Rapport de

Projet de Fin d’Année

Pour la 4ème Année en Génie Informatique

# 

**SCRUMIZE**

**Conception et la réalisation d’une plateforme qui gère les projets Scrum – Avec Angular et ASP CORE**

Réalisé par : AMMARI Abdelouhab

TAHRI Mostafa

AMARA Younes

Encadré par : Monsieur MANI Mohammed Adil

Membres du jury :

Monsieur MOUHIB Imad

Monsieur MANI Mohammed Adil

Année universitaire : 2021-2022

Dédicaces

Nous dédions ce modeste travail à :

En premier lieu ceux que personne ne peut compenser les sacrifices qu’ils ont consentis pour notre éducation et notre bien-être à nos parents qui se sont sacrifiés pour nous prendre en charge tout au long de notre formation et qui sont l’origine de notre réussite que dieu les garde et les protège.

A notre famille et nos chers amis qui nous ont accordé leur soutien dans les instants les plus difficiles.

Toute personne qui de près ou de loin a participé à notre formation.

***Remerciements***

Au terme de ce travail, nous tenons à remercier très chaleureusement notre encadrant Mr MANI Mohammed Adil, pour son encadrement, sa disponibilité, ses conseils avisés et son esprit critique durant toute la durée de notre travail.

Un grand merci au corps professoral et administratif de l'Ecole des Hautes Etudes d'Ingénierie d'Oujda présidé par Mr. BARBOUHCA Mohammed, pour la formation enrichissante qu'il nous a prodigué durant cette année.

Nous tenons également à exprimer l'honneur que font les membres du jury : Mr le professeur MOUHIB Imad et Mr le professeur MANI Mohammed Adil, d'avoir accepté de nous prêter leur attention et évaluer notre travail.

Pour tous ceux qui nous ont aidés, de près ou de loin, tous ceux que nous avons oublié avec la plume et non avec le cœur et pour lesquels nous réservons toute l'estime et la considération qu’ils trouvent ici l'expression de nos meilleurs sentiments.

***Résumé***

Notre projet consiste à créer une plateforme pour la gestion des projets agile, cette plateforme va aider les équipes à gérer leurs projets selon la méthode SCRUM appartenant à la famille agile, il permet aux équipes de collaborer, planifier, analyser et gérer les tâches quotidiennes, simplifier la planification des projets et garder les priorités au point.

Notre plateforme est composée de 2 parties,

* La première partie dédiée au Scrum Master pour ajouter les tâches du projet sous forme des user stories, attribuer chaque user story au développeur et au testeur, suivre l’état d’avancement du projet et être averti si la finalisation d’une Tâche est en retard.
* La 2ème partie dédiée aux développeurs qui peuvent accéder à l'application pour changer l'état de la tâche entre : TODO, DOING et DONE, consulter le tableau de bord des tâches pour voir le flux de travail du projet et la date estimée pour la livraison de la version courante.

***Abstract***

Our project consists of creating a platform for agile project management, this platform will help teams manage their projects using the SCRUM method which belongs to the agile family, it allows teams to collaborate, plan, analyze and manage daily tasks, simplify project planning and keep priorities in point.

Our platform is composed of 2 parts,

* The first part dedicated to the Scrum Master to add the tasks of the project in the form of user stories, assign each user story to the developer and the tester, follow the progress of the project and be notified if the finalization of a Task is late.
* The 2nd part dedicated to developers who can access the application to change the status of the task between: TODO, DOING and DONE, consult the task dashboard to see the project workflow and the estimated date for the delivery of the current version.

***Listes des Abréviations***

|  |  |
| --- | --- |
| Acronyme | Définition |
| EHEIO | Ecole des Hautes Etudes d'Ingénierie Oujda |
| CDI | Contrat de travail à durée indéterminée |
| UML | Unified Modeling Language |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheets |
| 2D | Deux dimensions |
| 3D | Trois dimensions |
| SPA | Single Page Application |
| EFC | Entity Framework Core |
| EF | Entity Framework |
| ASP | Active Server Page |
| ORM | Object–relational mapping |
| JSON | JavaScript Object Notation |
| PDF | Portable Document Format |
| MD | Markdown |

***Listes des Figures***

[Figure 1 : Etapes du Méthode en Cascade 5](#_Toc107194108)

[Figure 2 : Etapes des méthodes Agile 7](#_Toc107194109)

[Figure 3 Cas d'utilisation 10](#_Toc107194110)

[Figure 4 Diagramme de séquence : authentification 12](#_Toc107194111)

[Figure 5 Diagramme de séquence : ajouter projet 13](#_Toc107194112)

[Figure 6 Diagramme de séquence : Affectation des Taches par méthode de pocker 14](#_Toc107194113)

[Figure 7 Diagramme de séquence : Affectations de tâches détaillées 15](#_Toc107194114)

[Figure 8 Diagramme de séquence : affectations de tâches directes 16](#_Toc107194115)

[Figure 9 Diagramme de séquence : Suivi l’état d’avancement 16](#_Toc107194116)

[Figure 10 Diagramme de classe 18](#_Toc107194117)

[Figure 11 Plateforme : page d'accueil 25](#_Toc107194118)

[Figure 12 Plateforme : page d'inscription 26](#_Toc107194119)

[Figure 13 Plateforme : termes et conditions 26](#_Toc107194120)

[Figure 14 Plateforme : page de connexion 27](#_Toc107194121)

[Figure 15 Plateforme : tableau de bord 27](#_Toc107194122)

[Figure 16 Plateforme : page d'ajout d'un projet 28](#_Toc107194123)

[Figure 17 Plateforme : tableau de bord avec un projet ajouté 29](#_Toc107194124)

[Figure 18 Plateforme : Tableau de bord du projet Scrum 30](#_Toc107194125)

[Figure 19 Plateforme : ajouter un membre 31](#_Toc107194126)

[Figure 20 Plateforme : liste des membres 31](#_Toc107194127)

[Figure 21 Plateforme : Backlog 32](#_Toc107194128)

[Figure 22 Plateforme : les user stories 32](#_Toc107194129)

[Figure 23 Plateforme : ajouter une user story 33](#_Toc107194130)

[Figure 24 Plateforme : Liste des user stories 33](#_Toc107194131)

[Figure 25 Plateforme : les tâches 34](#_Toc107194132)

[Figure 26 Plateforme : ajouter une tache 34](#_Toc107194133)

[Figure 27 Plateforme : liste des tâches par user story 35](#_Toc107194134)

[Figure 28 Plateforme : ajouter un sprint 35](#_Toc107194135)

[Figure 29 Plateforme : liste des sprints 36](#_Toc107194136)

[Figure 30 Plateforme : ajouter une user story à sprint 36](#_Toc107194137)

[Figure 31 Plateforme : liste des user stories dans chaque sprint 37](#_Toc107194138)

[Figure 32 Plateforme : affectation des user stories 37](#_Toc107194139)

[Figure 33 Plateforme : tableau de bord de Scrum 38](#_Toc107194140)

[Figure 34 Plateforme : Burndown chart 39](#_Toc107194141)

***Sommaire :***

[Introduction 1](#_Toc107414536)

[I. Contexte du Projet : 2](#_Toc107414537)

[1. Présentation de l'EHEIO : 2](#_Toc107414538)

[1. Fiche technique : 2](#_Toc107414539)

[2. Les avantages de l'EHEIO : 2](#_Toc107414540)

[2. Analyse des Besoins 4](#_Toc107414541)

[1. Avant Agile : Waterfall Cycle en Cascade (Ou Cycla V) : 4](#_Toc107414542)

[2. La Méthode Scrum Agile : 6](#_Toc107414543)

[II. Analyse et Conceptions 9](#_Toc107414544)

[1. Diagramme de Cas d’Utilisation : 9](#_Toc107414545)

[a. Définition: 9](#_Toc107414546)

[b. Proposition: 10](#_Toc107414547)

[2. Diagramme de Séquence : 11](#_Toc107414548)

[a. Définition : 11](#_Toc107414549)

[b. Propositions : 12](#_Toc107414550)

[3. Diagramme de Classe : 17](#_Toc107414551)

[a. Définition : 17](#_Toc107414552)

[b. Proposition : 17](#_Toc107414553)

[III. Réalisation 19](#_Toc107414554)

[A. Gestion du Projet 19](#_Toc107414555)

[B. Outils, Frameworks, langages 20](#_Toc107414556)

[1. Lucidchart : 20](#_Toc107414557)

[2. HTML : 20](#_Toc107414558)

[3. CSS : 20](#_Toc107414559)

[4. JavaScript : 21](#_Toc107414560)

[5. Typescript : 21](#_Toc107414561)

[6. Bootstrap : 21](#_Toc107414562)

[7. Angular : 22](#_Toc107414563)

[8. Csharp : 22](#_Toc107414564)

[9. Asp.net Core : 22](#_Toc107414565)

[10. Entity Framework Core : 23](#_Toc107414566)

[11. JSON : 23](#_Toc107414567)

[12. Git : 23](#_Toc107414568)

[13. GitHub : 24](#_Toc107414569)

[14. Markdown : 24](#_Toc107414570)

[C. Captures d’écran 25](#_Toc107414571)

[Conclusion et perspectives 40](#_Toc107414572)

[Bibliographie & Webographie 41](#_Toc107414573)

# Introduction

Depuis les années 70, les méthodes de gestion des projets ont connu une évolution massive grâce à l’évolution des projets eux même qui ont devenus plus en plus compliqués, ce qu’il exige des méthodes très évoluées afin qu’on puisse gérer ces projets d’une manière plus efficace, et augmenter la qualité des produits. Pour toutes ces raisons des chercheurs ont inventés la méthode appelée Agile. Ce flux a commencé dans le secteur automobile et après été implémenté par tous les autres secteurs, que ce soit le secteur industriel, le secteur de gestion ou même le secteur de développement informatique.

L'objectif principal de ce projet est d'enrichir nos connaissances et notre savoir-faire, sur le développement des applications et plus précisément les application web.

Notre projet se repose sur « la création d'une plateforme web » qui consiste à gérer les projets agiles afin d’augmenter la qualité des produits développés en facilitant la gestion de chaque étape que ce soit pour les SCRUM Master qui s'engage à respecter les valeurs et les pratiques Scrum ou pour l’équipe de développement qui participe au projet.

Afin de réaliser notre Projet de Fin d’Année nous avons choisi ASP CORE en backend, avec Angular en frontend.

Notre rapport est composé de 3 grandes parties, nous commençons par le contexte de projet, puis nous passons vers la modélisation et nous finissons par la description des étapes de réalisation en détaillant les outils utilisés.

# Contexte du Projet :

## Présentation de l'EHEIO :

### Fiche technique :

* EHEI : école d'ingénierie privée, reconnu par le ministère d'enseignement supérieur de la recherche scientifique et de la formation des cades sous le numéro 344/2011
* Adresse : Rue de la liberté-Hay al Hikma-Oujda-Maroc
* Téléphone : 0536533076
* Fax : 0536533075
* Gmail : eheio.oujda@gmail.com

### Les avantages de l'EHEIO :

* ***Avantage 1 :***

Notre formation est base sur l'acquisition des compétences. C'est pourquoi, le corps professoral est composé, en plus de nos enseignants permanents, docteurs d'universitaires, de praticiens d'entreprise, d'experts et de consultants du secteur de l'industrie et de l'informatique.

* ***Avantage 2 :***

L'EHEI dispose d'un matériel pédagogique de haut niveau permettant à ses étudiants l'acquisition d'une formation pratique à haute valeur ajoutée.

* ***Avantage 3 :***

Les programmes de formation de l'EHEI sont conçus en relation avec les entreprises partenaires pour répondre au mieux aux besoins évolutifs du marché de l'emploi. Ils sont validés par le Conseil scientifique de l'établissement et accrédités par le Ministère de l'Enseignement Supérieur, de la Recherche Scientifique et de la Formation des Cadres.

* ***Avantage 4 :***

Près de 70% des étudiants, soit plus de 120 diplômés de l'EHEI depuis sa création en 2011, ont trouvé un emploi sans délai à l'issue de leur formation. En moyenne, la durée de recherche n'excède pas 3 mois. Les diplômés sont pour la plupart en CDI, à des postes d'encadrement, le statut cadre leur étant donné soit à l'embauche soit après une période d'essai ou de stage de pré-embauche.

* ***Avantage 5 :***

L'EHEI met tout en œuvre pour assurer la réussite de ses étudiants via un accompagnement permanent. Dès la recherche de l'entreprise d'accueil, I'EHEI fait bénéficier les étudiants d'un ensemble de services : entretien d'évaluation professionnelle, accompagnement personnalisé….

* ***Avantage 6 :***

Intégrer I'EHEI, 'est choisir une école accréditée par l'Etat, multidisciplinaire spécialisée dans les technologies de pointe. Les filières sont axées sur les stratégies et techniques de communication. L’ouverture vers l'international. Les étudiants sont bien encadrés et aidés dans leur choix d'avenir.

## Analyse des Besoins

Notre projet de Fin d’Année consiste à créer une Plateforme pour la gestion des projets Agile, l’objectif de cette plateforme est de faciliter la gestion d’un projet en utilisant la Méthode Agile : Scrum, elle va donner la possibilité de suivre chaque étape de la réalisation plus facilement pour toutes les participants et les contributeurs au projet.

### 1. Avant Agile : Waterfall Cycle en Cascade (Ou Cycla V) :

La méthode de **gestion de projet waterfall**, ou « en cascade », ou encore « cycle en V », consiste à passer par une liste d’étapes prédéfinies de façon séquentielle. L’avantage de cette méthode est que l’intégralité du projet est définie et planifiée dès le départ, et par conséquent, le budget global est connu.

L’analyse des besoins du client est naturellement toujours la première étape de la méthode de gestion waterfall. Viennent ensuite la rédaction des spécifications fonctionnelles, la construction du budget et la planification du développement. Le processus de développement peut alors débuter avec la création d’une première fonctionnalité, sa validation et/ou son éventuelle correction, puis sa maintenance. Cet enchaînement d’étapes recommence alors séquentiellement, jusqu’à ce que toutes les fonctionnalités demandées par le client aient été réalisées.

Si à l’issue de la dernière livraison, des mises à jour ou des modifications sont nécessaires, le même cycle sera répété.

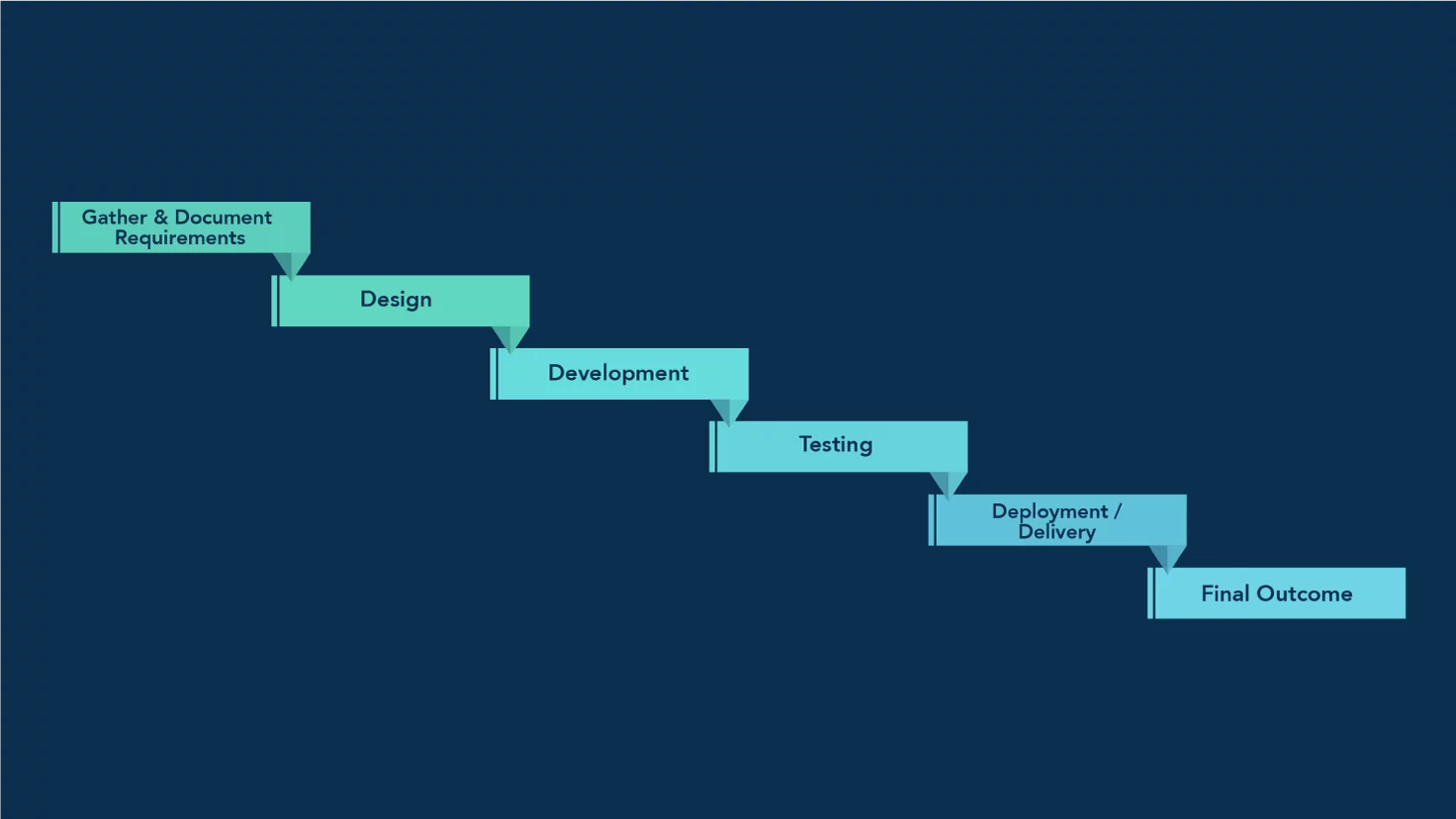


Figure 1 : Etapes du Méthode en Cascade

***Les Problèmes du Projet Waterfall ou Cycle en Cascade :***

Pour commencer, cette méthode implique une analyse complète des besoins avec le client. Si pour des petits projets, cela peut être efficace, dès que la complexité augmente, ce qui arrive vite de nos jours, cette phase peut prendre des semaines, voire des mois. Si l’analyse des besoins prend des mois, et que la livraison du produit fini également, le risque de se retrouver face à un besoin qui a évolué entre temps est grand. L’équipe va donc se retrouver face à des demandes de modifications ou de création de fonctionnalités qui n’étaient pas prévues à l’origine, et [rencontrer des difficultés](https://www.anyideas.net/2012/12/comment-planter-premier-projet/) à les prendre en compte parce qu’elles n’auront pas été planifiées.

L’équipe ne dispose donc d’aucune souplesse dans la réalisation du projet, puisqu’avec un déroulement séquentiel, la réalisation d’une fonctionnalité dépend des réalisations précédentes. Il est donc particulièrement complexe de prendre en compte toute demande de changement, car elle aura un impact sur les développements courants et futurs, et nécessitera de revenir sur des fonctionnalités déjà en place, avec tous les effets négatifs que cela peut engendrer.

### 2. La Méthode Scrum Agile :

Si la méthode de gestion de projet waterfall nous paraît si logique est naturelle, c’est que nous l’utilisons tous les jours. Elle nous permet de planifier nos congés, des voyages ou encore des loisirs, de construire un fichier Excel correspond à un besoin…

En revanche, dans la gestion d’un projet, elle devient rapidement inadaptée. Avec une méthode agile, l’approche va être différente. Comme nous l’avons vu, l’inconvénient d’une planification complète comme celle faite dans le cadre d’une gestion de projet waterfall, c’est qu’elle ne laisse aucune place à l’imprévu. Si la demande du client est modifiée, ou que les conditions de réalisation changent (départ d’une personne de l’équipe projet par exemple), toute l’organisation doit être revue et risque d’engendrer des dépassements budgétaires et des retards de livraison.

Une démarche agile comme la méthodologie Scrum va permettre d’apporter de la souplesse par rapport à une **gestion de projet waterfall**. Le projet va lui-même être découpé en étapes (pouvant chacune représenter un petit projet) aboutissant à la réalisation d’une fonctionnalité. L’objectif fixé, au lieu d’être à long terme, va au contraire devoir être réalisé dans un délai beaucoup plus court. La suite du projet sera ainsi planifiée en fonction de la réalisation précédente.

A chaque étape, il y a une livraison testée. Si le test est concluant, nous passons à la fonctionnalité suivante, sinon, nous corrigeons. De cette façon, les développements déjà effectués seront validés à chaque étape et les nouveaux objectifs fixés en fonction des demandes et des ressources disponibles. L’approche est donc beaucoup plus souple puisqu’il est beaucoup plus facile de d’adapter aux changements, qui s’agisse des demandes du client ou de l’équipe qui évoluent.

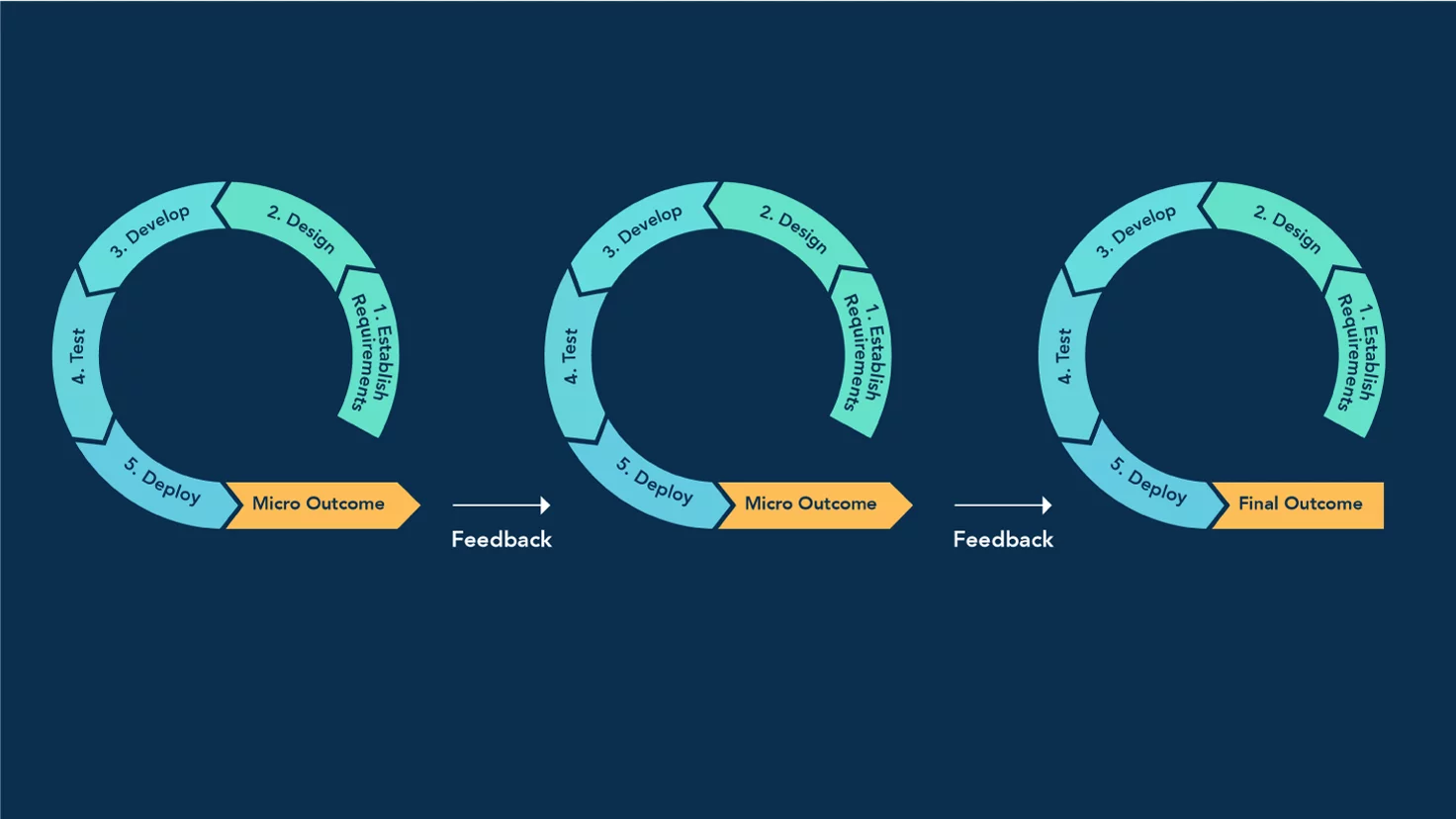


Figure 2 : Etapes des méthodes Agile

***Les Avantages des Méthodes Agile :***

Alors, qu'est-ce qu'Agile a à offrir par rapport aux méthodologies de développement de logiciels plus traditionnelles comme Waterfall ? Il doit y avoir une raison impérieuse d'utiliser la méthodologie Agile au-delà de la notion qu'elle est la « dernière et la meilleure ».

* Agile offre de la flexibilité. L'un des éléments fondamentaux de l'agilité est que la méthodologie offre une approche flexible du développement logiciel. Les priorités et les exigences peuvent facilement être ajustées tout au long du projet pour répondre aux besoins des parties prenantes.
* Agile responsabilise l'équipe. L'équipe interfonctionnelle travaille comme une unité pour définir, concevoir et construire le produit logiciel. L'équipe doit être autoorganisée et n'est pas dirigée par un manager. Cela permet aux membres de l'équipe de définir et de livrer leur propre travail comme ils l'entendent. L'équipe se voit confier la responsabilité de livrer le projet, ce qui la responsabilise.
* Le délai de mise sur le marché est accéléré. Avec les projets Agile, l'accent est davantage mis sur ce qui doit être fait et moins sur la planification et la documentation. L'énergie de l'équipe est consacrée au développement du produit logiciel et à la livraison d'un logiciel fonctionnel à chaque itération ou sprint.
* L'apprentissage est encouragé et adopté. L'apprentissage fait partie du processus - le produit est défini au fur et à mesure que l'équipe itère. Cela permet à l'équipe d'apprendre, d'ajuster le cours et de s'améliorer tout au long du projet. Les rétrospectives de sprint, un processus fondamental pour Agile, sont utilisées pour recueillir les commentaires de l'équipe sur la façon dont ils peuvent s'améliorer pour fournir des logiciels plus rapidement et avec une meilleure qualité.
* Il existe plus de possibilités de créativité. Agile fonctionne très bien lorsque la vision ou les fonctionnalités du produit ne sont pas bien définies. Agile permet aux propriétaires de produits d'ajuster les exigences et les priorités en cours de route pour tirer parti des opportunités et, en fin de compte, fournir un meilleur produit à toutes les parties prenantes du projet.

***Implémentation du Méthode Agile :***

Il y’a plusieurs solutions qui implémente la méthode Agile, La liste des méthodologies agiles ci-dessous comprend des types célèbres de méthodologie agile parmi lesquels nous pouvons choisir :

1. Kanban.
2. Scrum.
3. Extreme Programming (XP)
4. Crystal.
5. Dynamic Systems Development Method (DSDM).

Nous avons remarqué dans la phase d’étude du marché que toutes les bonnes solutions informatiques pour la gestion d’un projet Agile sont payantes, c’est pour cela nous avons décidé de créer une solution gratuite qui adopte la méthode Scrum pour la gestion du projet.

Pour le coté financière de notre projet, nous allons mettre des bannières publicitaires dans différents endroit sur la plateforme qui seront remplies par des annonces des entreprises ou des produits convenables.

# Analyse et Conceptions

## Diagramme de Cas d’Utilisation :

### a. Définition :

Un **cas d'utilisation** (en anglais *use case*) permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. Le format de représentation d'un cas d'utilisation est complètement libre mais UML propose un formalisme et des concepts issus de bonnes pratiques.

Le diagramme de cas d'utilisation permet de représenter visuellement une séquence d'actions réalisées par un système, représenté par une boîte rectangulaire, produisant un résultat sur un acteur, appelé *acteur principal*, et ceci indépendamment de son fonctionnement interne.

### b. Proposition :

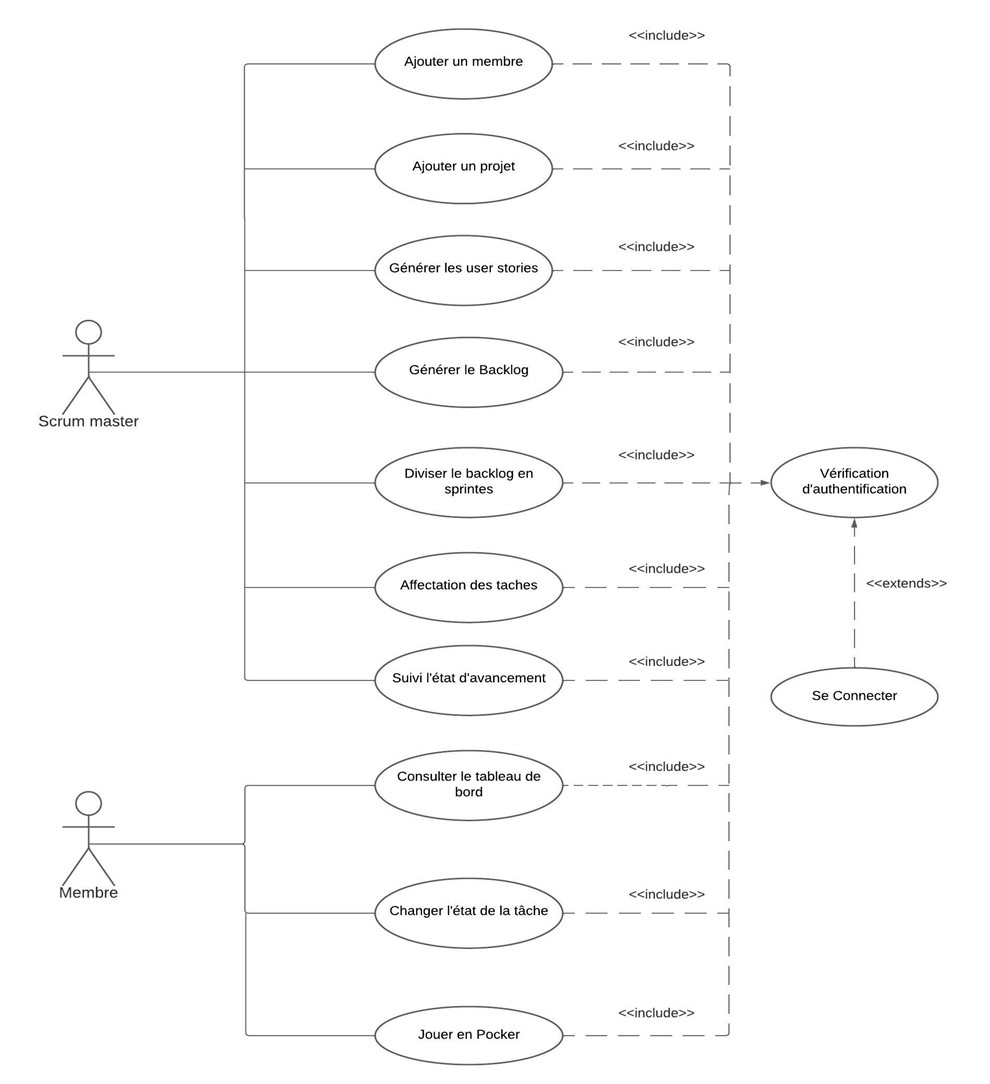


Figure 3 Cas d'utilisation

## Diagramme de Séquence :

1. Définition :

Un diagramme de séquence est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont. Les diagrammes de séquence sont organisés en fonction du temps.

Le temps s'écoule au fur et à mesure que vous parcourez le diagramme de haut en bas. Les objets impliqués dans l'opération sont répertoriés de gauche à droite en fonction du moment où ils prennent part dans la séquence de messages.

Un diagramme de séquence est un type de diagramme d'interaction, car il décrit comment et dans quel ordre plusieurs objets fonctionnent ensemble. C’est un algorithme séquentiel permettant de détecter les boucles, les conditions et plusieurs autres types de blocs permettant de modéliser la logique d’un cas d’utilisation.

Les avantages de l’utilisation des diagrammes de séquence dans notre projet sont :

* Représenter les détails d'un cas d'utilisation UML
* Modéliser le déroulement logique d'une procédure, fonction ou opération complexe
* Voir comment les objets et les composants interagissent entre eux pour effectuer un processus.
* Schématiser et comprendre le fonctionnement détaillé d'un scénario existant ou à venir

1. Propositions :

* **Authentification :**

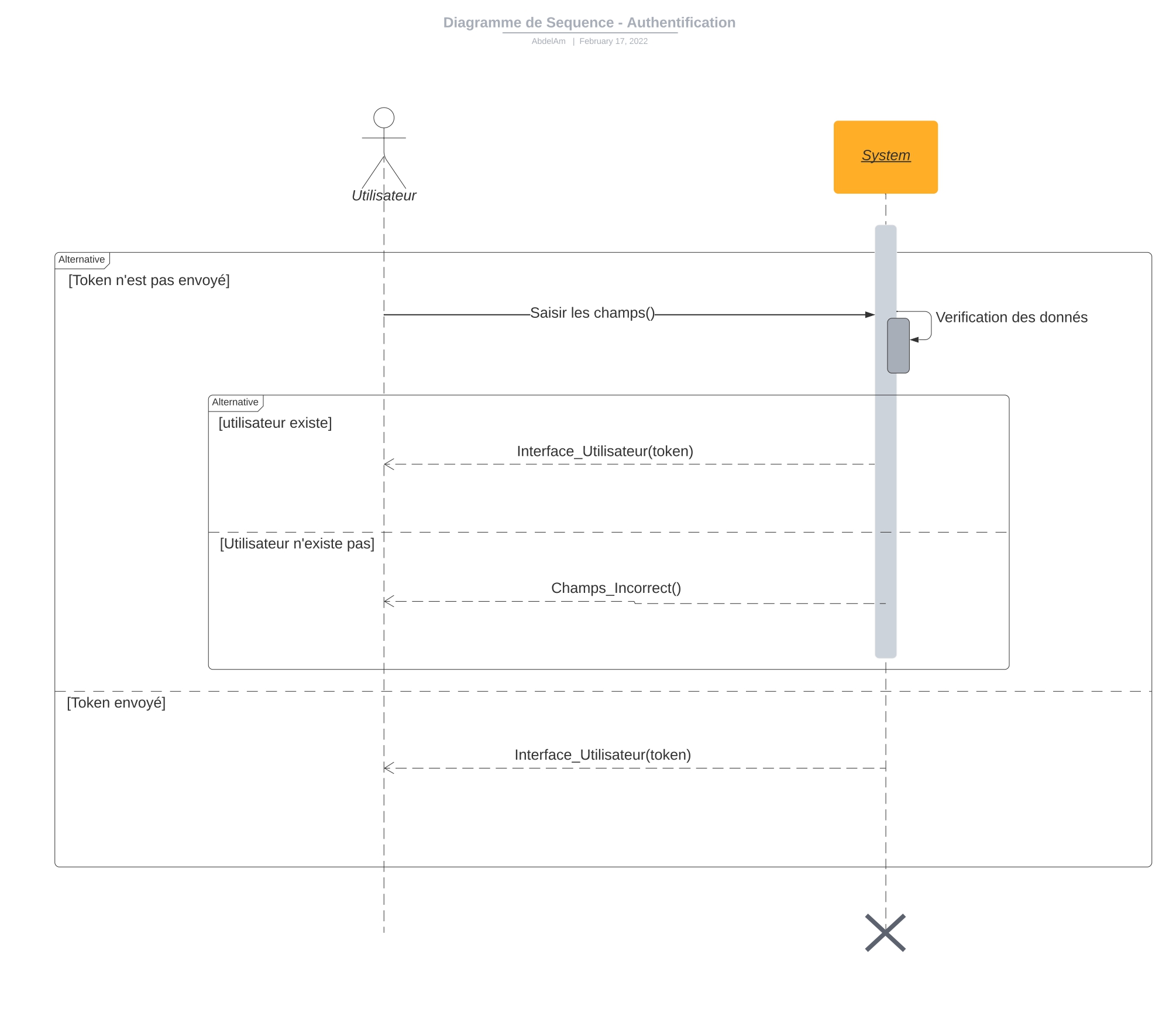


Figure 4 Diagramme de séquence : authentification

* **Ajouter Projet :**

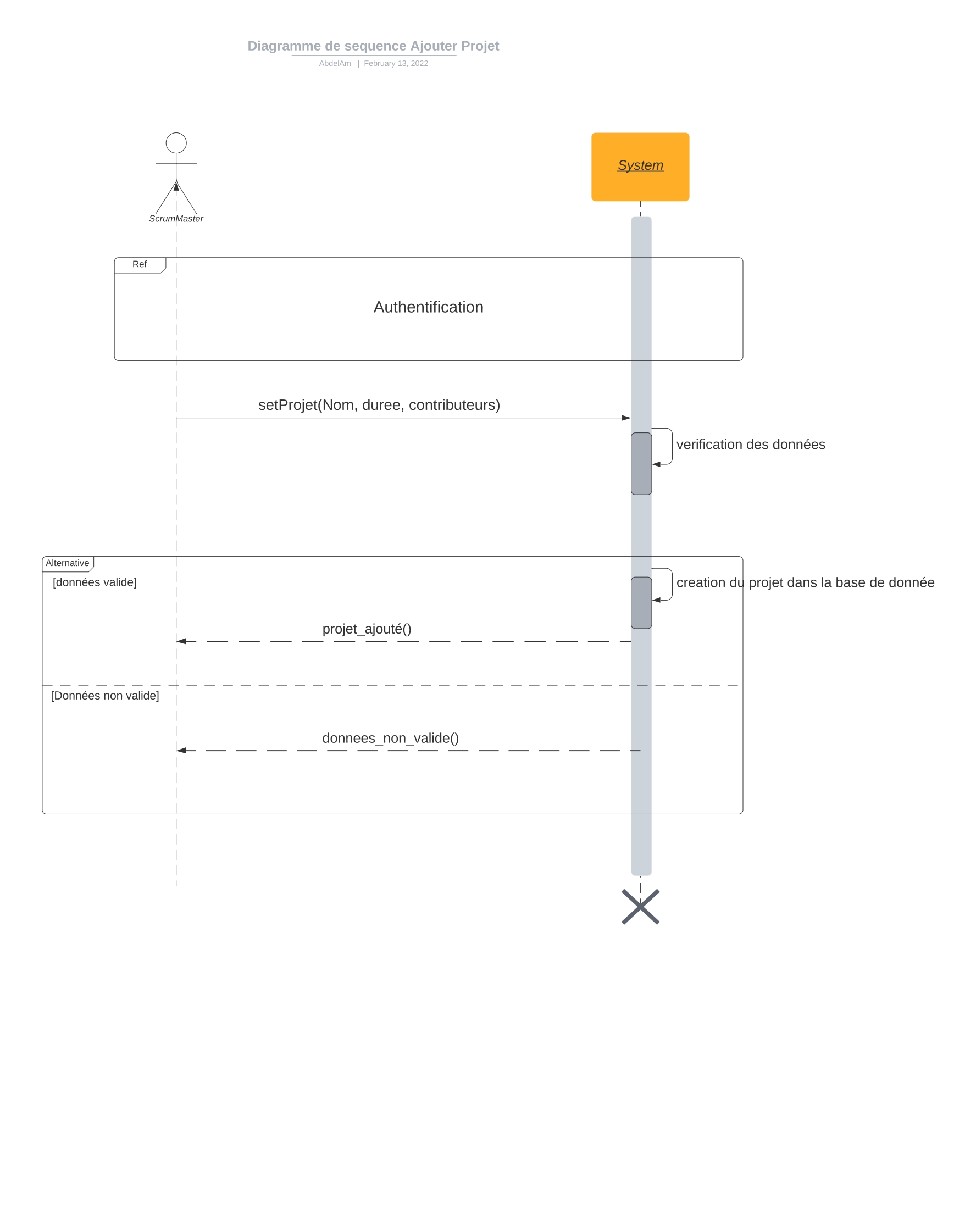


Figure 5 Diagramme de séquence : ajouter projet

* **Affectation des Taches :**

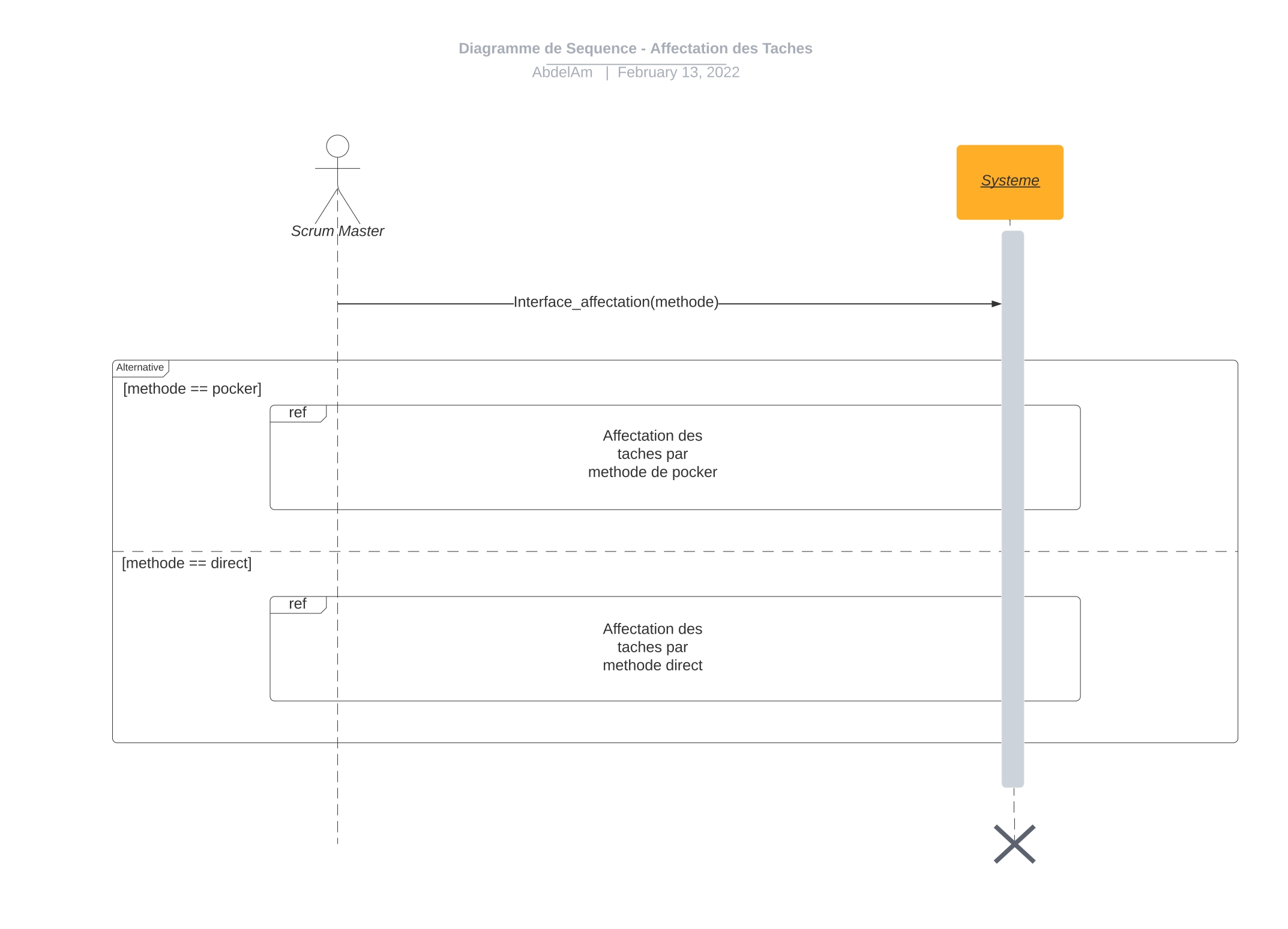
****

Figure 6 Diagramme de séquence : Affectation des Taches par méthode de pocker

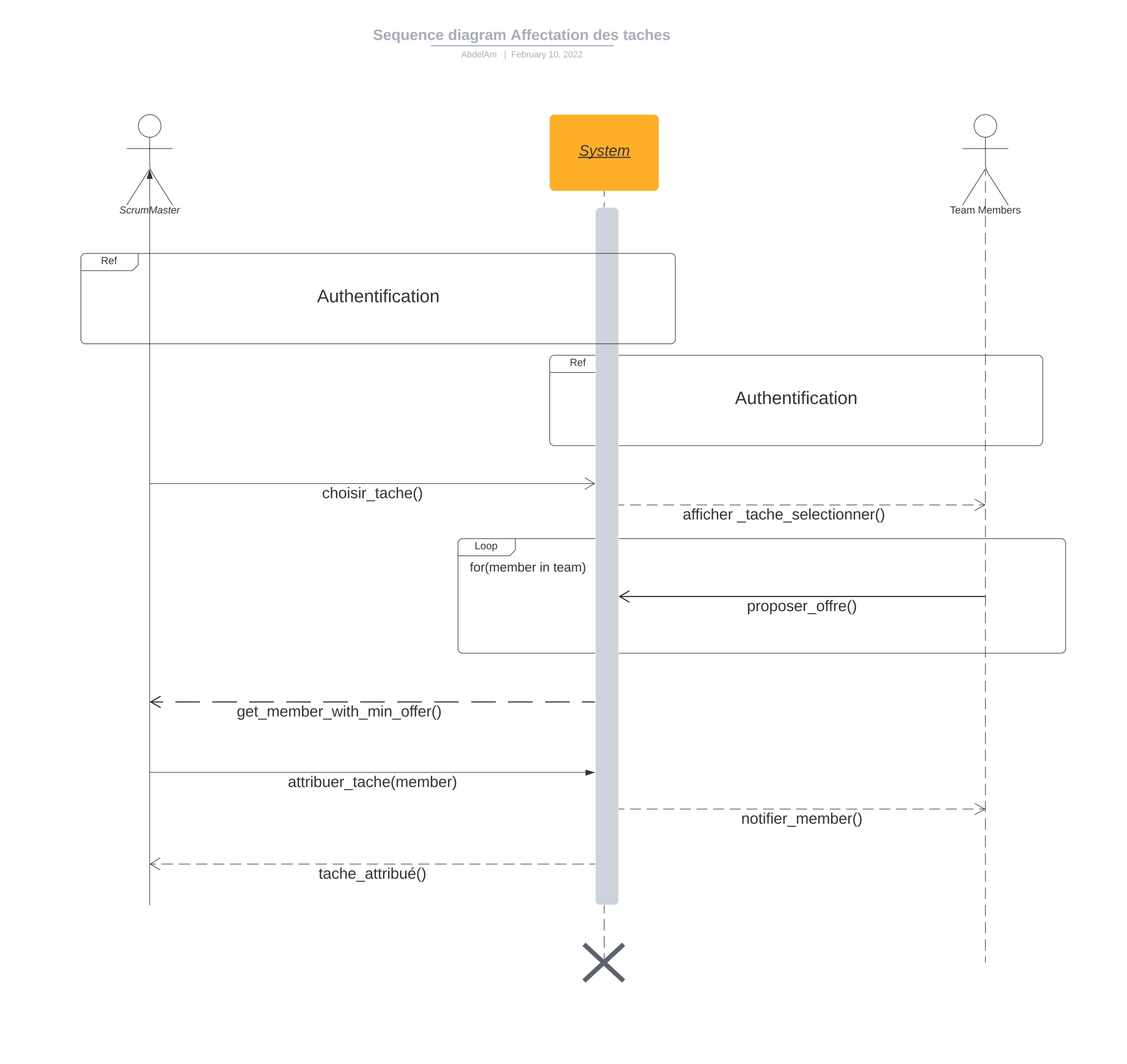
****

Figure 7 Diagramme de séquence : Affectations de tâches détaillées

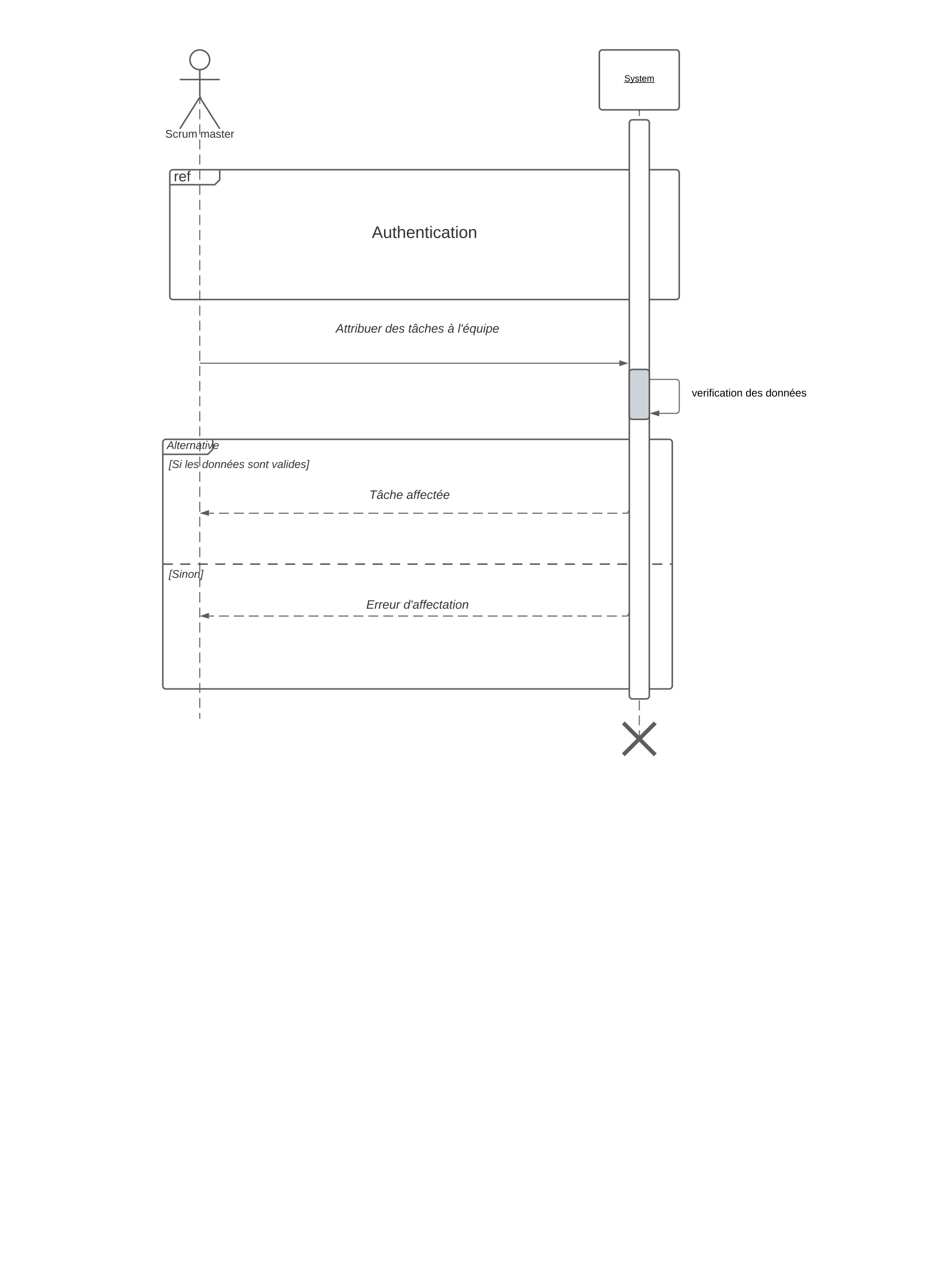
****

Figure 8 Diagramme de séquence : affectations de tâches directes

* **Suivi l’état d’avancement :**

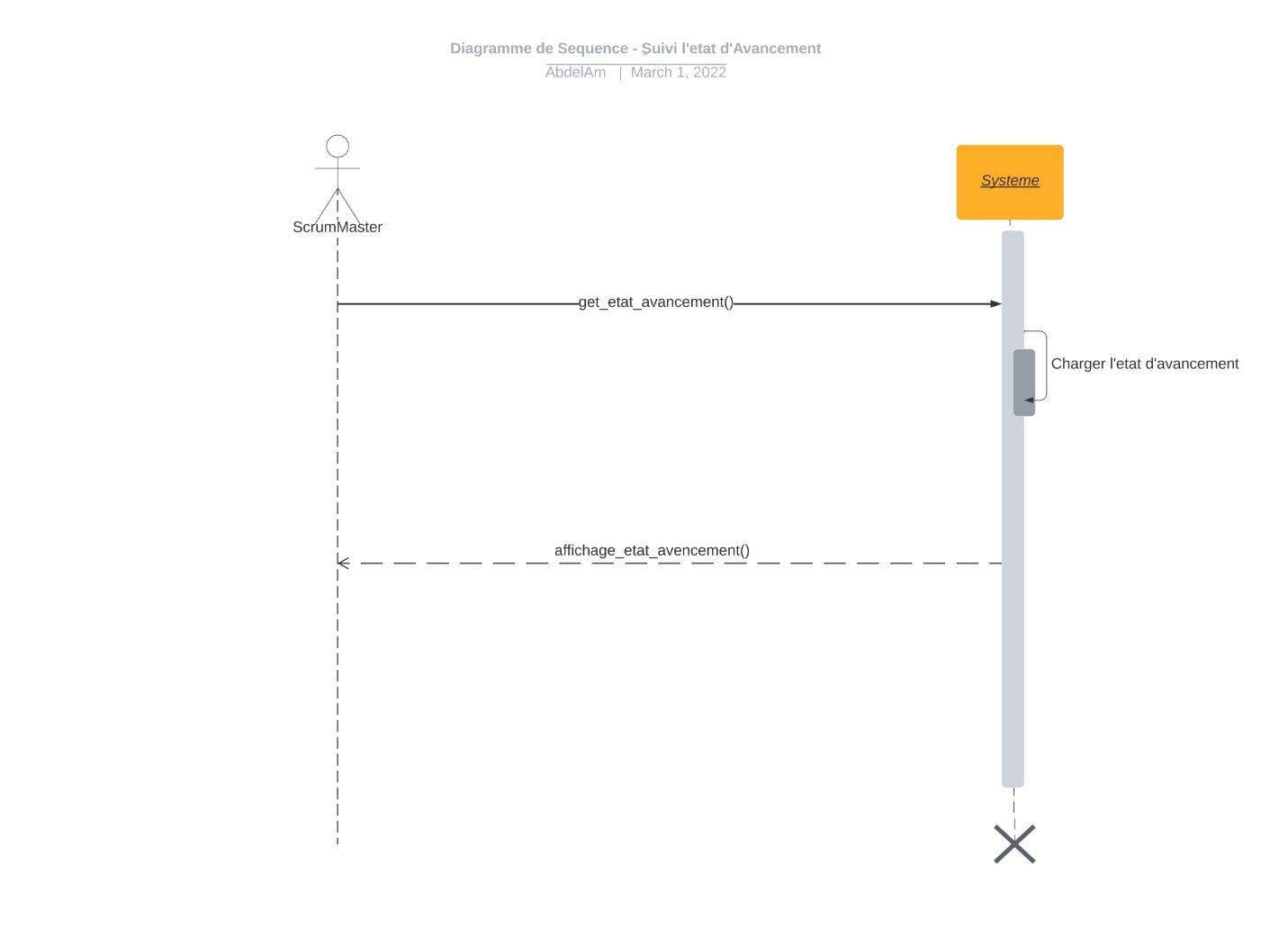
****

Figure 9 Diagramme de séquence : Suivi l’état d’avancement

## Diagramme de Classe :

1. Définition :

Un diagramme de classes fournit une vue globale d'un système en présentant ses classes, interfaces et collaborations, et les relations entre elles.

Les diagrammes de classes sont statiques : Ils affichent ce qui interagit mais pas ce qui se passe pendant l'interaction.

En notation UML, une classe est représentée sous la forme d'un rectangle divisé en plusieurs parties : le nom de la classe, les attributs (champs), les opérations (méthodes) et autres.

1. Proposition :

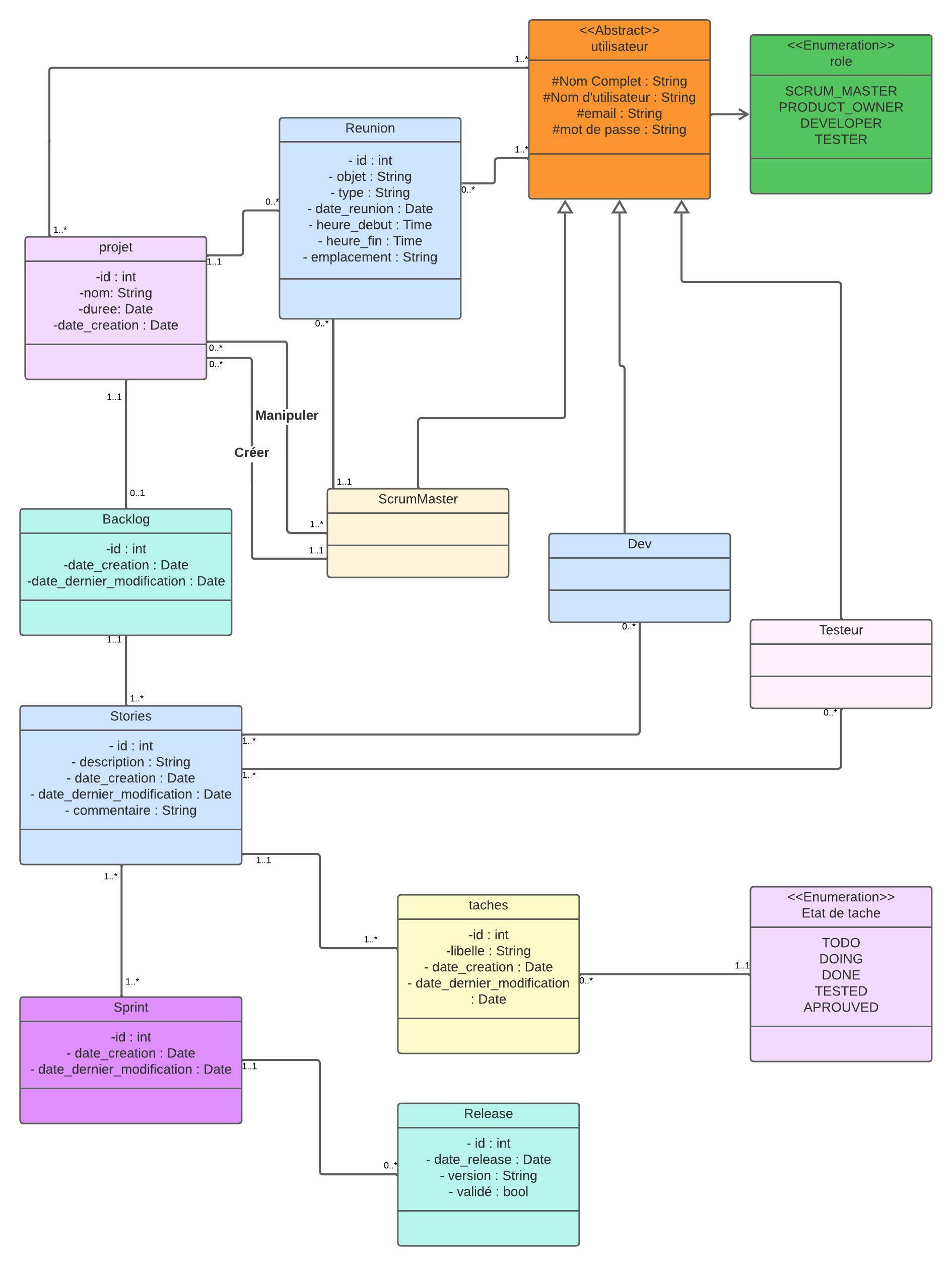


Figure 10 Diagramme de classe

# Réalisation

## A. Gestion du Projet

Nous avons adopté Scrum comme cadre de travail pour gérer notre projet. Le projet commence par une vision claire, et un ensemble de fonctionnalités produit par ordre d'importance fournie et entretenue par monsieur Mani qui est à la fois notre Product Owner et Scrum Master. Ces fonctionnalités font partie du Product Backlog.

Nous avons spécifié une durée appelée itération ou sprint, c'est le temps dont nous disposons pour terminer les fonctionnalités sélectionnées. Les sprints durent généralement d’une à quatre semaines, et cette durée est maintenue tout au long de la vie du projet afin d'établir une cadence.

Nous devons sélectionner les éléments du Backlog de produit que nous pensons pouvoir compléter dans le sprint, et nous créons un Sprint Backlog composé des fonctionnalités et des tâches dans le cadre de la réunion de planification de sprint.

Une fois que l'équipe s'est engagée dans un Sprint Backlog, le travail de tâche commence. Pendant ce temps du sprint, l'équipe est protégée contre les interruptions et autorisée à se concentrer sur l'atteinte de l'objectif du sprint. Aucune modification du Backlog de sprint n'est autorisée. Cependant le Product Backlog peut être modifié en vue du prochain sprint.

Pendant le sprint, nous nous connectons quotidiennement les uns aux autres sous la forme d'une réunion de 15 minutes connue sous le nom daily scrum. On se tient en cercle dans l'école pendant la pause et chacun raconte ce qu'il a fait hier, sur quoi il va travailler aujourd'hui et les difficultés qu'il a rencontré pendant la réalisation.

À la fin du sprint, nous démontrons le travail que nous avons effectué pour les parties prenantes (Monsieur Mani) et recueillir des commentaires qui affecteront ce sur quoi nous travaillerons dans le prochain sprint. Nous organisons également une rétrospective pour apprendre à nous améliorer.

Cette réunion est essentielle, car elle se concentre sur les trois piliers de Scrum : transparence, contrôle et adaptation. Ce cycle de vie se répète jusqu'à ce que nous terminions notre projet, à la fin nous avons fini avec presque cinq sprints

## B. Outils, Frameworks, langages



### Lucidchart :

Lucidchart est une plate-forme web qui permet aux utilisateurs de collaborer sur le dessin, la révision et le partage de graphiques et de diagrammes. Nous avons utilisé cette application pour modéliser nos différents diagrammes UML.



### HTML :

HTML, ou HyperText Markup Language, nous permet de créer et de structurer des sections, des paragraphes et des liens à l’aide d’éléments, de balises et d’attributs.



### CSS :

CSS signifie Cascading Style Sheets. Il décrit comment les objets HTML sont affichés sur le navigateur. Nous l’avons utilisé pour rendre nos pages Web plus magnifiques et dynamiques via des animations.



### JavaScript :

JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web. À chaque fois qu'une page web fait plus que simplement afficher du contenu statique — afficher du contenu mis à jour à des temps déterminés, des cartes interactives, des animations 2D/3D, des menus défilants, ou autre, JavaScript a de bonnes chances d'être impliqué. C'est la troisième couche des technologies standards du web.



### Typescript :

TypeScript est un langage de programmation développé et maintenu par Microsoft. Il s’agit d’un sur-ensemble syntaxique strict de JavaScript et ajoute un typage statique facultatif au langage. TypeScript est conçu pour le développement d’applications volumineuses et de transcompilations en JavaScript. Nous l’avons utilisé avec Angular au lieu de javascript.



### Bootstrap :

Bootstrap est un frameworm frontal gratuit et open-source pour la conception de sites Web et d’applications Web. Il contient des modèles de conception basés sur HTML et CSS pour tout, de la typographie, des formulaires, des boutons, de la navigation et d’autres composants d’interface, ainsi que des extensions JavaScript.



### Angular :

Angular est un framework TypeScript très populaire actuellement, visant à structurer et simplifier le développement d’applications mono-pages (Single Page Application – SPA). Il est développé par Google sous licence open-source, il offre les outils nécessaires pour développer du code bien organisé, modulaire, basé sur l’injection de dépendances et prêt pour les tests unitaires.



### Csharp :

C# est un langage de programmation moderne, orienté objet et de type sécurisé. C# permet aux développeurs de créer de nombreux types d'applications sécurisées et robustes qui s'exécutent dans .NET. C# a ses racines dans la famille de langages C et sera immédiatement familier aux programmeurs C, C++, Java et JavaScript.

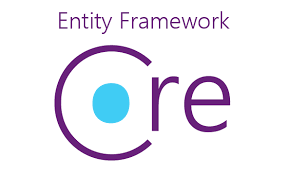
Nous avons utilisé le langage Csharp avec Asp.net Core dans la partie backend.



### Asp.net Core :

ASP.NET est un framework de développement Web populaire pour la création d’applications Web sur la plate-forme .NET.

ASP.NET Core est la version open source de ASP.NET, qui fonctionne sur MacOs, Linux et Windows. ASP.NET Core a été publié pour la première fois en 2016 et est une refonte des versions antérieures de Windows uniquement de ASP.NET



### Entity Framework Core :

Entity Framework (EF) Core est une version légère, extensible, open source et multiplateforme de la très connue technologie d’accès aux données Entity Framework.

EF Core peut servir de Mappeur relationnel objet (O/RM), qui permet aux développeurs .NET de travailler avec une base de données à l’aide d’objets .NET.



### JSON :

C’est un format de données qui est léger, facile à lire et écrire pour les humains et il est aisément analysable par les machines. Il est basé sur le langage JavaScript.

Nous avons choisi ce format pour traiter la connexion aux bases de données, ainsi que pour manipuler le transfert de données dans la coté mobile de l’application.



### Git :

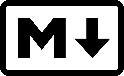
Git est un système de contrôle de version distribué gratuit et open source. Le but de Git est de garder une trace des projets et des fichiers à mesure qu'ils changent au fil du temps avec des manipulations effectuées par différents utilisateurs. Git stocke des informations sur l'avancement du projet dans un référentiel. Un référentiel a des *commits* sur le projet ou un ensemble de références aux *commits*.

Toutes ces informations sont stockées dans le même dossier que le projet dans un sous-dossier appelé « .git » et seront généralement masquées par défaut dans la plupart des systèmes.



### GitHub :

GitHub est une plateforme collaborative gratuite mais qui propose également des plans payants. C’est une interface web créée pour faciliter l’interaction avec Git. Utilisé principalement comme outil de collaboration, il offre la possibilité de travailler à plusieurs sur le même projet, chaque développeur peut ainsi apporter sa pierre à l’édifice. Cet outil facilite la communication entre les développeurs et à la façon d’un réseau social, permet de partager son code, ses conseils et d’aider la communauté.



### Markdown :

Markdown est donc un langage informatique, plus précisément un langage de balisage qui a pour but d’offrir une syntaxe facile à lire et à écrire. Ainsi, votre document Markdown (.md ou .markdown) reste aisé à lire pour tout le monde et peut-être facilement converti dans d’autres formats de fichiers, en particulier en HTML structuré mais il existe aussi des possibilités d’export vers un PDF ou vers votre logiciel de traitement de texte préféré. Markdown a donc un réel intérêt pour quiconque publie sur Internet (via blog, un site, des commentaires ou même à travers la rédaction d’un mail) mais depuis quelques années, il suscite également l’intérêt d’autres utilisateurs, en particulier des personnes qui écrivent.

## C. Captures d’écran

En accédant à la plate-forme Scrumize, la première page que vous allez consulter est la page d’accueil qui a un design de style illustraion.

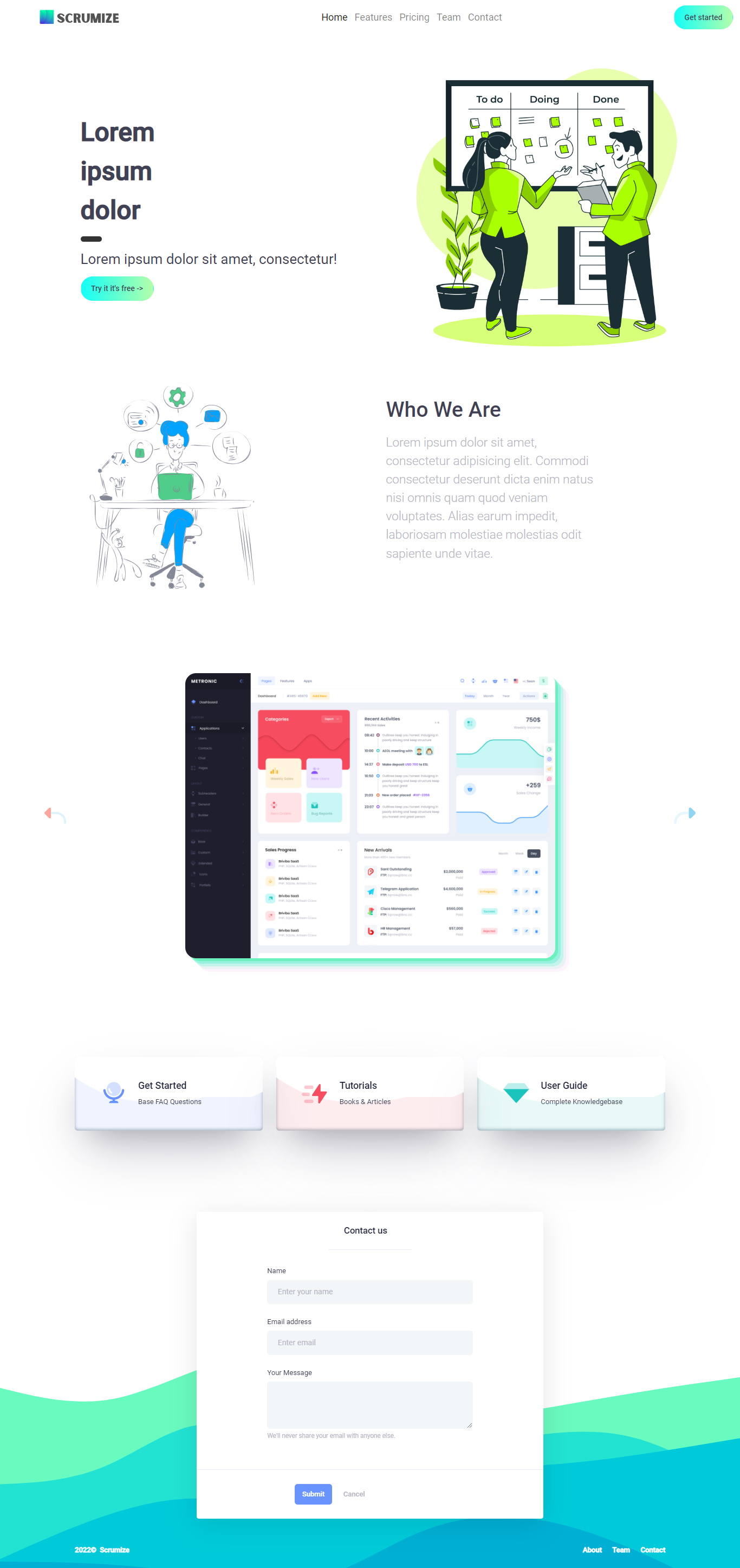


Figure 11 Plateforme : page d'accueil

Une fois que l’utilisateur décide d’utiliser la solution en cliquant sur le bouton « Get started », il sera redirigé vers la page d’inscription.

Pour vous inscrire, vous avez dû insérer une photo personnelle, un nom d’utilisateur, votre adresse e-mail, un mot de passe correct et sélectionner un type de compte dans la liste qui comporte quatre types un Scrum master, un développeur, un testeur et un Product owner.

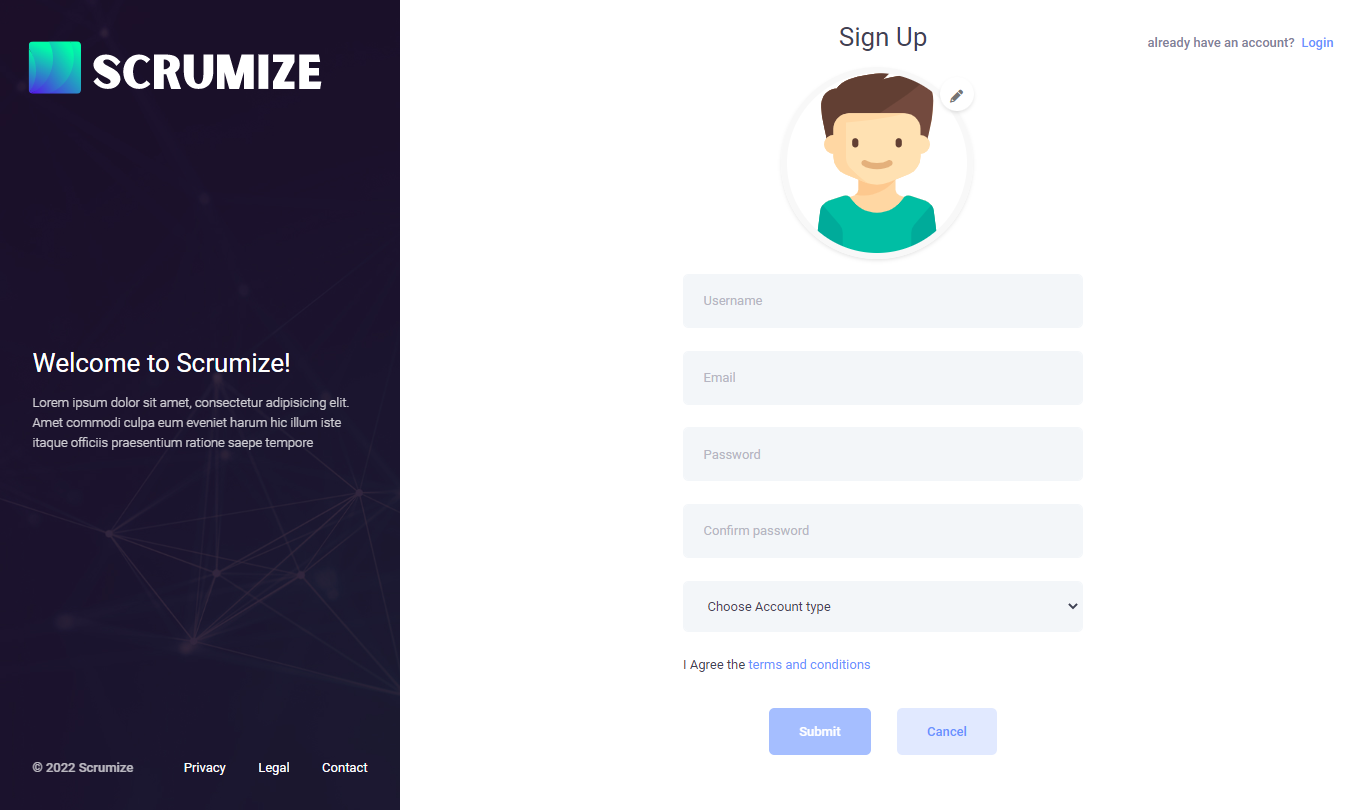


Figure 12 Plateforme : page d'inscription

L’inscription ne sera pas complète sans l’acceptation des termes et conditions d’utilisation.

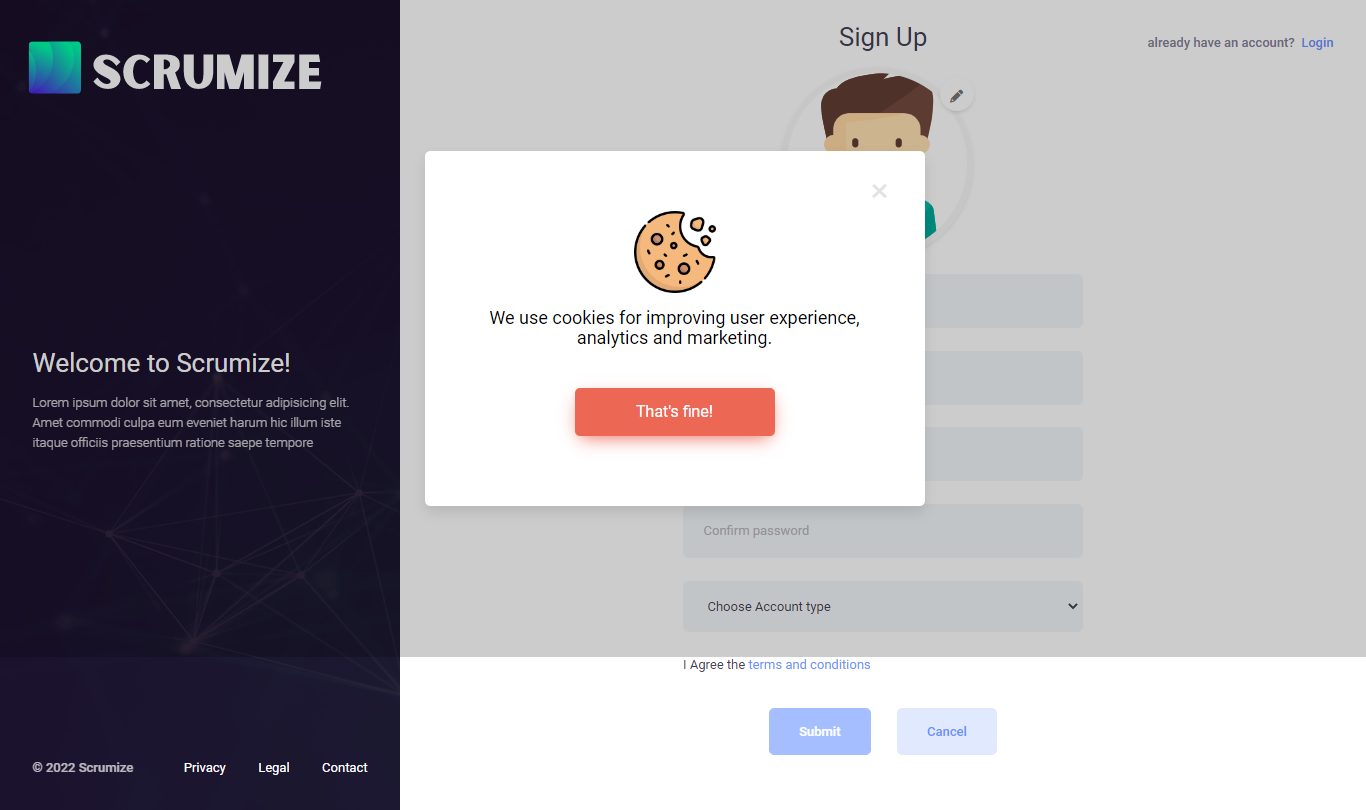


Figure 13 Plateforme : termes et conditions

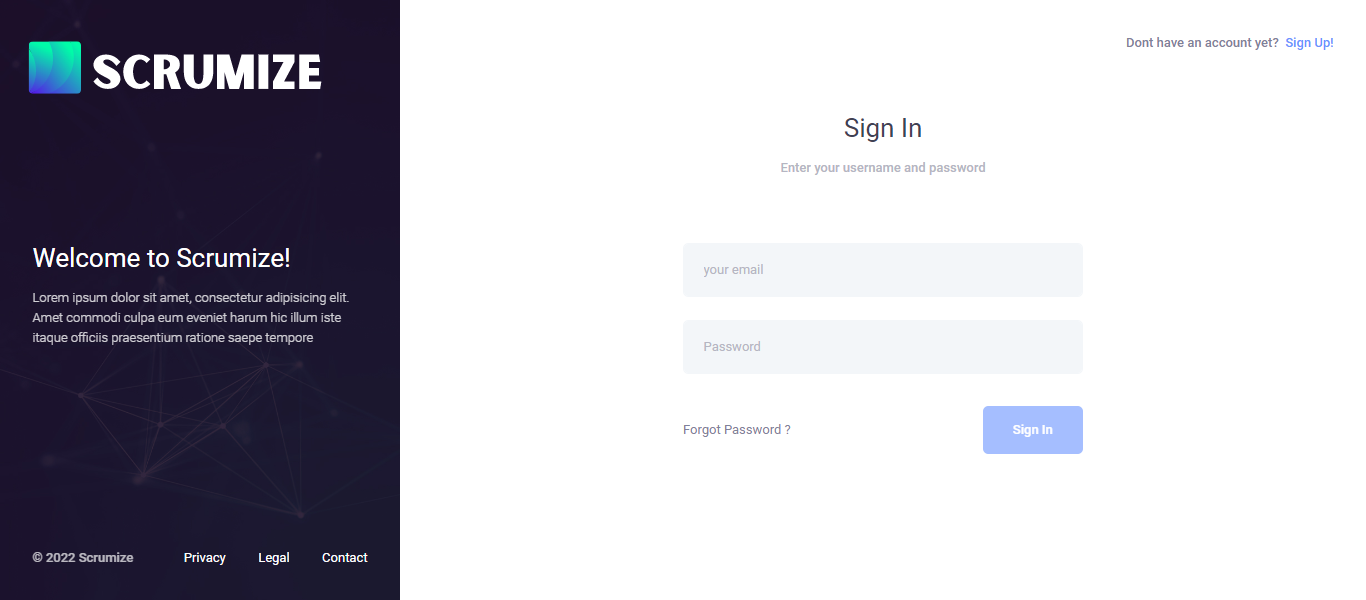


Figure 14 Plateforme : page de connexion

Une fois que l’utilisateur est inscrit, il doit se connecter pour utiliser la plate-forme.

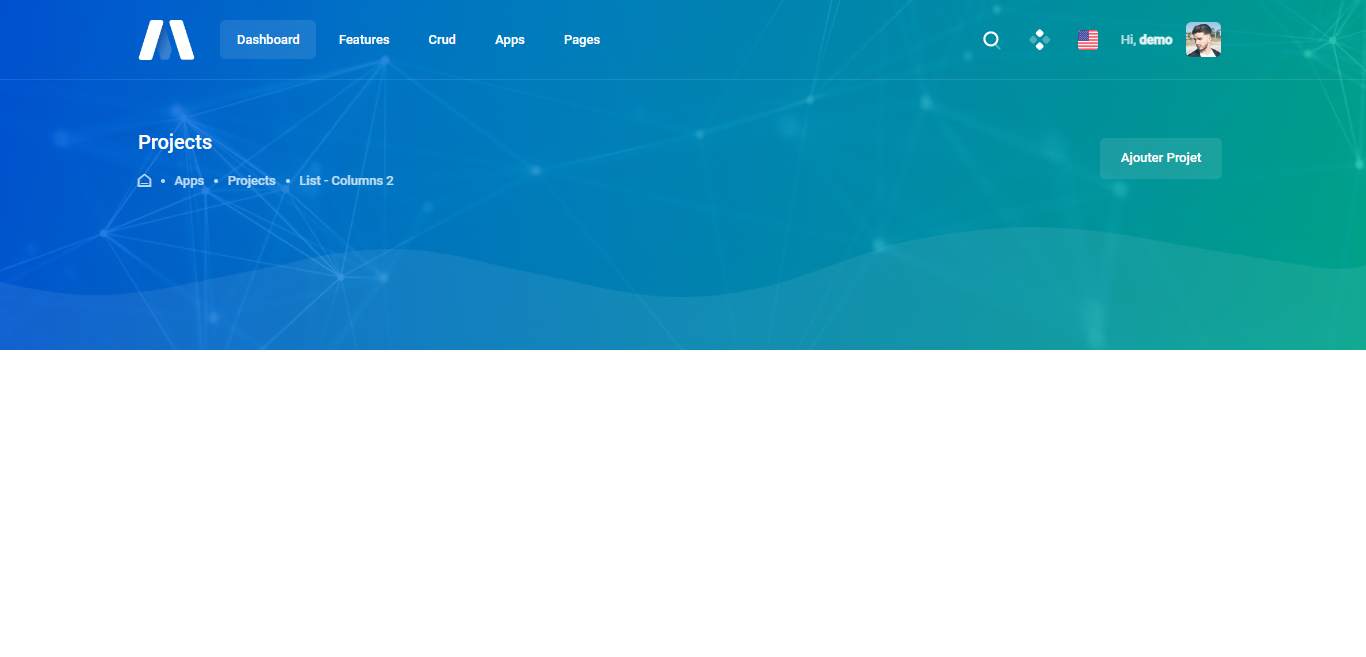


Figure 15 Plateforme : tableau de bord

Une fois authentifié, il sera redirigé vers le tableau de bord qui sera vide au début car nous n’avons pas encore ajouté de projet.

En cliquant sur le bouton « ajouter projet » vous vous retrouverez dans une page qui a un formulaire avec les champs suivants :

* Nom du projet
* Date de début
* Date prévue de fin
* La durée du sprint
* Le nombre de jour de travail par jour.



Figure 16 Plateforme : page d'ajout d'un projet

Une fois les données vérifiées et le projet ajouté avec succès à la base de données, l’utilisateur aura le choix d’ajouter un autre projet ou de rediriger vers le tableau de bord.

Le tableau de bord a maintenant le projet ajouté avec la date de début et de fin, une barre de progression et le nombre total des user stories.

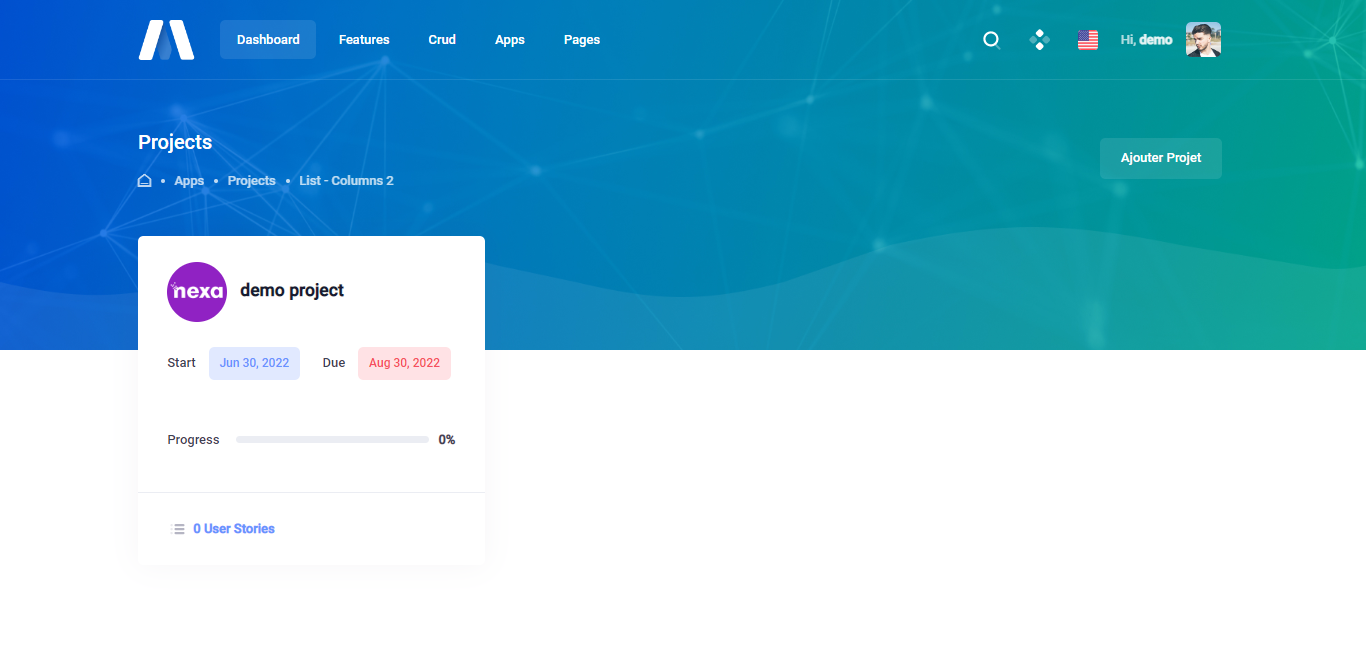


Figure 17 Plateforme : tableau de bord avec un projet ajouté

En cliquant sur la fiche du projet, l’utilisateur sera redirigé vers une page où il pourra gérer tous les aspects de son projet.

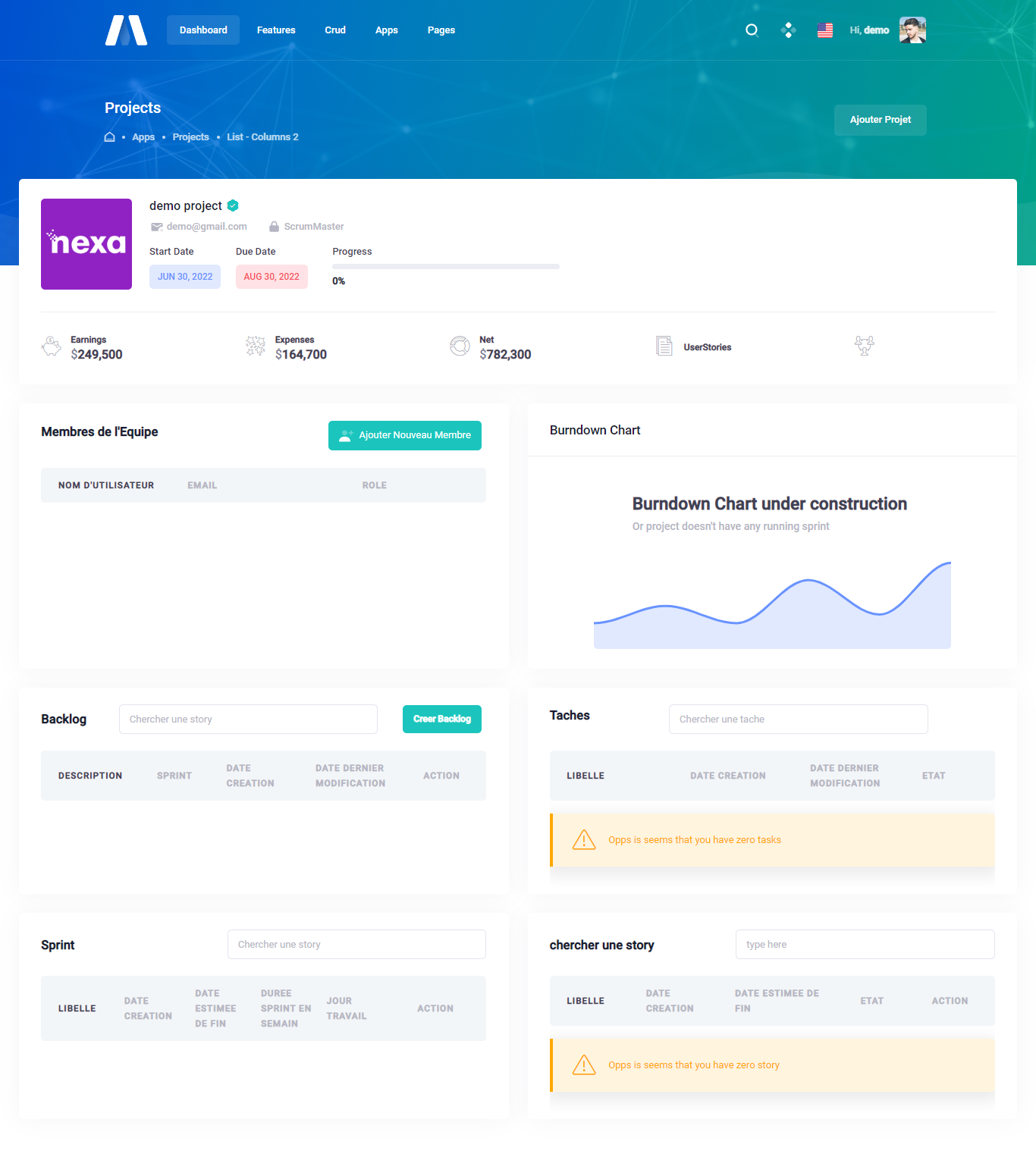


Figure 18 Plateforme : Tableau de bord du projet Scrum

En commençant par la première carte, le Scrum master peut ajouter des membres au projet qu'il soit développeur ou testeur.

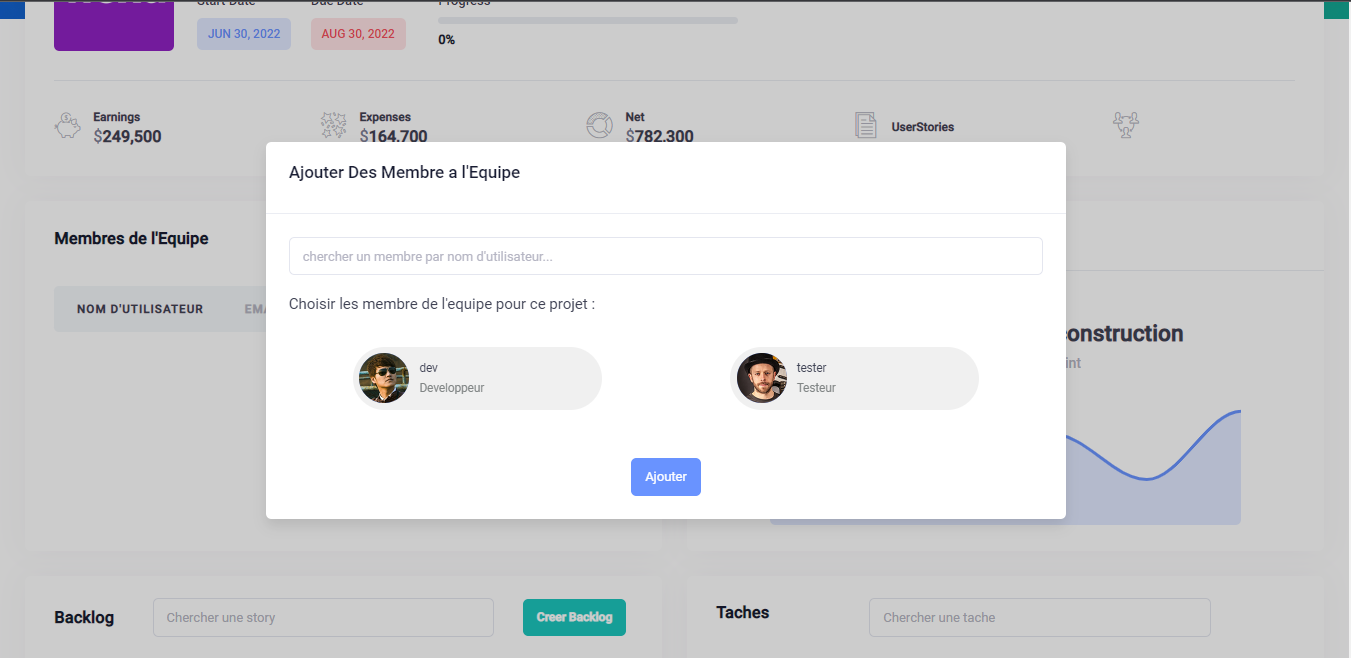


Figure 19 Plateforme : ajouter un membre

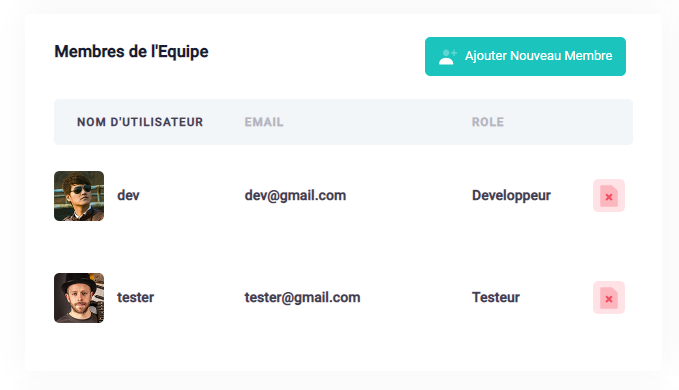


Figure 20 Plateforme : liste des membres

Afin de commencer à ajouter des user stories au Backlog, nous devons d’abord créer le Backlog en cliquant sur le bouton « Créer Backlog » qu’il ne fait rien juste initier une autre ligne de Backlog dans la base de données.

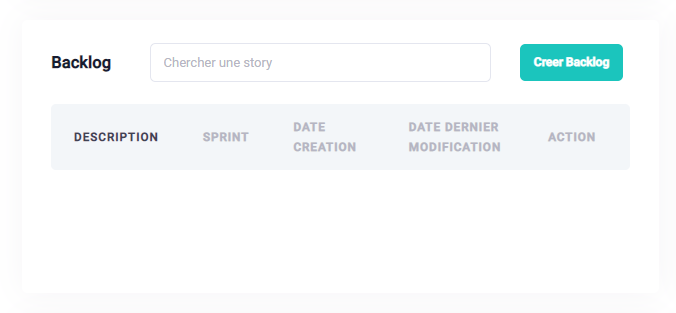


Figure 21 Plateforme : Backlog

Une fois le Backlog créé, nous pouvons y ajouter des user stories.

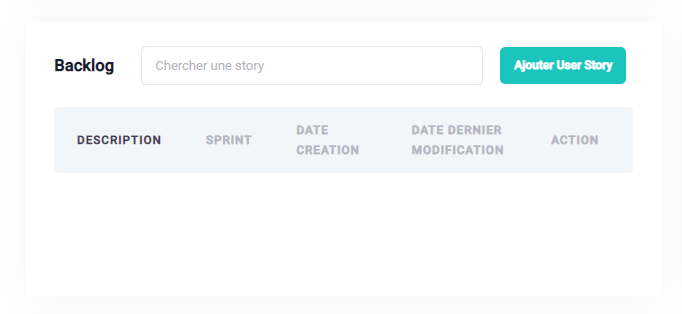


Figure 22 Plateforme : les user stories

Nous devons spécifier une description pour la user story.

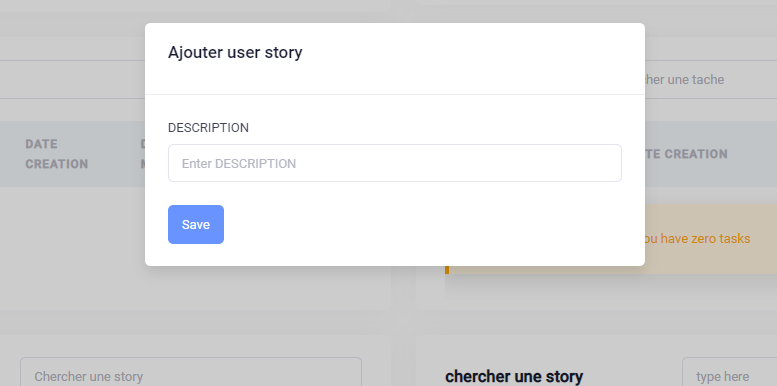


Figure 23 Plateforme : ajouter une user story

Le backlog a maintenant des user stories, chacune appartient à un sprint que nous ajouterons plus tard, une date de création et de dernière modification et un bouton pour éditer et supprimer une user story et un autre pour affecter la user story pour un membre spécifique .

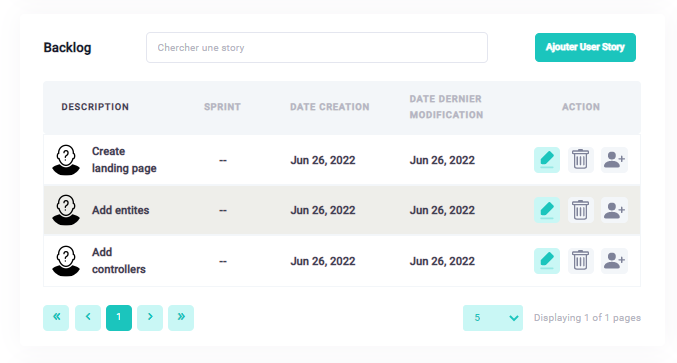


Figure 24 Plateforme : Liste des user stories

En cliquant dans chaque user story, vous pourrez ajouter des tâches pour cette dernière.

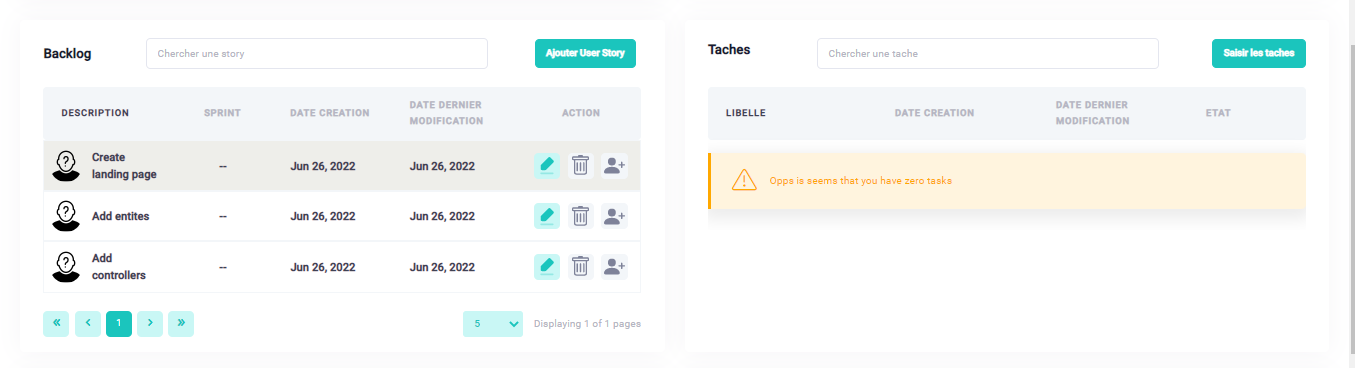
**

Figure 25 Plateforme : les tâches

Vous devez spécifier un libelle et une description pour cette tâche que vous pouvez modifier son apparence .

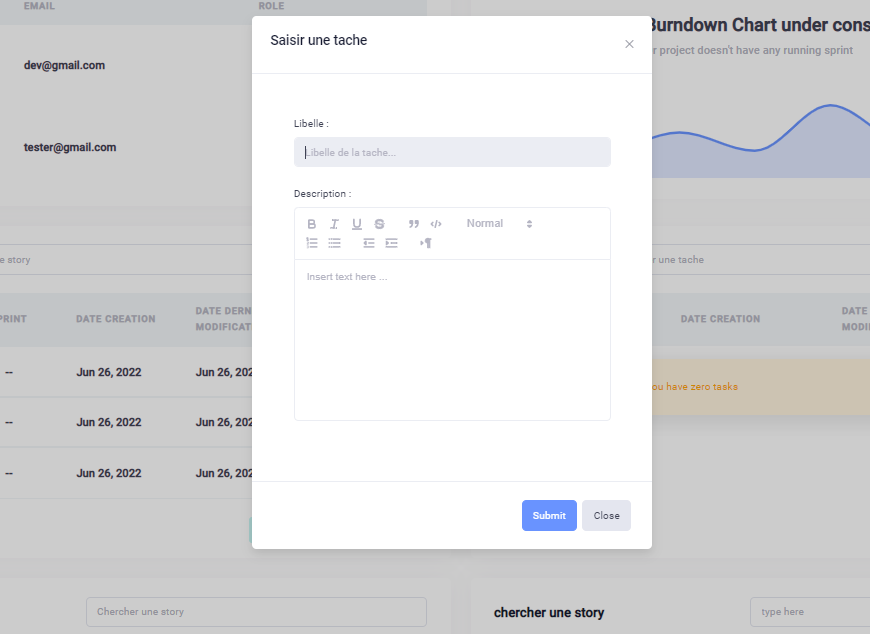


Figure 26 Plateforme : ajouter une tache

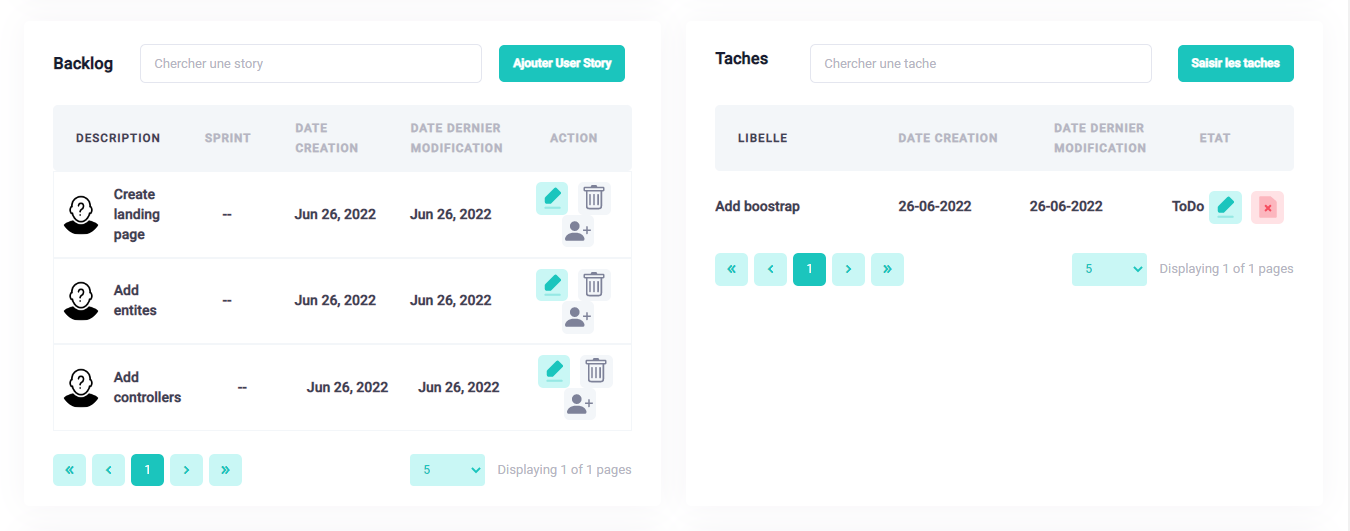


Figure 27 Plateforme : liste des tâches par user story

Une fois que nous avons fini d’ajouter des user stories et leur description, nous devons maintenant ajouter un sprint.

Nous devons spécifier un libelle, une date de fin estimée, la durée du sprint et le nombre de jours de travail.



Figure 28 Plateforme : ajouter un sprint

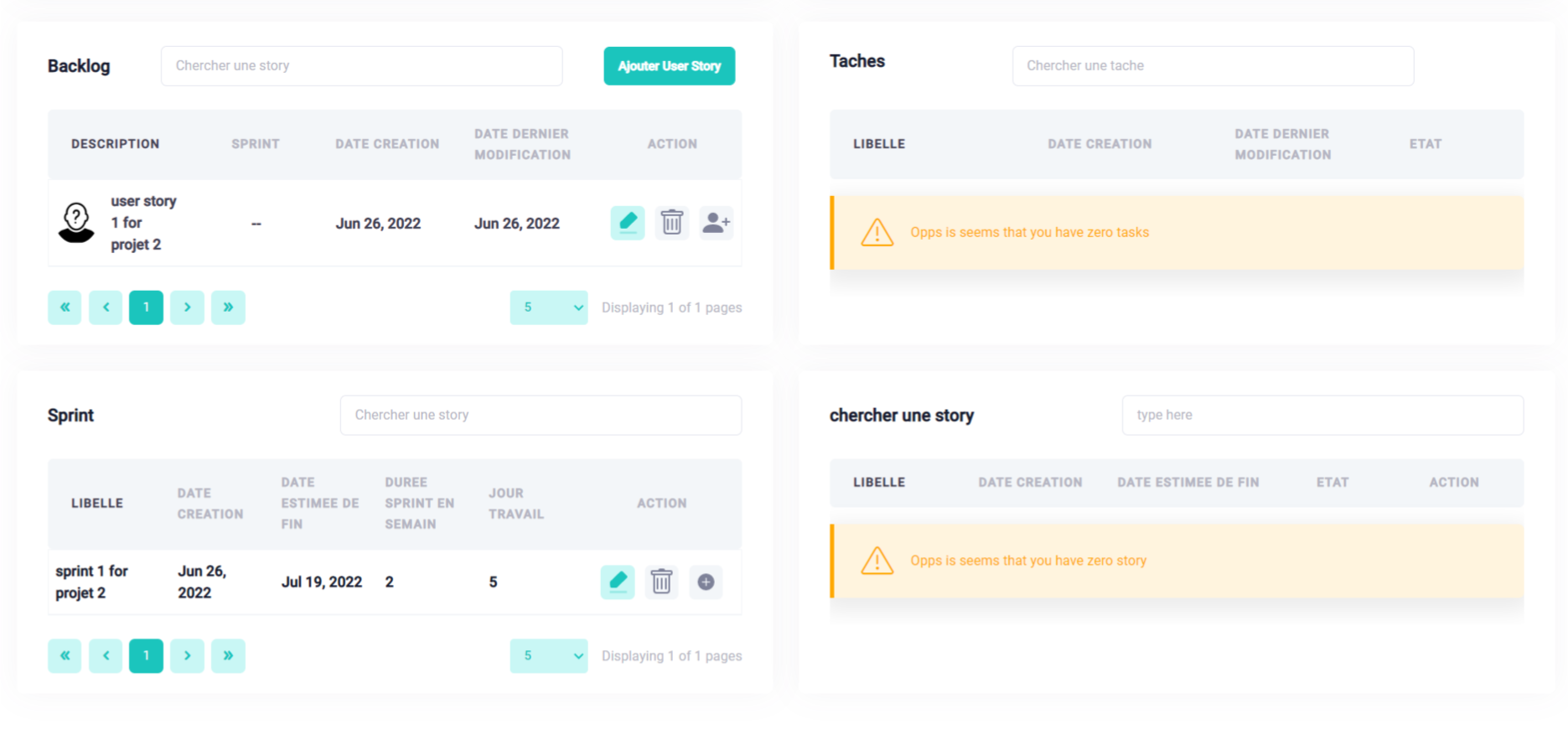


Figure 29 Plateforme : liste des sprints

Chaque sprint dispose d’une barre de recherche afin de trouver un sprint spécifique, et d’un bouton d’édition, de suppression et d’ajout pour ajouter les user stories au sprint.

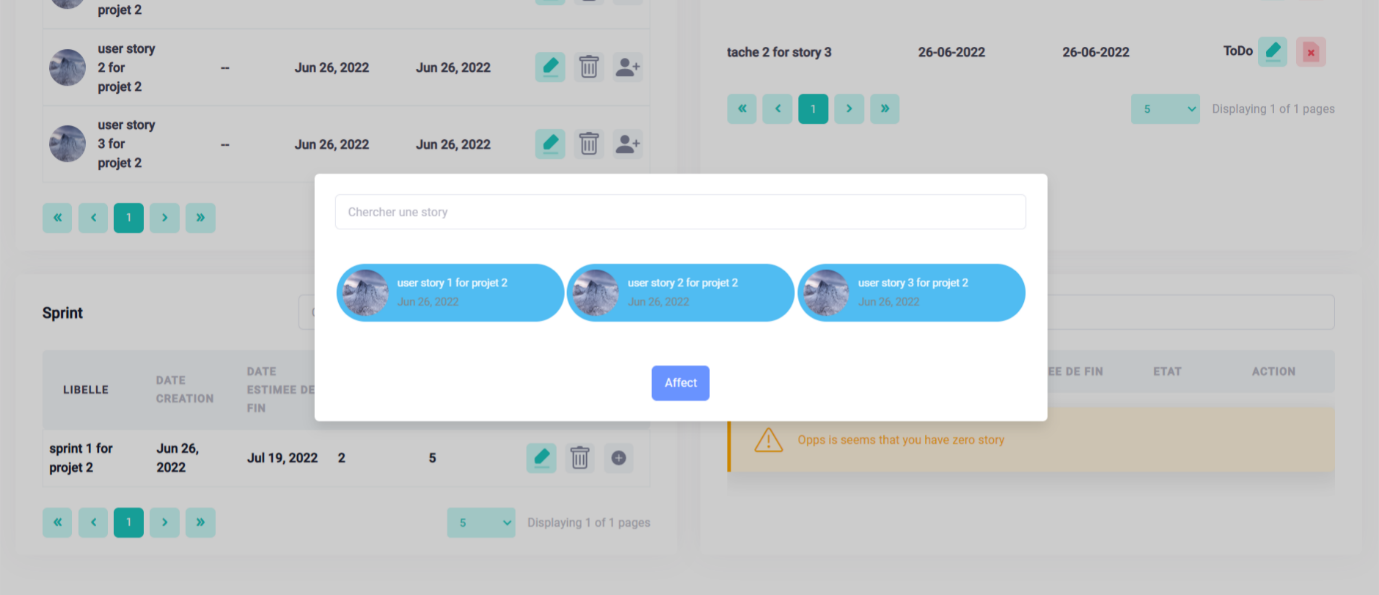


Figure 30 Plateforme : ajouter une user story à sprint

On finit par avec un sprint Backlog qui contient des user stories hautement prioritaires sur lesquelles nous allons travailler.

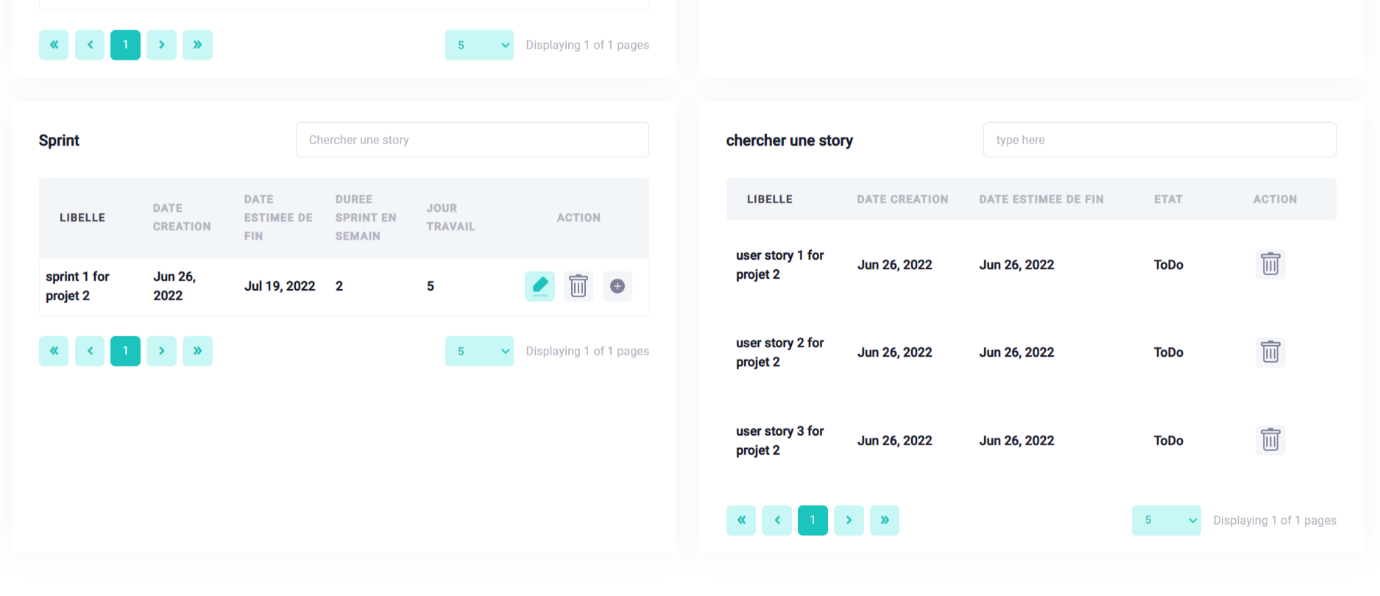


Figure 31 Plateforme : liste des user stories dans chaque sprint

Maintenant, nous devons affecter les user stories au membre de l’équipe selon les jours restants dans le sprint et les jours disponibles pour chaque membre.



Figure 32 Plateforme : affectation des user stories

D’autre part dans le compte du développeur lorsqu’il se connecte, il va trouver ses tâches dans la partie Todo du tableau de bord, une fois commencer à travailler, il doit glisser une user stories dans la partie Doing jusqu’à ce qu’il ait fini et la glisser dans la partie Done.

Lorsque le testeur se connecte, sa carte Todo sera remplie par tâches terminées par un développeur, une fois terminer de tester le travail de développement et si tous les tests sont réussis, l’user story sera glissée vers la Tested sinon elle reviendra vers la partie Doing du développeur.

Les user stories testées ont besoin d’un examen du Scrum master pour être approuvées.

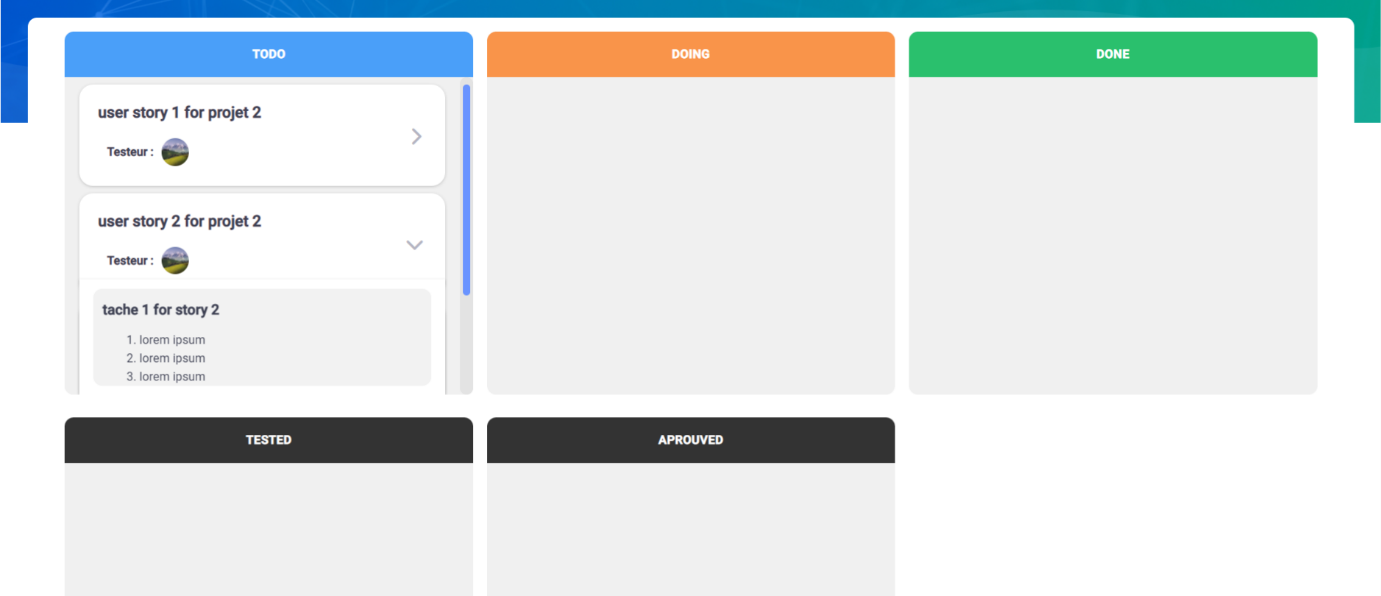


Figure 33 Plateforme : tableau de bord de Scrum

Une fois le sprint démarré, un graphique est généré afin de suivre l’avancement du projet, ce dernier a deux barres une pour l’estimation et l’autre est la réelle.

Si la barre réelle dépasse la barre estimée, le projet se déroule bien, sinon le projet a quelques problèmes et le produit sera livré en retard.



Figure 34 Plateforme : Burndown chart

# Conclusion et perspectives

La réalisation de ce projet « Conception et la réalisation d’une plateforme qui gère les projets Scrum » nous a été bénéfique sur tous les plans.

Sur le plan technique, ce projet nous a été une bonne occasion pour découvrir et maitriser les technologies ASP.Net Core et Angular, et d’approfondir nos connaissances sur le plan des nouvelles technologies de communications.

Sur le plan humain, ce projet a été une véritable occasion de vivre l’expérience de travail selon les règles Scrum, qui exige la ponctualité et l’intégration dans un groupe de travail.

Ce qui nous a permis d’améliorer nos capacités de communication et de nous adapter à la vie professionnelle. Nous avons fait de notre mieux pour bien laisser une bonne impression sur notre discipline, nos qualités et nos compétences techniques afin de présenter un travail à la hauteur de la formation qui nous avons eu au sein de l’EHEIO.

La plate-forme sera en développement continu non seulement un projet de fin d’année ordinaire, nous visons à ajouter la possibilité de choisir des tâches en utilisant le jeu de cartes poker en plus de développer une version mobile pour iPhone et Android et de l’héberger à la fin, qui sera disponible comme une solution gratuite.

# Bibliographie & Webographie

Cohn, M. (2006). Agile estimating and planning. Upper Saddle River, NJ: Pearson Education, Inc.

O'Reilly Media, Inc. (2018) Angular: Up and Running by Shyam Seshadri

Schwaber, K. (2004). Agile project management with Scrum. Redmond, WA : Microsoft Press.

Schwaber, K., & Beedle, M. (2001). Agile software development using Scrum. Upper Saddle River, NJ : Prentice Hall.

1. [**https://fr.wikipedia.org/wiki/Scrum\_(d%C3%A9veloppement)**](https://fr.wikipedia.org/wiki/Scrum_(d%C3%A9veloppement))
2. [**https://angular.io/docs**](https://angular.io/docs)
3. [**https://docs.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-5.0**](https://docs.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-5.0)
4. [**https://www.typescriptlang.org/docs/**](https://www.typescriptlang.org/docs/)
5. [**https://docs.microsoft.com/en-us/ef/core/**](https://docs.microsoft.com/en-us/ef/core/)
6. [**https://docs.npmjs.com/**](https://docs.npmjs.com/)
7. [**https://apexcharts.com/docs/installation/**](https://apexcharts.com/docs/installation/)
8. [**https://git-scm.com/docs**](https://git-scm.com/docs)
9. [**https://developer.mozilla.org/fr/docs/Web/CSS**](https://developer.mozilla.org/fr/docs/Web/CSS)
10. [**https://developer.mozilla.org/fr/docs/Web/JavaScript**](https://developer.mozilla.org/fr/docs/Web/JavaScript)
11. [**https://developer.mozilla.org/fr/docs/Web/HTML**](https://developer.mozilla.org/fr/docs/Web/HTML)
12. [**https://getbootstrap.com/docs/4.6/getting-started/introduction/**](https://getbootstrap.com/docs/4.6/getting-started/introduction/)
13. [**https://docs.microsoft.com/en-us/dotnet/csharp/**](https://docs.microsoft.com/en-us/dotnet/csharp/)
14. [**https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-5.0&tabs=visual-studio**](https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-5.0&tabs=visual-studio)
15. [**Qu'est-ce que le langage UML | Lucidchart**](https://www.lucidchart.com/pages/fr/langage-uml)
16. [**https://www.lucidchart.com/pages/fr/diagramme-de-cas-dutilisation-uml**](https://www.lucidchart.com/pages/fr/diagramme-de-cas-dutilisation-uml)
17. [**https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml**](https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml)
18. [**https://www.lucidchart.com/pages/fr/diagramme-de-sequence-uml**](https://www.lucidchart.com/pages/fr/diagramme-de-sequence-uml)