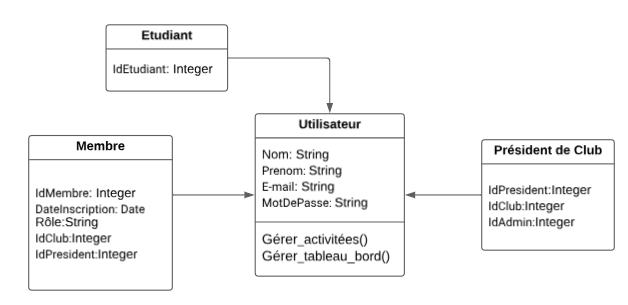
Dans cette partie, nous recensons pour chaque thème les différentes classes qui interviennent à sa réalisation dans le but de les faire intervenir dans les modélisations statique et dynamique. Le tableau 9 résume pour chaque thème du Sprint 3 les classes qui participent à sa réalisation.

| **Classe** | **Description** | **Attributs et types** |
| --- | --- | --- |
| **Utilisateur** | Elle encapsule les informations principales relatives aux utilisateurs de l’application | Nom: String  Prenom: String  E-mail: String  MotDePasse: String  Gérer\_activités()  Gérer\_tableau\_bord() |
| **Etudiant** | Elle hérite de la classe « Utilisateur » et contient les attributs relatifs d'étudiant. | IdEtudiant: Integer |
| **Membre** | Elle hérite de la classe « Utilisateur » et contient les attributs supplémentaires relatifs à membre. | IdMembre: Integer  DateInscription: Date  Rôle: String  IdClub: Integer  IdPresident: Integer |
| **Président de Club** | Elle hérite la classe « Utilisateur » et contient les attributs supplémentaires relatives à Président de Club. | IdPresident: Integer  IdClub: Integer  IdAdmin: Integer |

**Tableau9: Analyse du domaine du sprint 3**

**1.2.2. Diagramme de classe d’analyse du Sprint 3 :**

La Figure 21 représente diagramme de classe d’analyse du troisième sprint.

****

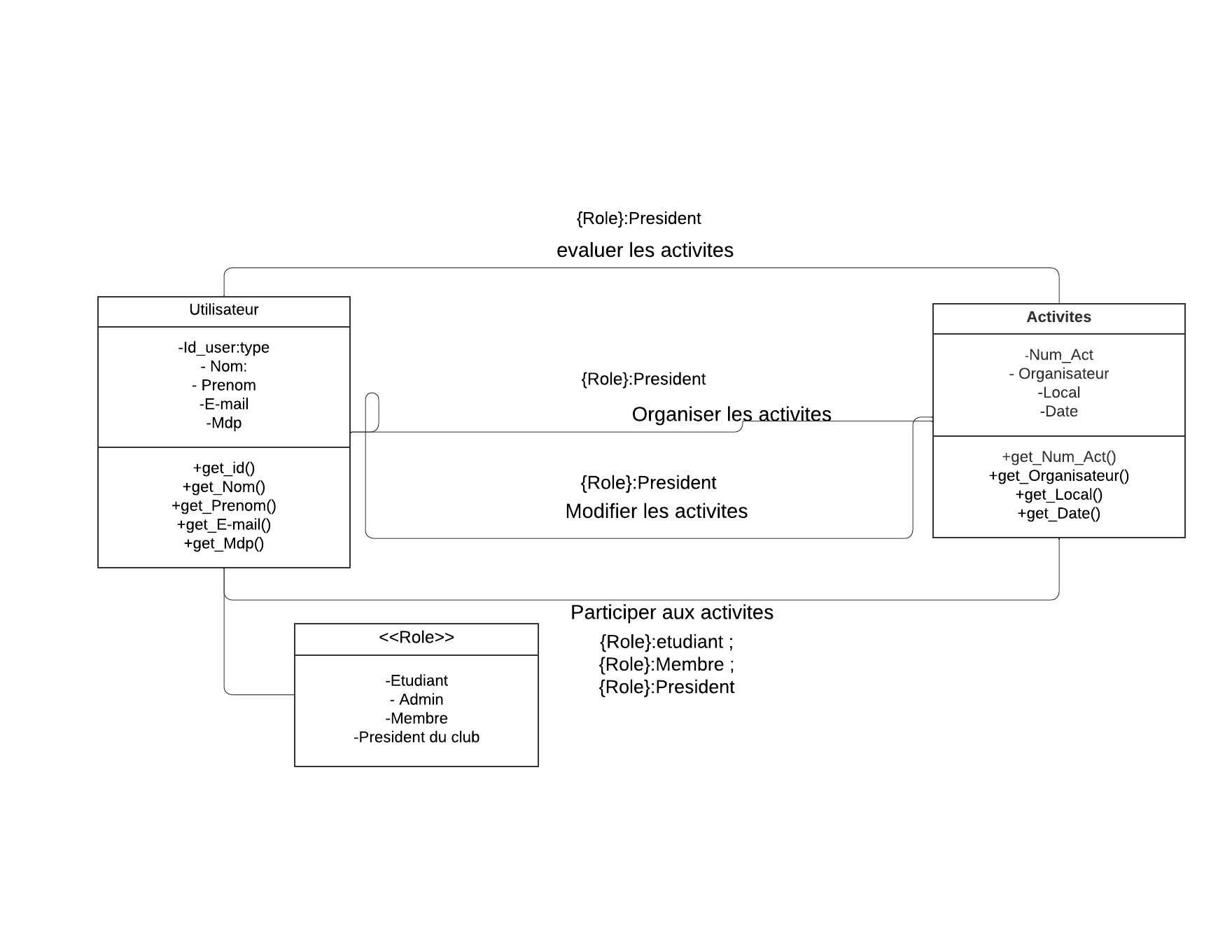
**Figure21 :**

**1.3. Conception du Sprint 3 :**

**1.3.1. Vue statique du Sprint 3 :**

**Diagramme de classe :**

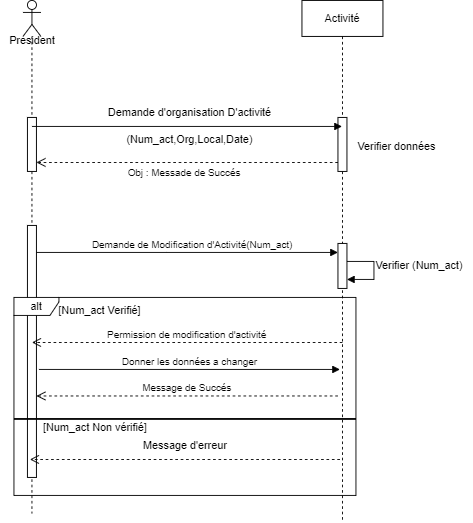
Dans cette partie, nous allons décrire la vue statique du Sprint 3 en utilisant le diagramme de classes, lequel sera conçu en suivant le modèle MVC.

****

**Figure22 : Diagramme de classe de sprint**

**1.3.2. Vue dynamique du Sprint 3 :**

**Diagramme de Séquence du sprint :**

****

**2.Itération du sprint4 :**

**2.1. Spécification des besoins du sprint 4 :**

**2.1.1. Backlog du produit :**

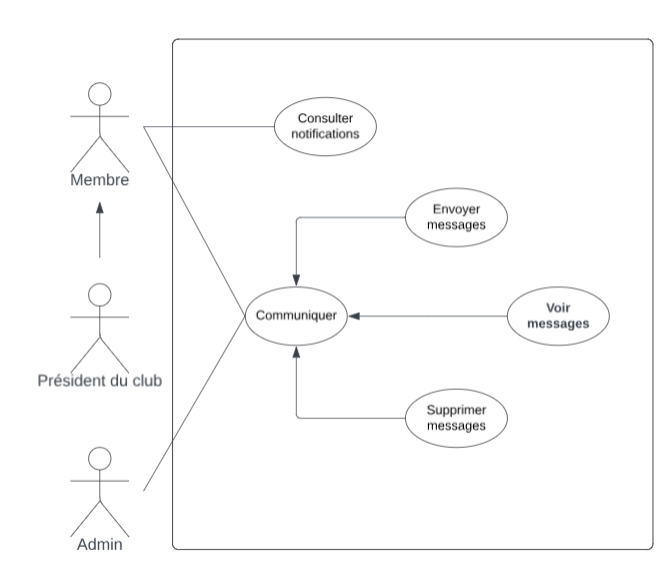
Le Backlog du troisième sprint présenté par le tableau 10 contient une liste des tâches identifiées qui devront être réalisées avant la fin du sprint.

| **Id** | **Thème** | **User story** | **Estimation (durée par heures)** |
| --- | --- | --- | --- |
| 1 | **Communiquer** | En tant que membre, président du club et admin je peux communiquer avec les autres acteurs | 6h |
| En tant que membre, président du club et admin je peux consulter mes notifications | 5h |

**Tableau10 : Backlog du produit sprint 4**

**2.1.2. Diagramme de cas d’utilisation du sprint 4 :**

La figure si dessous représente le diagramme de cas d’utilisation du sprint 4.

****

**Figure23 : Diagramme de cas d’utilisation du sprint4**

**2.2. Analyse de sprint 4 :**

Dans cette partie, nous présenterons l’analyse du domaine du Sprint 4 et l’analyse de la réalisation des cas d’utilisation.

**2.2.1. Analyse de la réalisation de sprint 4 :**

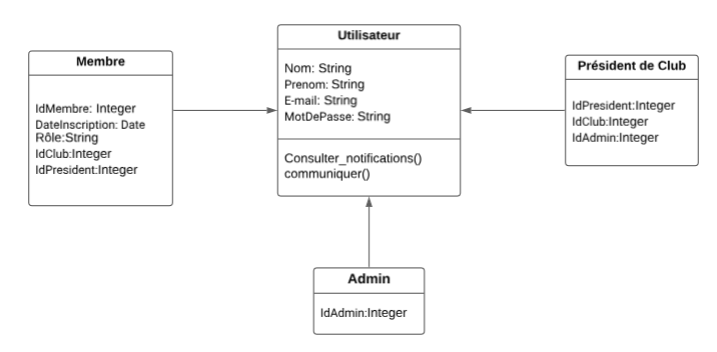
Dans cette partie, nous recensons pour chaque thème les différentes classes qui interviennent à sa réalisation dans le but de les faire intervenir dans les modélisations statique et dynamique. Le tableau 11 résume pour chaque thème du Sprint 4 les classes qui participent à sa réalisation.

| **Classe** | **Description** | **Attributs et types** |
| --- | --- | --- |
| **Utilisateur** | Elle encapsule les informations principales relatives aux utilisateurs de l’application | Nom: String  Prenom: String  E-mail: String  MotDePasse: String  Consulter\_notification()  Communiquer() |
| **Admin** | Elle hérite de la classe « Utilisateur » et contient les attributs relatifs d'étudiant. | IdEtudiant: Integer |
| **Membre** | Elle hérite de la classe « Utilisateur » et contient les attributs supplémentaires relatifs à membre. | IdMembre: Integer  DateInscription: Date  Rôle: String  IdClub: Integer  IdPresident: Integer |
| **Président de Club** | Elle hérite la classe « Utilisateur » et contient les attributs supplémentaires relatives à Président de Club. | IdPresident: Integer  IdClub: Integer  IdAdmin: Integer |

**Tableau11 : Analyse du domaine du sprint 4**

**2.2.2. Diagramme de classe d’analyse du Sprint 4 :**

La figure 24 représente diagramme de classe d’analyse du quatrième sprint.

****

**Figure24 :**

**2.3. Conception du Sprint 2 :**

**2.3.1. Vue statique du Sprint 4 :**

Dans cette partie, nous allons décrire la vue statique du Sprint 2 en utilisant le diagramme de classes, lequel sera conçu en suivant le modèle MVC.

**Diagramme de classe :**

La figure si dessous représente le diagramme de classe de sprint 4.

**Conclusion :**

Ce chapitre vient de présenter les deux dernières releases du projet où nous avons planifiés, détaillés.