**TCH099 – Projet intégrateur en informatique Sprint 0 (Rapport d’analyse)**

**pour**

**Projet integrateur**

**Version 1.0 approuvée**

**Préparé par Azennoud Ilias, Benali Azuau, Dahmane Amine, Neacsu Theodore et Todoran Darius**

**Abdian**

**10 mars 2025**

Table des matières

[1 INTRODUCTION 1](#_Toc192888814)

[1.1 But 1](#_Toc192888815)

[1.2 Conventions des Documents 1](#_Toc192888816)

[1.3 Les Références 1](#_Toc192888817)

[2 Description générale 2](#_Toc192888818)

[2.1 Perspective du produit 2](#_Toc192888819)

[2.2 Fonctions du produit 3](#_Toc192888820)

[3 Portée du projet 3](#_Toc192888821)

[3.1 Limites du projet 3](#_Toc192888822)

[3.2 Livrables principaux 3](#_Toc192888823)

[3.3 Critères de succès du projet 4](#_Toc192888824)

[4 Analyses des besoins 4](#_Toc192888825)

[4.1 Description des parties prenantes (Stakeholders) principaux de votre produit 4](#_Toc192888826)

[4.2 Identification des besoins utilisateurs : 4](#_Toc192888827)

[4.3 Analyse des exigences fonctionnelles et non fonctionnelles: 5](#_Toc192888828)

[4.3.1 Exigences fonctionnelles: 5](#_Toc192888829)

[4.3.2 Exigences non-fonctionnelles: 9](#_Toc192888830)

[5.Gestion de projet 10](#_Toc192888831)

[5.1 Méthodologie de développement 10](#_Toc192888832)

[5.2 Planification et gestion des tâches : 11](#_Toc192888833)

[Annexe A 12](#_Toc192888834)

# 1 INTRODUCTION

## 1.1 But

Le produit concerné par ce document est la version 1.0 de la Plateforme de Réservation de Billets d’Avion **ABDIAN**. Cette version couvre le développement initial de la plateforme, incluant l’application mobile Android, l’interface Web pour les administrateurs, ainsi que l’API REST en PHP assurant la communication entre les différents systèmes. La portée de ce document couvre l’ensemble du système, incluant les fonctionnalités suivantes : la recherche et la réservation de vols, la gestion des utilisateurs et des réservations, le paiement sécurisé via PayPal et Stripe, ainsi que l’interface administrateur permettant la gestion des vols et des utilisateurs. Ce document décrit l’ensemble des exigences et fonctionnalités du système et ne se limite pas à un sous-système spécifique.

## 1.2 Conventions des Documents

Plusieurs normes et conventions typographiques ont été suivies afin d'assurer une cohérence et une compréhension claire des exigences du système. Les titres principaux sont mis en gras et en plus grand afin de les différencier clairement. Les sous-sections sont également mises en gras pour une meilleure lisibilité. Chaque énoncé d’exigence est identifié avec un numéro unique, ce qui facilite son suivi tout au long du projet. De plus, les exigences prioritaires sont classées selon trois niveaux : 1 (important), 2 (peu important), 3 (très peu important). Les exigences de niveau supérieur transmettent automatiquement leur priorité aux exigences détaillées qui leur sont associées, garantissant ainsi une hiérarchisation cohérente des éléments du projet.

## 1.3 Les Références

Voici les documents et sources utilisés dans le cadre de ce projet :

* **TCH099 - Documentation des requis**. (2025). Contient la liste des exigences fonctionnelles et non fonctionnelles. École de technologie supérieure (ÉTS), Montréal.
* **TCH099 - Documentation des cas d’utilisation**. (2025). Décrit les scénarios d’utilisation de la plateforme. École de technologie supérieure (ÉTS), Montréal.
* **TCH099 - Méthode Agile Scrum**. (2025). Présente la méthodologie de gestion de projet adoptée. École de technologie supérieure (ÉTS), Montréal.

1. **OpenAI**. (2025). Interaction avec ChatGPT (Version 4) le 11 mars 2025. Utilisé pour la clarification du Sprint 0 ainsi que la citation des sources. Également utilisé pour corriger les fautes d’orthographes.
2. Guide de rédaction d’un rapport de projet, d’un mémoire ou d’une thèse de l’ÉTS : <https://www.etsmtl.ca/etudes/guides-gabarits>

Ces documents ont permis de structurer le projet et d’assurer la conformité aux exigences académiques et techniques.

# 2 Description générale

## 2.1 Perspective du produit

Notre projet consiste à développer une plateforme en ligne permettant aux utilisateurs de rechercher, réserver et gérer des billets d’avion via une application mobile Android et une interface Web. Une API REST en PHP sera utilisée pour la communication entre les différents services.

Cette plateforme vise à simplifier le processus de réservation de vols en offrant une interface intuitive et des options de filtrage avancées. Elle s’adresse aux voyageurs fréquents ainsi qu’à toute personne souhaitant planifier un déplacement en avion de manière rapide et efficace.

Le système repose sur une architecture modulaire composée des éléments suivants :

* **Application mobile Android** : Interface principale pour les utilisateurs permettant la recherche et la réservation de vols.
* **Interface Web** : Alternative accessible depuis un navigateur pour effectuer les mêmes actions.
* **API REST en PHP** : Backend gérant les interactions entre l’application, la base de données et les services externes (paiement).
* **Base de données** : Stocke les informations relatives aux vols, réservations et utilisateurs.

## 2.2 Fonctions du produit

Le produit offrira les fonctionnalités suivantes :

* **Réservation et gestion des billets** :
  + Recherche de vols selon plusieurs critères (dates, prix, compagnies aériennes, destinations, etc.).
  + Réservation de billets directement via l’application.
  + Gestion des billets (modification, annulation, historique des réservations).
* **Recherche des vols par filtres** :
  + Filtrage avancé permettant de trouver des vols en fonction du prix, de la date ou de la compagnie.
* **Historique des réservations** :
  + Accès à l’historique des billets réservés.
  + Détails des transactions et possibilité d’émettre des factures.
* **Paiement sécurisé** :
  + Intégration des services de paiement Stripe et PayPal.
  + Chiffrement des données sensibles pour assurer la sécurité des transactions.

# 3 Portée du projet

## 3.1 Limites du projet

En règle générale, notre projet va inclure la création d’une application mobile Android avec Android Studio, le développement d’une API REST en PHP, l’intégration des systèmes de paiement avec Stripe et Paypal ainsi que la création de l’interface WEB pour les administrateurs. Il va, cependant, devoir exclure la gestion avancée des bagages et des autres services tel que la nourriture a bord de l’avion, les accès VIP, etc…, la création d'une application mobile IOS ainsi que les partenariats avec les autres compagnies aériennes.

3.2 Livrables principaux

Comme livrables principaux, notre projet inclura d’abord: une application Android écrite en Java pour permettre la gestion des billets pour les passagers. Ces derniers seront en mesure de rechercher les billets d’avions en fonction de leur date, destination et/ou prix, réserver les billets au choix et les annuler. Le paiement se fera de manière sécurisée à travers Paypal, Stripe ou une autre solution.   
Ensuite, nous livrerons une interface web pour la gestion des vols et des réservations côté administrateur, c'est-à-dire: gestion des horaires et des prix des vols disponibles, gestion des passagers. L’interface web accordera aussi l’accès aux analyses de données de réservation des billets pour suivre les ventes et les tendances.   
Finalement, le projet inclura une API REST écrite en PHP qui permettra une authentification sécurisée des utilisateurs, une gestion des opérations Create, Read, Update et Delete pour les vols, réservations, utilisateurs et paiements. L’API inclura aussi un service de recherche de vols avec gestion des critères (date, destination, prix ...) et un service lié au traitement des paiements via des API externes.

3.3 Critères de succès du projet  
Le succès de ce projet sera mesuré à travers plusieurs critères: la mesure de l’expérience de l’utilisateur pour savoir si la navigation est facile et intuitive, le niveau de sécurisation des données des utilisateurs et transactions, la stabilité et l’efficacité de la gestion des requêtes simultanées et le niveau d'optimisation de la performance lors d’un grand volume de transactions.

# 4 Analyses des besoins

## 4.1 Description des parties prenantes (Stakeholders) principaux de votre produit

* Passagers: effectuent des réservations et fournissent du feedback
* Administrateurs: gèrent les billets d’avions, prix, horaires
* Équipe de développement: Développe l’application mobile et web
* Support Client: Personne chargée d’aider les clients en leur modifications

## 4.2 Identification des besoins utilisateurs :

* S’inscrire et consulter leur propre compte
* Pouvoir faire des réservations de vols
* Consulter et annuler une réservation si applicable
* Contacter le service clientèle pour des problèmes rencontrés ou pour un remboursement
* Effectuer des recherches personnalisées sur la base de données des billets d’avions
* Effectuer des paiements sécurisés sans se soucier du vol de données
* Gérer les utilisateurs, billets de vols et réservations: créer, supprimer ou mettre à jour les données affichées aux utilisateurs
* Gérer les modalités de paiements disponibles

## 4.3 Analyse des exigences fonctionnelles et non fonctionnelles:

### 4.3.1 Exigences fonctionnelles:

Diagramme de cas d’utilisation sur l’annexe A ci-dessous

Tableau de cas d’utilisation:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **CU-001 :**  Inscription | | | **Description** | Ce cas d'utilisation décrit le processus par lequel un client s'inscrit sur le système de réservation. | | **Acteurs** | Client | | **Déclencheur** | Le client souhaite créer un compte. | | **Précondition** | Le client ne doit pas déjà avoir un compte. | | **Postcondition** | Un compte utilisateur est créé. Le client peut se connecter. | | **Résultat** | Le client dispose d'un compte actif. | | **Scénario principal** | 1. Le client accède à la page d'inscription.2. Il remplit les champs obligatoires (nom, email, mot de passe, etc.).3. Il valide l'inscription.4. Le système crée le compte et envoie une confirmation par email.5. Le client reçoit une confirmation et peut se connecter. | | **Scénario alternatif** | Si le client entre un email déjà utilisé, le système affiche un message d'erreur et lui propose de se connecter. |  |  |  | | --- | --- | | **CU-002 :**  Connexion | | | **Description** | Ce cas d'utilisation décrit le processus par lequel un client se connecte à son compte. | | **Acteurs** | Client | | **Déclencheur** | Le client souhaite accéder à son compte. | | **Précondition** | Le client doit avoir un compte existant. | | **Postcondition** | Le client est connecté au système. | | **Résultat** | Le client accède à son espace personnel. | | **Scénario principal** | 1. Le client accède à la page de connexion.2. Il entre ses identifiants (email et mot de passe).3. Il valide la connexion.4. Le système vérifie les informations et, si elles sont correctes, connecte le client.5. Le client accède à son espace utilisateur. | | **Scénario alternatif** | Si les identifiants sont incorrects, le système affiche un message d’erreur et propose de réessayer. |  |  |  | | --- | --- | | **CU-003 :**  Recherche de vol | | | **Description** | Ce cas d'utilisation permet au client de rechercher un vol selon ses critères. | | **Acteurs** | Client, service client | | **Déclencheur** | Le client souhaite trouver un vol. | | **Précondition** | Aucune. | | **Postcondition** | Une liste de vols correspondant aux critères du client est affichée. | | **Résultat** | Le client peut consulter les vols disponibles. | | **Scénario principal** | 1. Le client accède à l'interface de recherche.2. Il entre les critères de recherche (ville de départ, ville d’arrivée, dates, nombre de passagers).3. Il valide la recherche.4. Le système affiche une liste de vols disponibles correspondant aux critères.5. Le client peut consulter les détails des vols. | | **Scénario alternatif** | Si aucun vol ne correspond aux critères, le système affiche un message indiquant l'absence de résultats. |  |  |  | | --- | --- | | **CU-004 :**  Reserver un vol | | | **Description** | Ce cas d'utilisation permet au client de réserver un vol après l'avoir sélectionné. | | **Acteurs** | Client | | **Déclencheur** | Le client sélectionne un vol et souhaite le réserver. | | **Précondition** | Le client doit être connecté.Le vol sélectionné doit être disponible. | | **Postcondition** | Une réservation est enregistrée pour le client.Une confirmation de réservation est envoyée. | | **Résultat** | Le client a une réservation confirmée. | | **Scénario principal** | 1. Le client sélectionne un vol disponible.2. Il clique sur "Réserver".3. Il renseigne les informations nécessaires pour la réservation.4. Il confirme la réservation.5. Le système enregistre la réservation et envoie une confirmation par email. | | **Scénario alternatif** | Si le vol est complet, le système affiche un message d'indisponibilité.Si le paiement échoue, le client est invité à réessayer ou à modifier son mode de paiement. | |

Tableau des requits fonctionnelles:

|  |  |
| --- | --- |
| **RF-001 : Inscription des utilisateurs** | |
| **Description** | Le système doit permettre aux clients de créer un compte en fournissant leurs informations personnelles |
| **Priorité** | Haute |
| **Critère(s) d’acceptation** | -Le client peut saisir son nom, son couriel et son mot de passe  -Une confirmation par courriel est envoyée après que le client s’est inscrit |
| **Dépendance(s)** | Aucun |
| **Acteurs** | Client |
| **Cas d’usage** | CU-001 |

|  |  |
| --- | --- |
| **RF-002 : Connexion des utilisateurs** | |
| **Description** | Le système doit permettre aux utilisateurs de se connecter en utilisant un identifiant et un mot de passe |
| **Priorité** | Haute |
| **Critère(s) d’acceptation** | -L’utilisateur peut saisir son couriel et son mot de passe  -Le système vérifie la validité des informations de l’utilisateur |
| **Dépendance(s)** | Aucun |
| **Acteurs** | Client et Admin |
| **Cas d’usage** | CU-002 |
| **RF-003 : Recherche de vols** | |
| **Description** | Le client doit pouvoir rechercher un vol en fonction de critères |
| **Priorité** | Haute |
| **Critère(s) d’acceptation** | -L’utilisateur peut saisir les critères de recherche comme destination et date  -Les résultats affichent les vols correspondants à la recherche de l’utilisateur |
| **Dépendance(s)** | Aucun |
| **Acteurs** | Client |
| **Cas d’usage** | CU-003 |

|  |  |
| --- | --- |
| **RF-004 : Consultation des données** | |
| **Description** | Consulter les différents vols, billets et réservations disponibles |
| **Priorité** | Haute |
| **Critère(s) d’acceptation** | -L’utilisateur peut saisir les critères de recherche comme destination et date  -Les résultats affichent les vols correspondants à la recherche personnalisée de l’utilisateur |
| **Dépendance(s)** | Aucun |
| **Acteurs** | Client |

|  |  |
| --- | --- |
| **RF-005 : Gérer les vols et réservations** | |
| **Description** | Le client doit gérer ses vols et ses réservations |
| **Priorité** | Moyenne |
| **Critère(s) d’acceptation** | -L’utilisateur peut modifier ses vols, ses billets et ses réservations  -L’utilisateur peut annuler ses vols, ses billets et ses réservations  -Le système accepte les modifications  -Le système fait un remboursement à l’utilisateur si applicable |
| **Dépendance(s)** | Aucun |
| **Acteurs** | Client |

|  |  |
| --- | --- |
| **RF-006 : Gérer les données des vols, réservations et utilisations** | |
| **Description** | Modifier, supprimer, ajouter, lire les données des utilisateurs, billets, réservations et vols |
| **Priorité** | Moyenne |
| **Critère(s) d’acceptation** | Être en mesure de mettre à jour des données d’un utilisateur, réservation et vol |
| **Dépendance(s)** | Aucun |
| **Acteurs** | Administrateur |

### 4.3.2 Exigences non-fonctionnelles:

Tableau des requis non-fonctionnelles:

|  |  |
| --- | --- |
| **RNF-001 : Authentification sécurisée** | |
| **Description** | Authentification sécurisée des utilisateurs via JWT ou OAuth2. |
| **Justification** | Permettre une authentification sans révéler les données personnelles aux autres personnes |
| **Priorité** | Haute |
| **Critère(s) d’acceptation** | Aucun moyen d’accéder aux données personnelles des utilisateurs |
| **RNF-002 : Opérations Create, Read, Update, D** | |
| **Description** | Gestion des opérations CRUD pour les vols, réservations, utilisateurs et paiements. |
| **Justification** | Permettre la gestion des bases de données des billets, vols et utilisateurs |
| **Priorité** | Haute |
| **Critère(s) d’acceptation** | Pouvoir lire, ajouter, retirer et modifier les billets, vols et utilisateurs des bases de données |
| **RFN-003 : Service de recherche** | |
| **Description** | Service de recherche de vols avec gestion des critères (date, destination, prix, etc.). |
| **Justification** | Pouvoir trier de façon organisée les données |
| **Priorité** | Moyenne |
| **Critère(s) d’acceptation** | Le tri des données marche |
| **RFN-004 Identifiant : Traitement des paiements** | |
| **Description** | Traitement des paiements via des API externes comme PayPal, Visa ou autres. |
| **Justification** | Le payeur ne doit pas se faire scam |
| **Priorité** | L’importance relative de ce requis par rapport aux autres (ex. haute, moyenne, faible). |
| **Critère(s) d’acceptation** | Sécurisation réussie des paiements |

|  |  |
| --- | --- |
| **RFN-005 Identifiant : Echange des données** | |
| **Description** | Échanges de données via HTTP/HTTPS au format JSON. |
| **Justification** | Permettre une synchronisation des données entre l’application web, l’application mobile et la base de données |
| **Priorité** | Haute |
| **Critère(s) d’acceptation** | Synchronisation réussie et fluide |

# 5.Gestion de projet

## 5.1 Méthodologie de développement

Nous avions choisi la méthodologie de développement Scrum pour notre projet. La méthode Scrum est basée sur un principe itératif et incrémental. En effet, notre projet est divisé en plusieurs cycles de travail courts qui sont nommés sprints. Nous avions décidé d’utiliser la méthode Scrum puisque nous voulons avoir une bonne adaptation aux changements. Cette méthode nous demande de s’interagir souvent entre nous et nous avions la possibilité de le faire pour ce projet. Le projet nous demande d’interagir le plus possible avec nos membres d’équipe et nous avions une attende qu’il y ait beaucoup de changement tout au long du projet.

Nous avons 4 sprints tout au long de notre projet :

* Sprint 0 : analyse et planification
* Sprint 1 : développement des fonctionnalités de base
* Sprint 2 : intégration et tests
* Sprint 3 : déploiement, validation et remise

## 5.2 Planification et gestion des tâches :

Voici une liste des outils de gestion, communication, développement et documentation de notre projet :

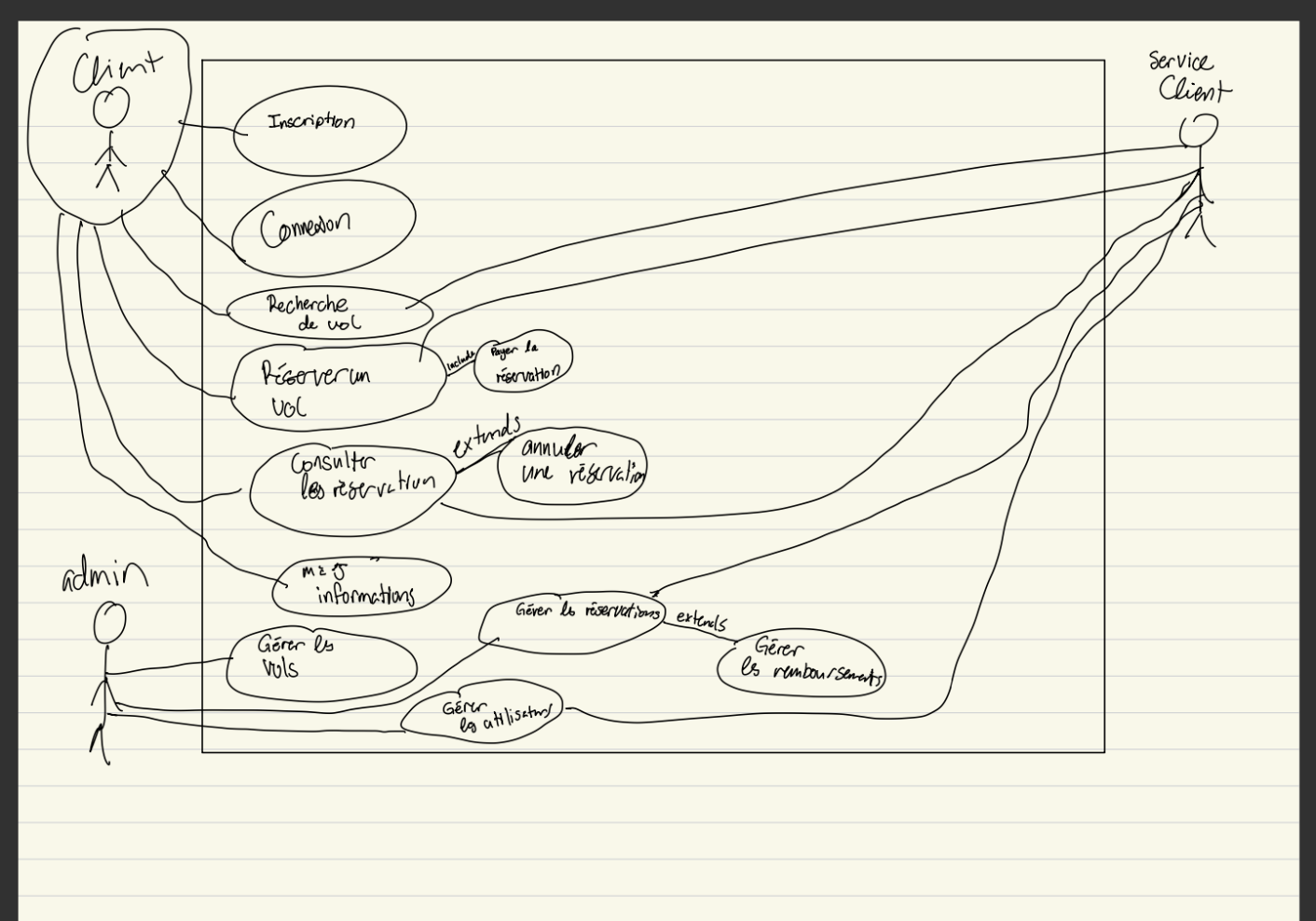
* Backlog JIRA pour le carnet du produit et du sprint.
* Discord
* SQL développer
* Visual Studio
* Android Studio
* GitHub
* IDE IntelliJ
* Onedrive

Notre équipe est formé de 5 membres qui ont chacun leur propre rôle pour le projet :

1. Theo Neacsu: Scrum Master
2. Azuau Benali: Product Owner
3. Amine Dahmane: Quality Assurance
4. Darius Todoran : Développeur
5. Ilias Azennoud: Développeur

Tous les membres font une réunion tous les jeudis et dimanche présentiel et par Discord pour planifier et s’échanger les idées sur la planification du projet. Nous faisons une révision de notre sprint chaque jour.

# Annexe A



Cas d’utilisation du système

**DoD:**

•Le code est écrit, revu et approuvé 3 personnes max.

•Le produit est déployé sur un environnement local.

• La documentation utilisateur et technique est mise à jour.

**DoR:**

•La User Story est clairement formulée et compréhensible par les développeurs.

•Les critères d'acceptation sont bien définis et validés par le Product Owner.

•L’équipe a estimé la complexité de la tâche (story points).