

Etude de cas thématique

Cahier des Charges V2

Master 2 MIAGE - UPJV

Tutrice universitaire : Anne LAPUJADE

SOMMAIRE

SOMMAIRE	2
I.Introduction	4
II.Contexte	5
II.A Description de l'organisme	5
II.B Origine du Besoin	5
II.B.1 Description générale du besoin	6
II.B.2 Choix de solution du client	6
II.B.3 Description des utilisateurs	7
III. Description Fonctionnelle	8
III.A DCU et sous-systèmes	8
III.A.1. DCU général	8
III.A.1. Sous-systèmes	9
1er sous-système : gestion des badges	9
2ème sous-système : gestion de la home page	10
3ème sous-système : gestion des murs	10
4ème sous-système : gestion des comptes	11
5ème sous-système : gestions des réactions	13
III.B Diagramme de fonctionnalités	14
III.C Description des fonctionnalités	15
Gestion des comptes	16
Consulter la liste des comptes	16
Rechercher un compte	17
Créer un compte unique	18
Création massive de comptes	19
Consulter un compte	20
Consulter son compte	21
Modifier un compte	22
Supprimer un compte	23
Finaliser inscription	24
Modifier son compte	25
Gestion des authentifications	26
Se connecter	26
Se déconnecter	27
Changer son mot de passe oublié	28
RGPD	29
Consulter la politique de confidentialité	29
Supprimer ses données	31
Gestion des réactions	32
Créer une réaction	32
Modifier une réaction	34
Supprimer une réaction	36

Consulter liste réaction	37
Gestion des murs	38
Consulter un mur	38
Rechercher un mur	40
Consulter mon mur	41
Poster une réaction	43
Gestion des classements	45
Sélectionner un classement	45
Consulter l'historique des classements	46
Consulter la home page	47
Gestion des badges	48
Créer un badge	48
Modifier un badge	51
Consulter un badge Administrateur	53
Consulter un badge Utilisateur	54
Rechercher un badge	55
Supprimer un badge	56
Consulter liste des badges	57
Attribuer un badge	58
Révoquer un badge	59
IV. Contraintes	60
IV.A Contraintes techniques	60
IV.A.1 Contraintes liées au développement	60
IV.A.2 Contraintes liées à l'infrastructure	62
IV.A.3 Contraintes liées à la gestion de projet	62
IV.B Contraintes d'environnement	62
IV.B.1 Contraintes humaines	62
IV.B.2 Contraintes financières	62
IV.B.3 Contraintes juridiques	63
IV.B.4 Contraintes temporelles	64
IV.B.5 Contraintes d'accessibilités	64
V. Identité visuelle	65
VI. Wireframe	65
VII. Risque de cette solution	68
VII.A Risques post-développement (Run)	68
Problèmes techniques après la fin du développement	68
Mises à jour futures obligatoires	68
VII.B Risque Financier	68
Différence entre le chiffrage initiale et le coût réel	68
VII.C Risque de manque de maîtrise d'œuvre	69
Manque d'expertise technique pour la réalisation de certaines parties du projet.	69
VIII. Lexique	70
IX. Annexe	72

I.Introduction

Ce document est la Version 1 du cahier des charges du projet "Step By Miage" à destination de MIAGE CONNECTION.

Il aura pour but, dans un premier temps, de décrire le contexte de ce projet, ensuite nous listerons l'intégralité des fonctionnalités initialement pensées pour ce projet avec une priorisation d'importance de ceux-ci. Enfin, nous mettrons en lumière toutes les différentes types de contraintes qu'on nous devrions prendre en compte lors de ce projet.

Ce projet était l'un des sujets proposés par notre établissement universitaire pour l'UE "Étude de cas thématique", nous l'avons choisi parmi les 9 sujets proposés, car nous avons été sensibles à l'idée de donner de notre temps et notre expertise à une association et qui plus est une association proche de nous. Il nous semblait également important de travailler avec un client externe à l'UPJV afin d'avoir une expérience la plus proche de la réalité possible.

II. Contexte

Ce projet a pour objectif de développer une solution accessible à tous les membres de l'association MIAGE CONNECTION, qui viserait à encourager l'activité physique de ses membres en leur permettant de suivre le nombre de pas réalisés quotidiennement. Ce projet a été lancé pour répondre à la problématique de la sédentarité, le tout en renforçant les interactions au sein de la communauté et en créant une compétition saine entre les membres.

II.A Description de l'organisme

Ce projet est réalisé pour l'association MIAGE CONNECTION.

MIAGE Connection est la fédération nationale des étudiants et diplômés de MIAGE, elle a pour but de fédérer les associations d'étudiants MIAGE de toute la France.

Voici une liste non-exhaustive des actions et des objectifs de cette association :

- Représenter et défendre les droits et intérêts tant individuels que collectifs des étudiants de la filière MIAGE
- Promouvoir le diplôme MIAGE auprès des futurs étudiants, des institutions et du monde professionnel
- Contribuer au développement de toute action permettant de favoriser l'insertion professionnelle des étudiants en MIAGE
- Développer les liens, les échanges, la coopération et la solidarité entre ses associations membres

Notre interlocuteur pour ce projet est Lucas Ricquebourg, aujourd'hui chargé de mission pour cette même association dans le cadre de la lutte contre la sédentarité, il en a été auparavant le président.

Ce projet de lutte contre la sédentarité est un réel combat pour notre notre interlocuteur depuis son mandat de président, cependant aucune solution informatique ressemblant à ce projet n'a encore été pensée et mise en place dans l'association MIAGE Connection.

II.B Origine du Besoin

Un mode de vie sédentaire, caractérisé par des intervalles prolongés passés en position assise et sans faire d'effort physique, est une préoccupation accrue parmi les personnels des bureaux, notamment pour les étudiants et professionnels issus de

formation MIAGE. Une étude réalisée en 2023 par Santé publique France montre que les employés de bureau restent souvent assis pendant de longues périodes sans interruption. Ce mode de vie sédentaire amplifie certains risques pour la santé, notamment les maladies cardiaques, le diabète de type 2, certaines tumeurs malignes, le découragement et le malaise. Ces informations soulignent l'importance de plaider en faveur de stratégies productives visant à réduire la durée des comportements sédentaires sur le lieu de travail et dans la vie de tous les jours et à améliorer le bien-être du personnel.

II.B.1 Description générale du besoin

L'objectif est donc de concevoir un outil permettant de recueillir le nombre de pas de chaque adhérent de MIAGE CONNECTION , qu'il soit membre d'un BDE Miage, d'une Junior entreprise ou d'une association de vieux afin d'encourager l'activité physique. L'idée est de rendre l'outil ludique et d'inciter à la compétition entre les adhérents, villes et associations d'adhérents pour mieux atteindre les objectifs de mobilité.

II.B.2 Choix de solution du client

Afin de répondre au besoin de notre client, nous avons élaboré une note de cadrage concernant plusieurs solutions disponibles, ce document recensait toutes les informations des différentes solutions, nécessaires pour que notre client prenne la décision qui lui convient le mieux.

Suite à plusieurs échanges, il a été acté que la solution sera une application mobile disponible gratuitement sur le Play Store d'Android et l'App Store d'IOS (documents disponibles en annexe)

II.B.3 Description des utilisateurs

Utilisateur



- Consulter son profil.
- Consulter les murs.
- Consulter les différents classements.
- Participer aux concours.
- Interagir avec les murs (réaction)
- Changer son mot de passe (oublié)

Administrateur



- Gérer les comptes (créer, modifier, ou supprimer des comptes utilisateurs).
- Gérer les réactions (consulter, créer, modifier, supprimer)
- Gérer les badges (créer, modifier, supprimer)
- Attribuer et révoquer des badges.

Utilisateur non inscrit



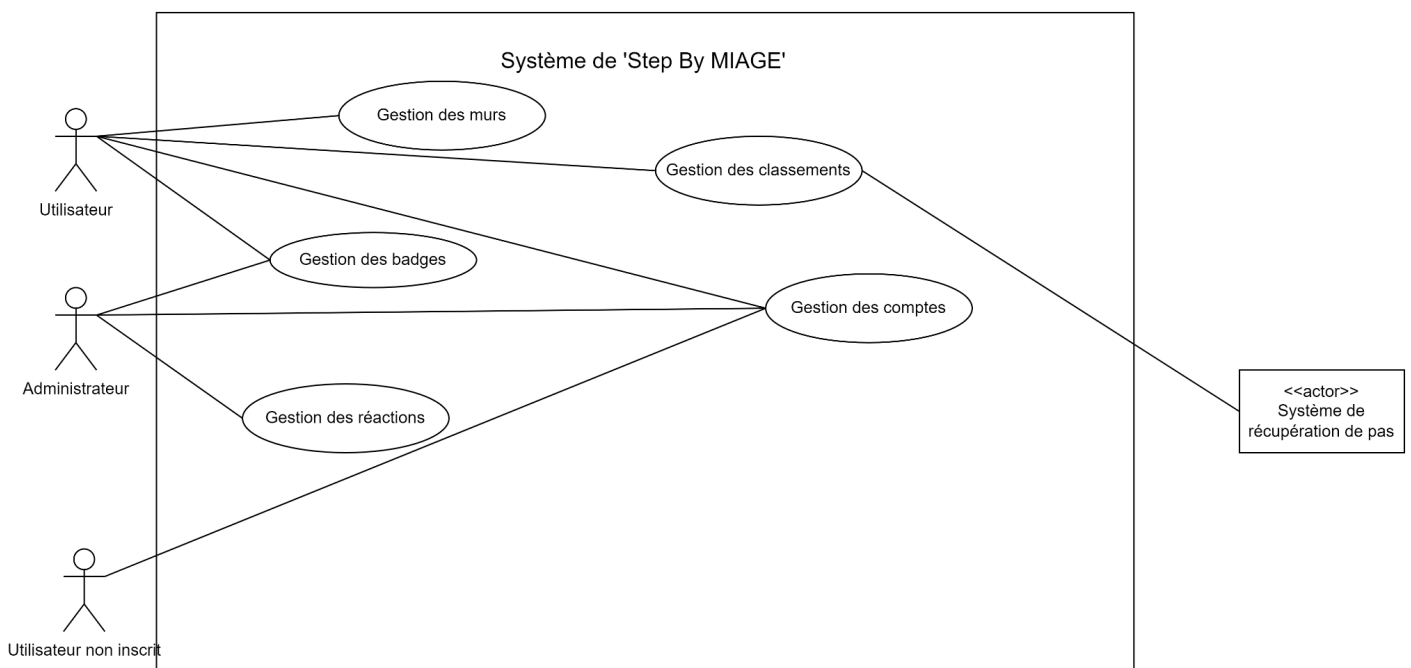
- Finaliser l'inscription

III. Description Fonctionnelle

La description fonctionnelle est une étape cruciale, c'est dans cette partie qu'une liste de fonctionnalités spécifique à la demande client est établie, ce qui constitue des bases fondamentales pour la future conception de cette solution.

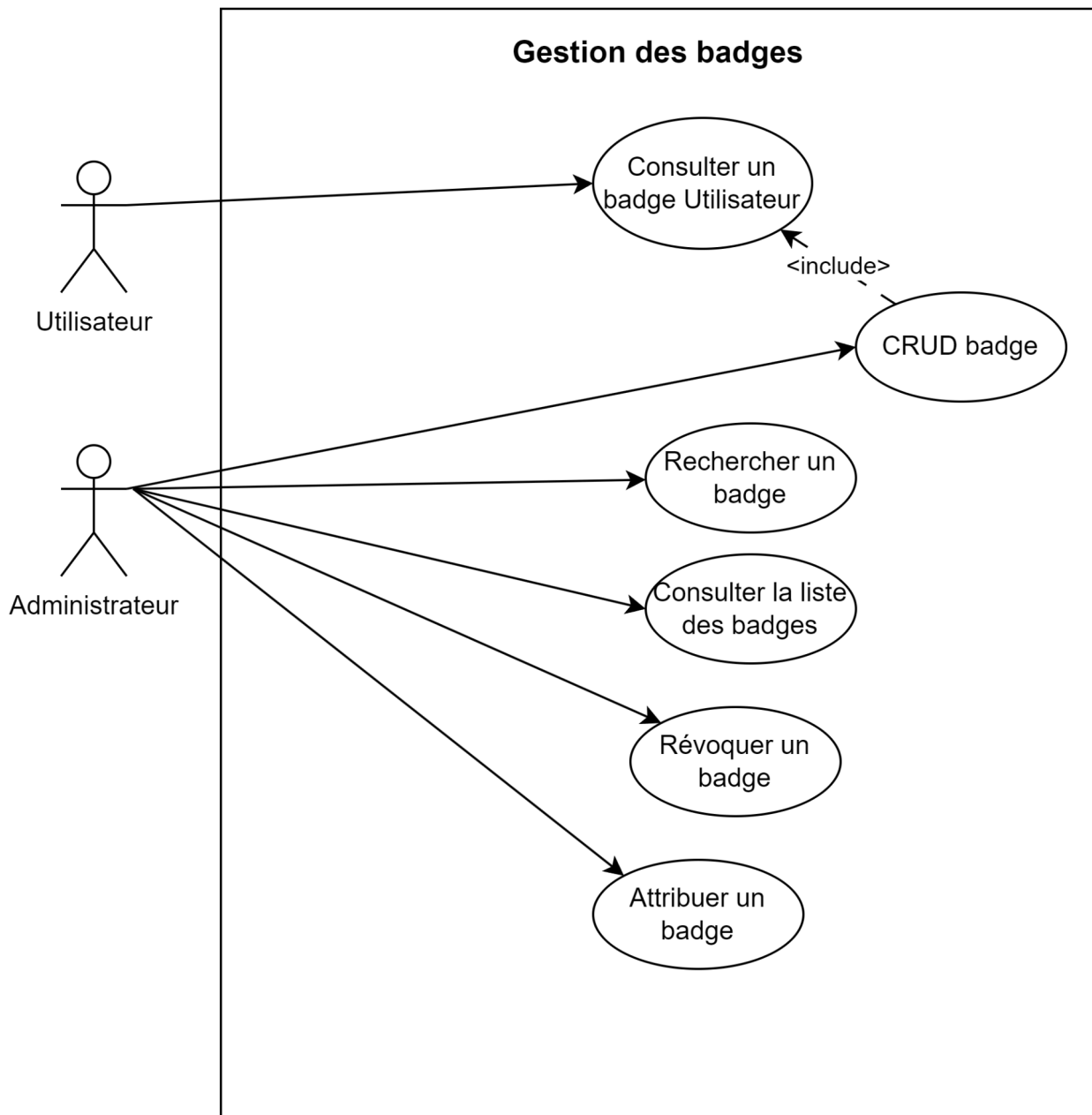
III.A DCU et sous-systèmes

III.A.1. DCU général

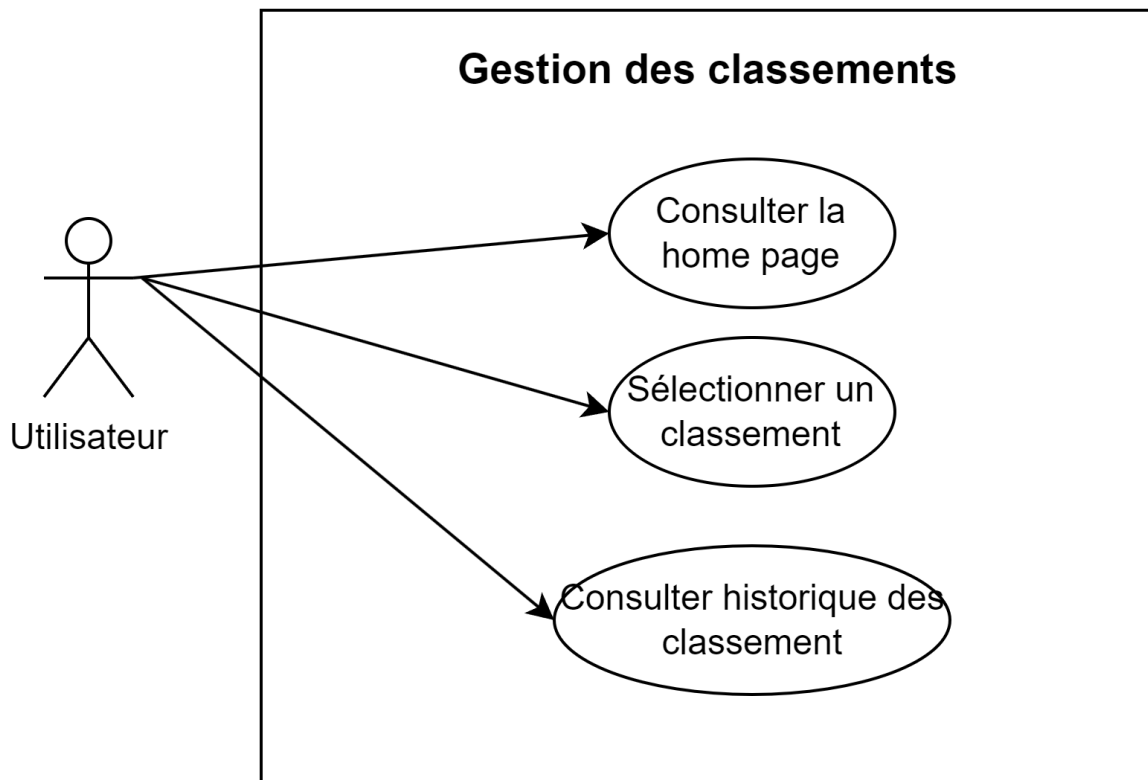


III.A.1. Sous-systèmes

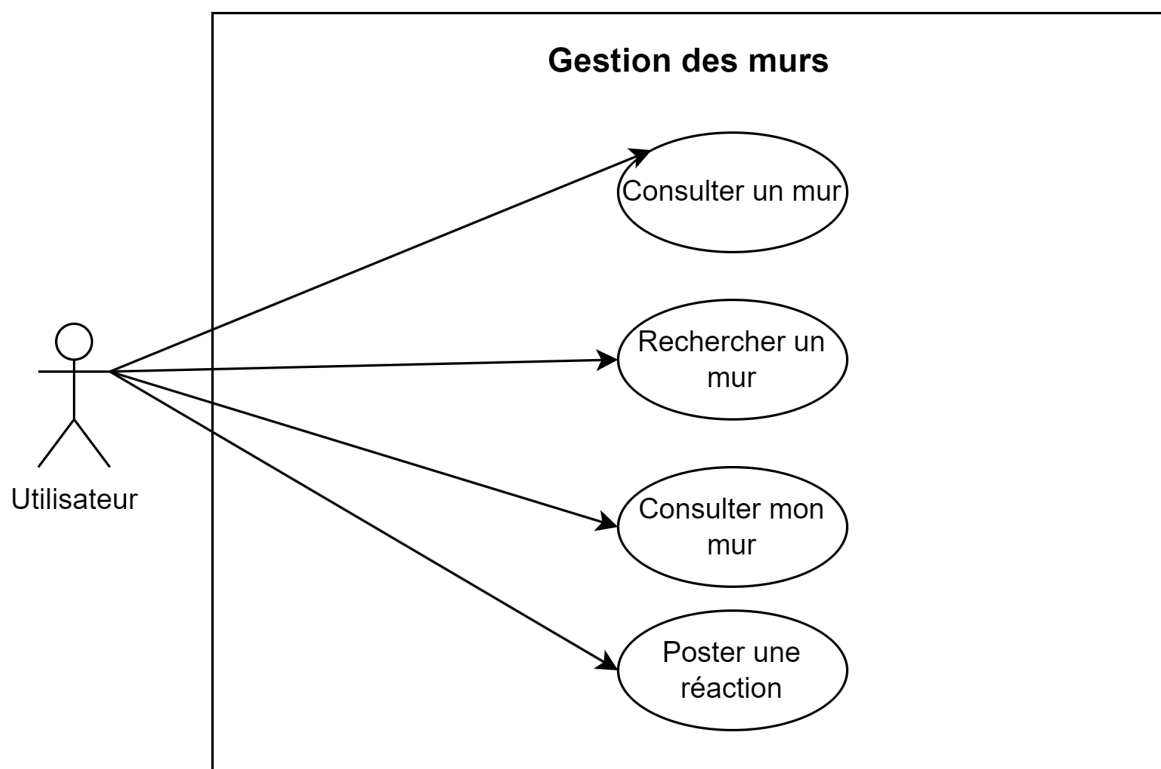
1er sous-système : gestion des badges



2ème sous-système : gestion de la home page

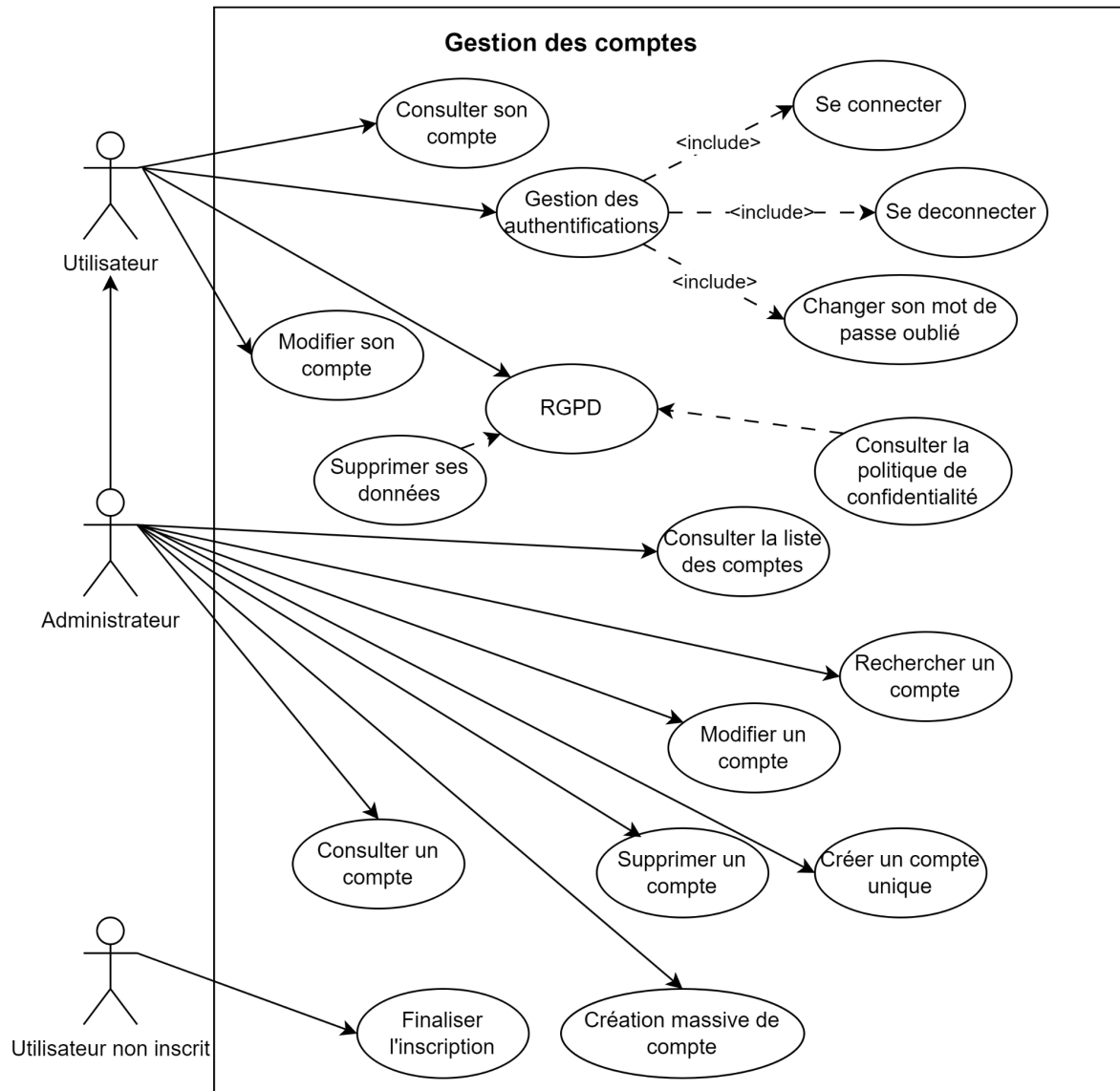


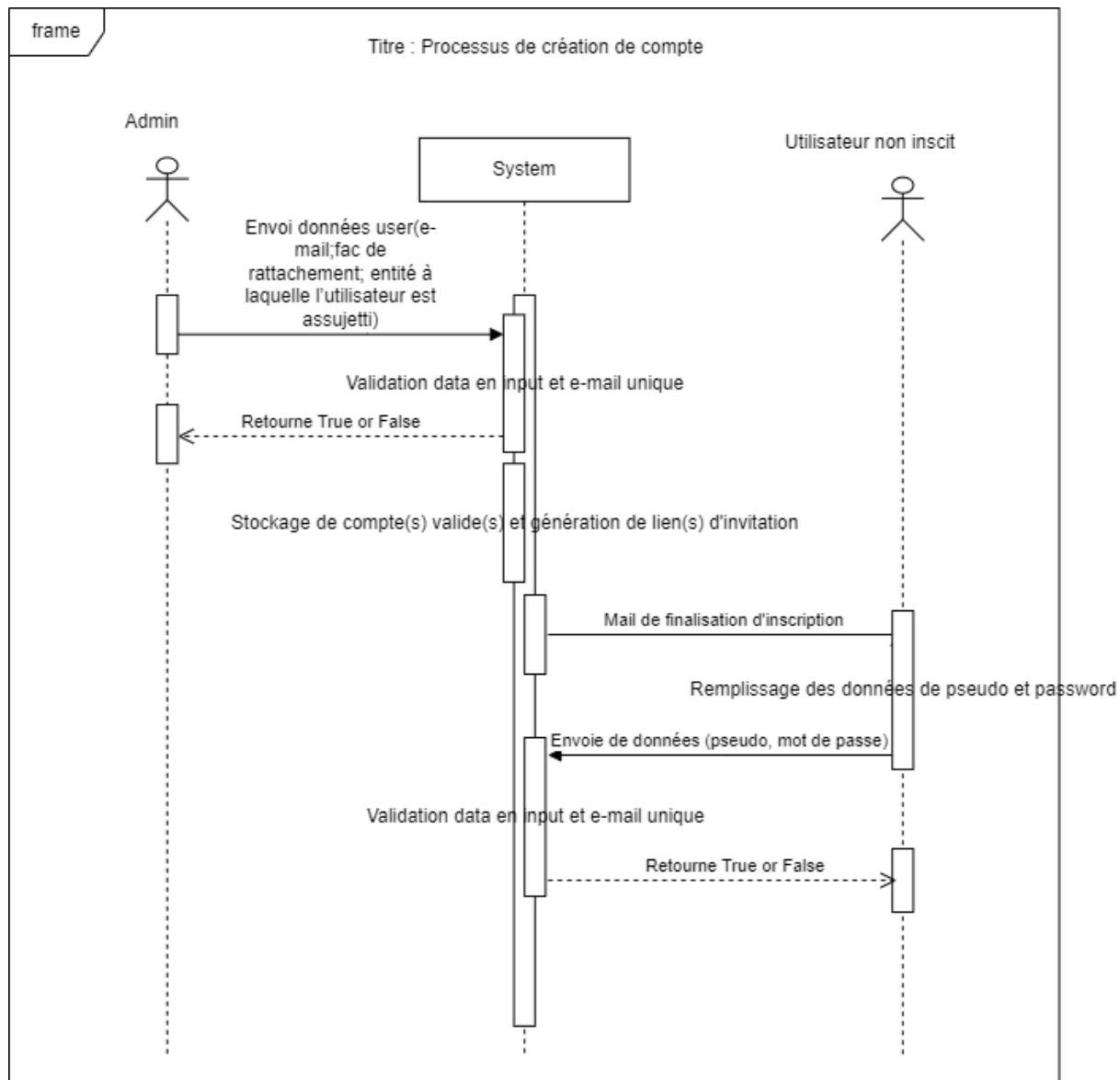
3ème sous-système : gestion des murs



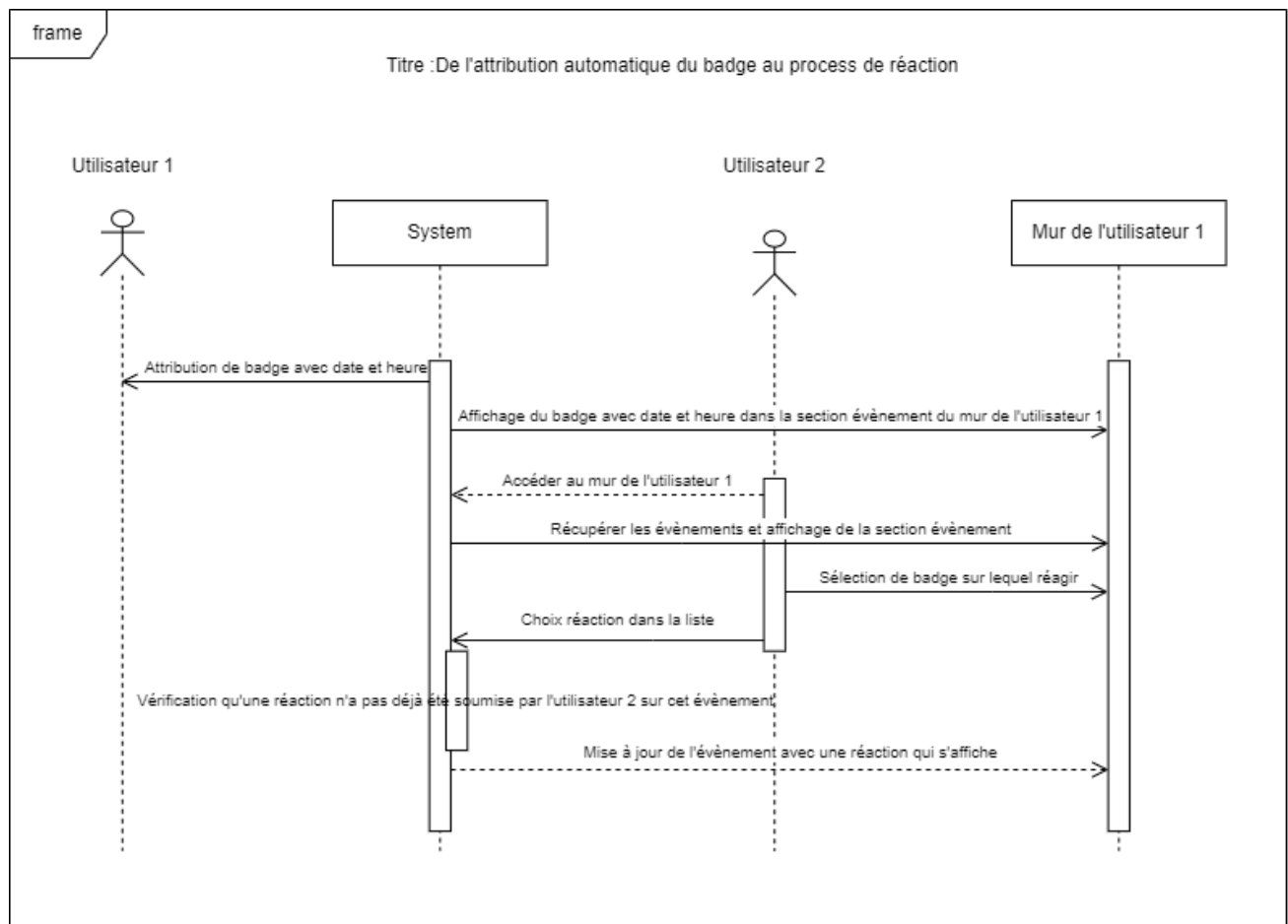
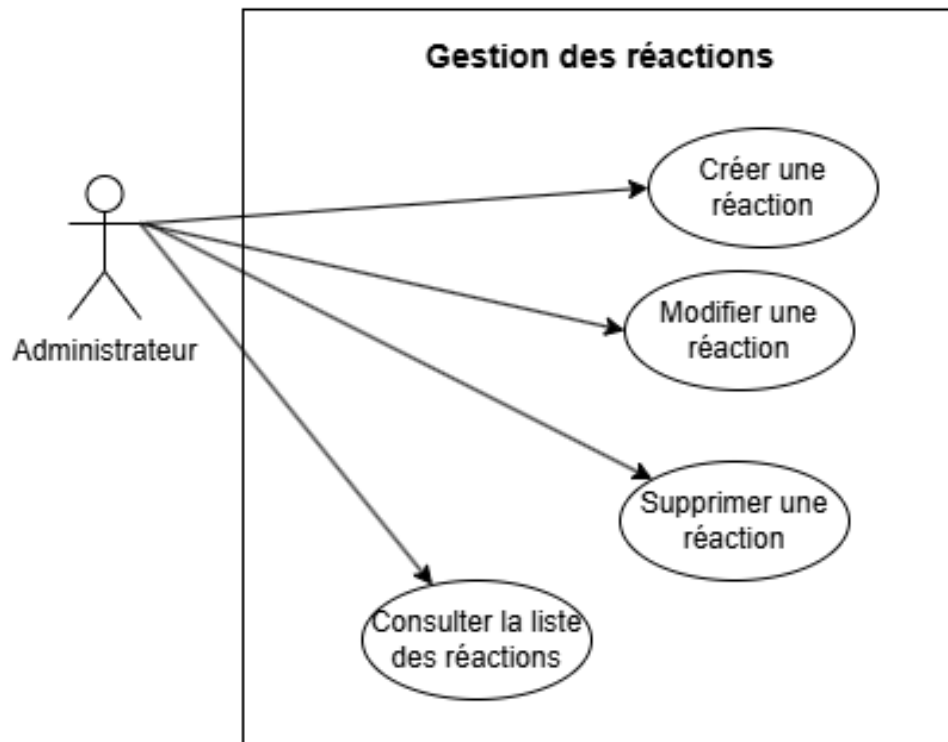
4ème sous-système : gestion des comptes

Gestion des authentifications + RGPD



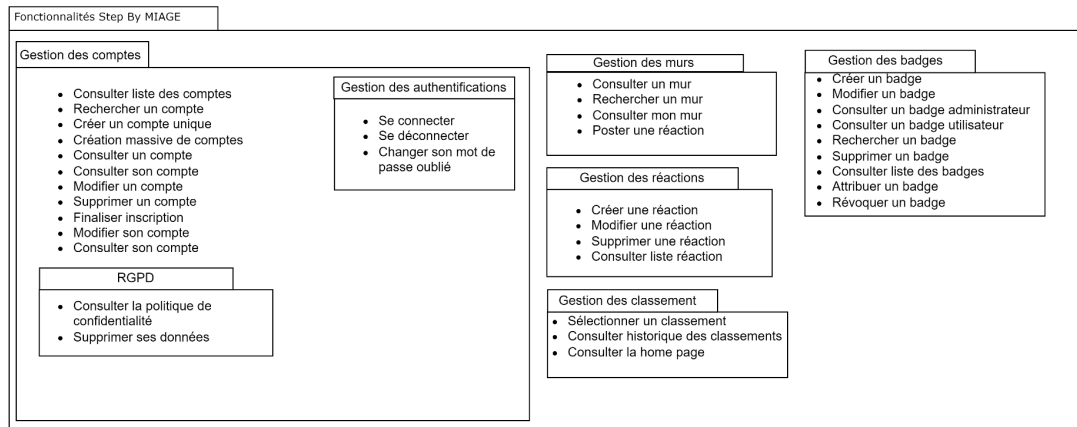


5ème sous-système : gestions des réactions



III.B Diagramme de fonctionnalités

De cette analyse détaillée des sous-systèmes de notre application découle un diagramme de fonctionnalités (package) que voici ci-dessous, permettant de représenter la segmentation des sous-systèmes et de leurs fonctionnalités dans notre application.



III.C Description des fonctionnalités

Chacune de ces fonctionnalités sera prioriser selon la méthode MoSCoW, méthode dans laquelle 4 niveaux de priorisation existe :

- Must : signifie "doit être fait"
- Should : signifie "devrait être fait dans la mesure du possible"
- Could : signifie "pourrait être fait si cela n'impacte pas d'autres tâches"
- Would : signifie "peut être fait plus tard"

Liste des statistiques	
Libellé	Description
Nombre de pas total	Nombre de pas en cumulé sur l'ensemble du compte
Nombre de pas journalier	Nombre de pas sur la journée de l'utilisateur
Nombre de pas médian par mois	Nombre de pas médian sur le mois
Streak flamme	Un compteur sera incrémenté de 1 tous les 3000 pas chaque jour. S'il rate un jour, le compteur sera réinitialisé.
Moyenne communautaire	Moyenne de tous les utilisateurs rattachés à une entité (université, asso)
Meilleur score quotidien	Nombre de pas le plus haut quotidien enregistré sur le compte
Graph mois	Graph des pas journalier sur le mois en cours

Liste des Fonctionnalités :

Gestion des comptes

- Consulter la liste des comptes

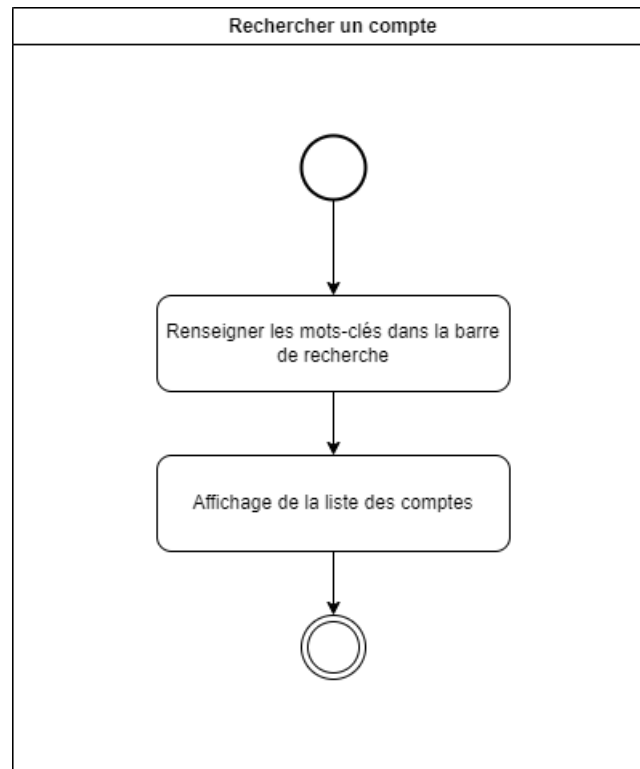
Nom de la fonctionnalité : Consultez la liste des comptes
Utilisateur : Administrateur
Description de la fonctionnalité : Permet à l'administrateur d'afficher une liste de tous les comptes. <div data-bbox="496 692 1080 1388" data-label="Diagram"> <pre> graph TD Start(()) --> UseCase[Affichage de la liste des comptes] UseCase --> End((())) </pre> <p>The diagram is a UML Use Case Diagram titled "Consulter la liste des comptes". It shows a single use case represented by a rounded rectangle labeled "Affichage de la liste des comptes". The flow starts from an initial state (represented by a circle) and ends at a final state (represented by a double circle), both connected to the use case by vertical arrows.</p> </div>
Règles de Gestion : L'administrateur peut trier la liste par rôle ou entité.
Autres contraintes :
Entrée :
Sortie : Liste des comptes (pseudo, e-mail, entité , rôle)
Priorité : Obligatoire (Must Have)

- Rechercher un compte

Nom de la fonctionnalité : Rechercher un compte

Utilisateur : Administrateur

Description de la fonctionnalité : Permet à l'administrateur de rechercher un compte via une entrée en barre de recherche depuis la liste des comptes.



Règles de Gestion : Empêcher les injections malveillantes, permettre de filtrer sur toutes les colonnes de la liste des comptes.

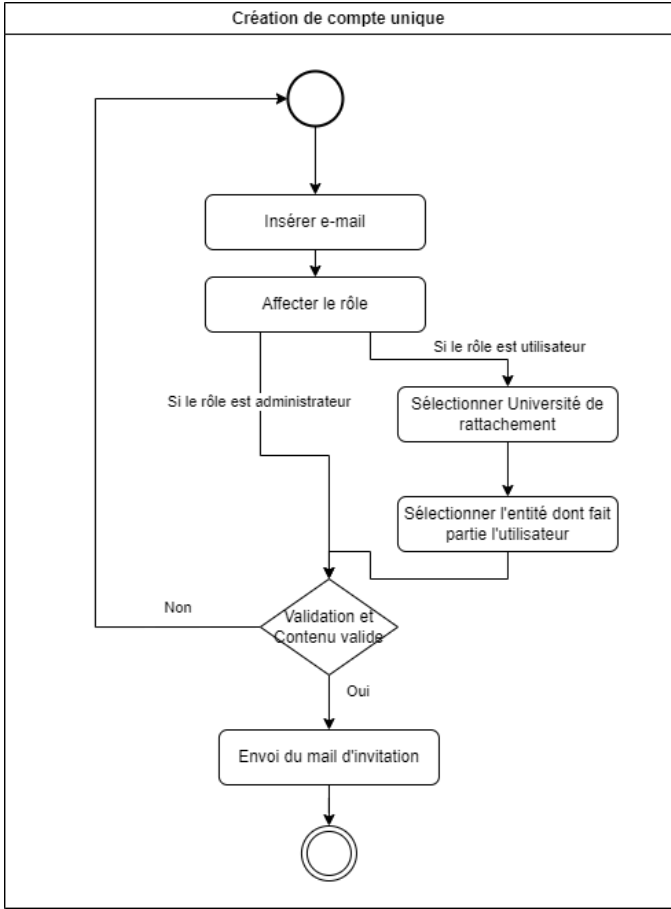
Autres contraintes : Empêcher l'entrée de certains caractères.

Entrée : Un ou plusieurs mots-clés.

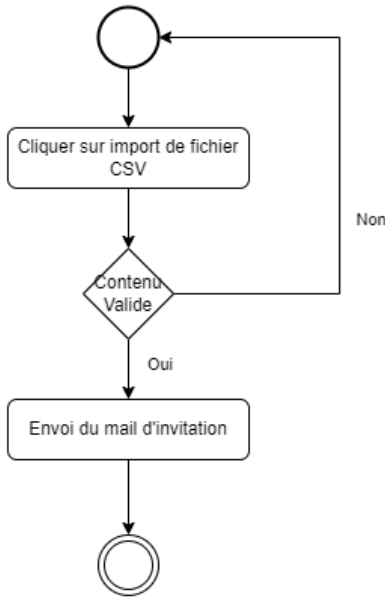
Sortie : Liste des comptes (pseudo, e-mail, entité , rôle) filtrée en fonction de l'entrée.

Priorité : Obligatoire (Must Have)

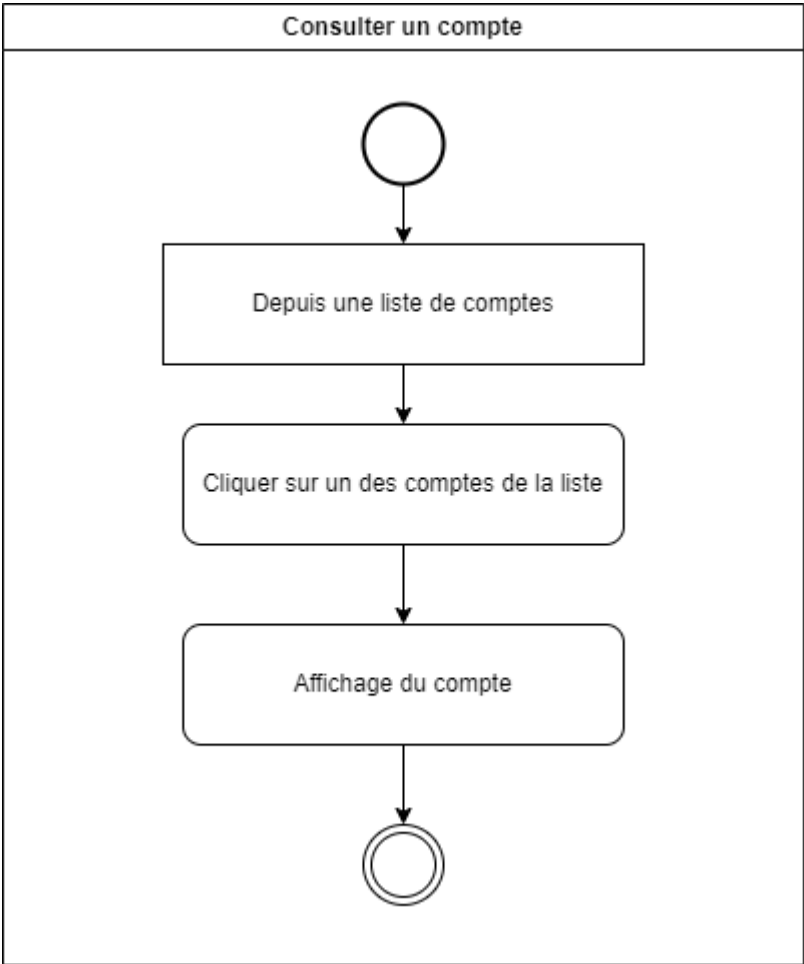
- Créer un compte unique

Nom de la fonctionnalité : Création de compte unique
Utilisateur : Administrateur
<p>Description de la fonctionnalité : Cette fonctionnalité permet aux administrateurs de créer des utilisateurs et de leur attribuer leur rôle.</p>  <pre> graph TD Start(()) --> InsertEmail[Insérer e-mail] InsertEmail --> AssignRole[Affecter le rôle] AssignRole -- "Si le rôle est administrateur" --> Validation{Validation et Contenu valide} AssignRole -- "Si le rôle est utilisateur" --> SelectUni[Sélectionner Université de rattachement] SelectUni --> SelectEntity[Sélectionner l'entité dont fait partie l'utilisateur] SelectEntity --> Validation Validation -- "Non" --> Start Validation -- "Oui" --> SendMail[Envoi du mail d'invitation] SendMail --> End((())) </pre>
Règles de Gestion : L'email devra être unique et ne pas déjà exister dans le système.
Autres contraintes : Vérifier la validité de l'e-mail, toutes les données doivent être renseignées, le compte doit avoir au minimum un rôle.
Entrée : E-mail, la fac de rattachement, l'entité à laquelle l'utilisateur est assujetti (càd BDE OU association de vieux OU Junior Entreprise OU membre de bureau Miage Connection) et le rôle.
Sortie : Réussite ou échec
Priorité : Obligatoire (Must Have)

- Création massive de comptes

Nom de la fonctionnalité : Création massive de comptes
Utilisateur : Administrateur
Description de la fonctionnalité : Cette fonctionnalité permet aux administrateurs de créer des utilisateurs et de leur attribuer leur rôle massivement via import de fichier CSV.
<p>Règles de Gestion : Les emails doivent être uniques et ne pas déjà exister dans le système. Empêcher les injections malveillantes</p> <div data-bbox="438 698 1134 1438" data-label="Diagram"> <p style="text-align: center;">Création de compte massive</p>  <pre> graph TD Start(()) --> Click[Cliquer sur import de fichier CSV] Click --> Valid{Contenu Valide} Valid -- Non --> Start Valid -- Oui --> Send[Envoi du mail d'invitation] Send --> End((())) </pre> </div>
Autres contraintes : Vérifier la validité des e-mail, toutes les données doivent être renseignées, le compte doit avoir au minimum un rôle. Empêcher l'entrée de certains caractères.
<p>Entrée : Fichier CSV avec des entrées de la forme : E-mail;fac de rattachement; entité à laquelle l'utilisateur est assujetti (càd BDE ou association de vieux OU Junior Entreprise OU membre de bureau Miage Connection);rôle.</p>
Sortie : Réussite ou Échec
Priorité : Bien de l'avoir (Could Have)

- Consulter un compte

Nom de la fonctionnalité : Consulter un compte
Utilisateur : Administrateur
Description de la fonctionnalité : Permet à l'administrateur de consulter un compte depuis la liste des comptes. <div data-bbox="392 521 1198 1482" data-label="Diagram"> <p style="text-align: center;">Consulter un compte</p>  <pre> graph TD Start(()) --> A[Depuis une liste de comptes] A --> B(Cliquer sur un des comptes de la liste) B --> C(Affichage du compte) C --> End((())) </pre> </div>
Règles de Gestion :
Autres contraintes :
Entrée :
Sortie : pseudo, e-mail, entité , rôle, mot de passe
Priorité : Obligatoire (Must Have)

- Consulter son compte

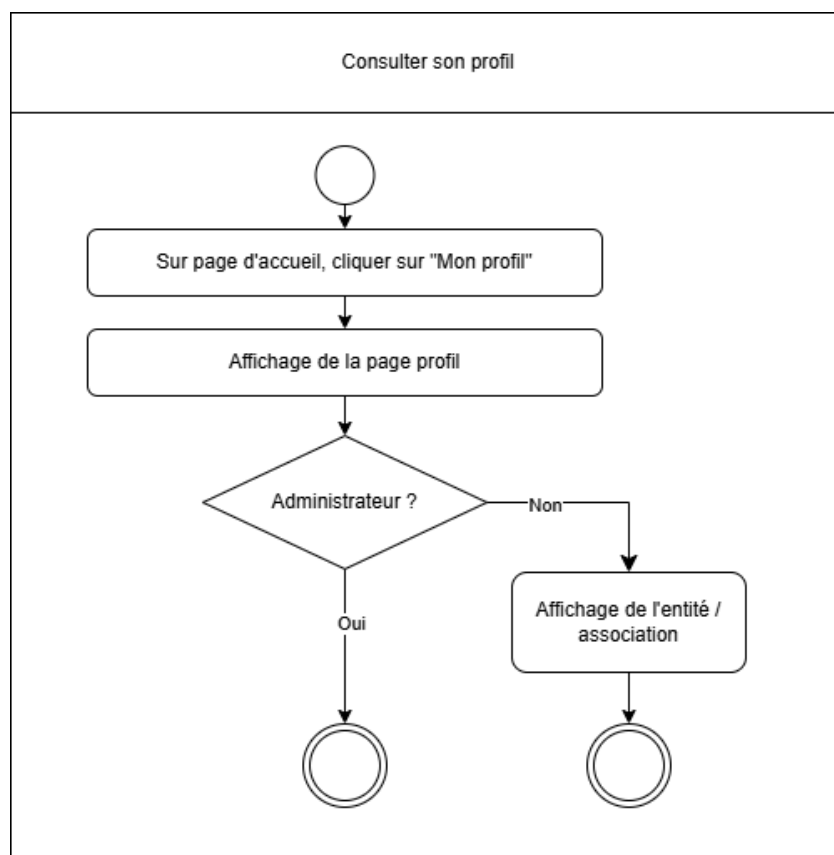
Nom de la fonctionnalité : Consulter son compte

Utilisateur : Utilisateur / Administrateur

Description de la fonctionnalité : Cette fonctionnalité permet de consulter sa page de compte. Nous pourrions retrouver les éléments suivants :

- Photo de profil
- Pseudo
- Adresse mail

S'il est utilisateur, les informations sur les entités sont renseignées.



Règles de Gestion :

Autres contraintes :

Entrée : Compte de l'utilisateur

Sortie : affichage de la page

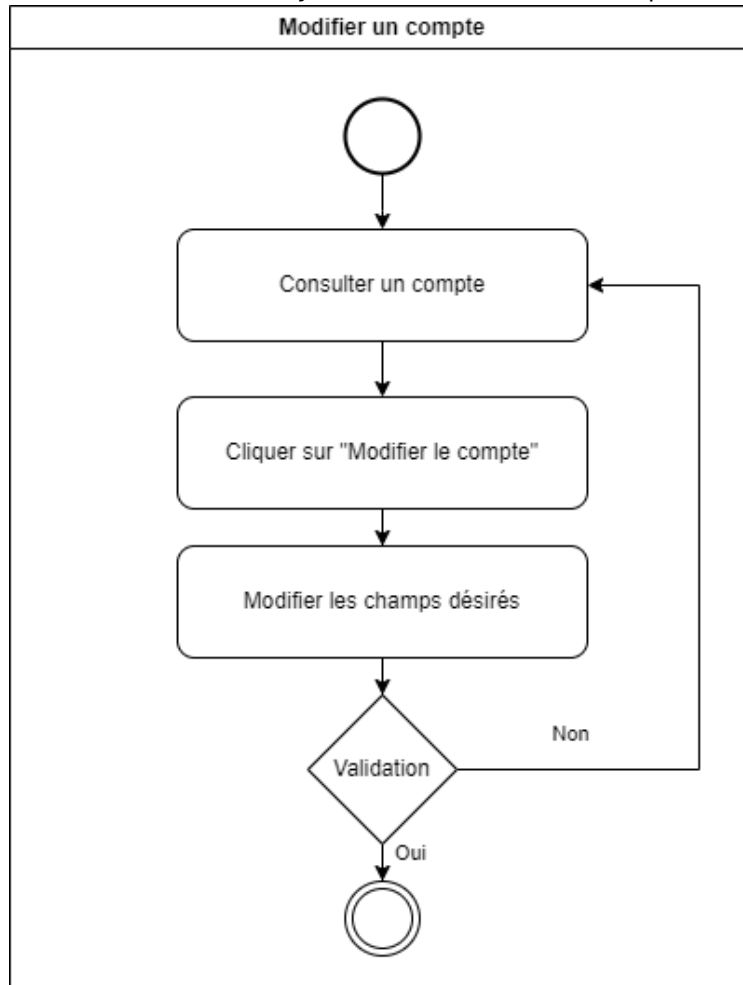
Priorité : Obligatoire (Must Have)

- Modifier un compte

Nom de la fonctionnalité : Modifier un compte

Utilisateur : Administrateur

Description de la fonctionnalité : Sur la page du compte utilisateur, permet à l'administrateur de modifier les informations du compte, tels que l'e-mail, le rôle, l'entité à laquelle l'utilisateur est assujetti mais aussi le mot de passe si besoin



Règles de Gestion : L'email devra être unique et ne pas déjà exister dans le système. Empêcher les injections malveillantes.

Autres contraintes : Vérifier la validité de l'e-mail, toutes les données doivent être renseignées, le compte doit avoir au minimum un rôle. Le mot de passe devra faire minimum 8 caractères avec au moins un chiffre.

Entrée : Donnée modifiée

Sortie : Réussite ou échec

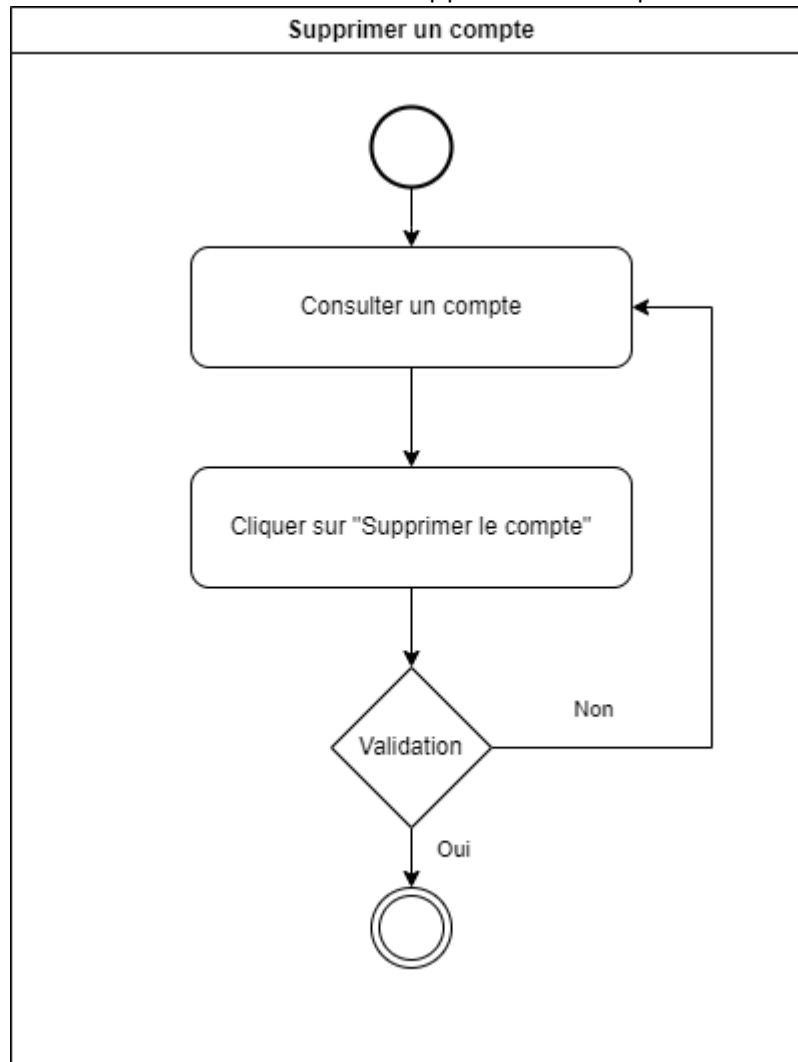
Priorité : Obligatoire (Must Have)

- Supprimer un compte

Nom de la fonctionnalité : Supprimer un compte

Utilisateur : Administrateur

Description de la fonctionnalité : Sur la page du compte d'un utilisateur, permet à l'administrateur de supprimer le compte.



Règles de Gestion :

Autres contraintes :

Entrée :

Sortie : Réussite ou échec

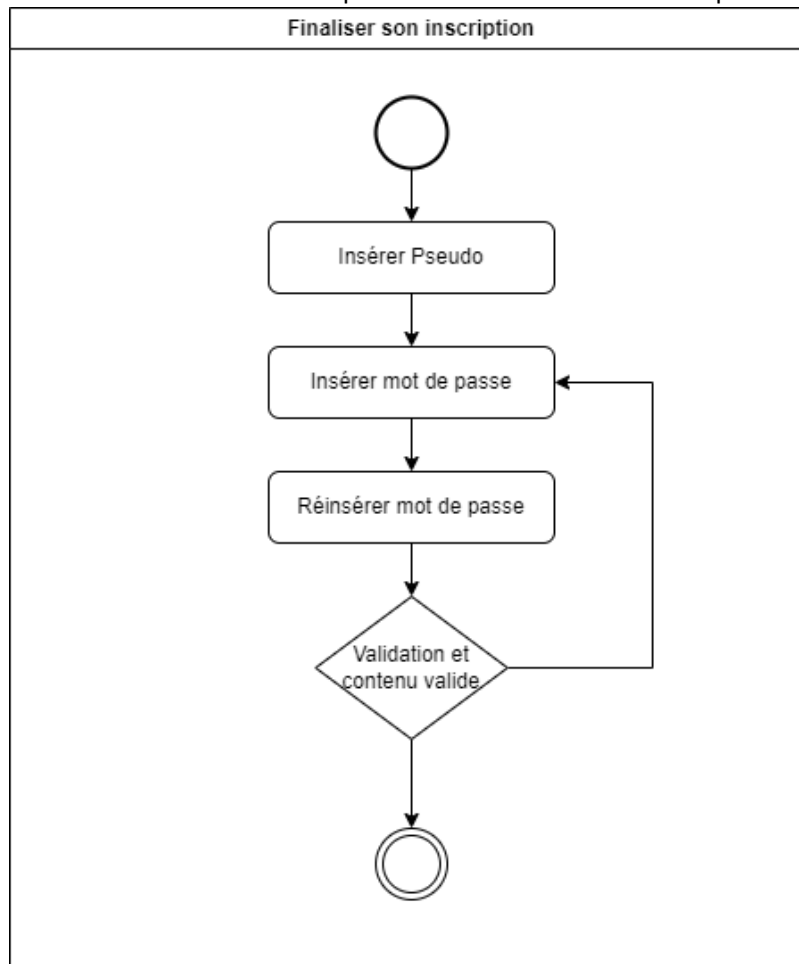
Priorité : Obligatoire (Must Have)

- Finaliser inscription

Nom de la fonctionnalité : Finaliser son inscription

Utilisateur : Utilisateur non-inscrit

Description de la fonctionnalité : Permet à l'utilisateur non-inscrit qui vient de recevoir le mail de finalisation d'inscription de finaliser son inscription.



Règles de Gestion : Le pseudo devra être unique et ne pas déjà exister dans le système.

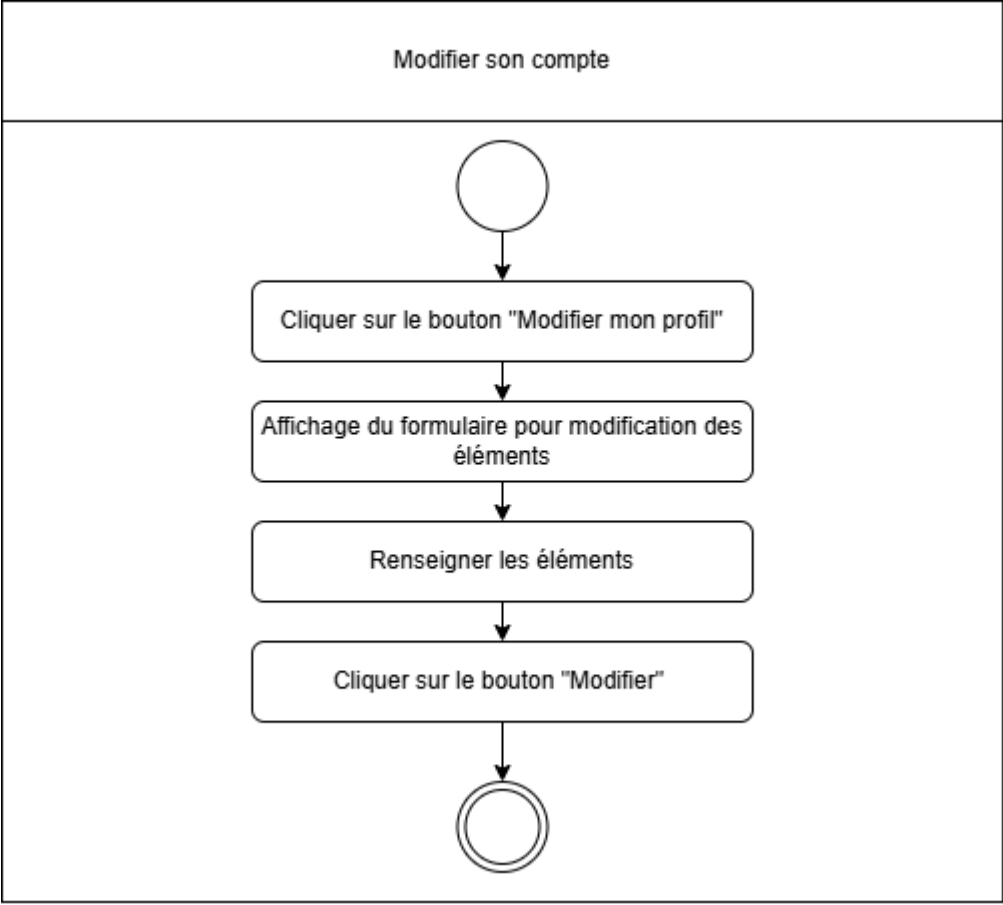
Autres contraintes : Toutes les données doivent être renseignées. Le mot de passe devra faire minimum 8 caractères avec au moins un chiffre.

Entrée : Pseudo et mot de passe.

Sortie : Réussite ou échec

Priorité : Obligatoire (Must Have)

- Modifier son compte

Nom de la fonctionnalité : Modifier son compte
Utilisateur : Utilisateur / Administrateur
<p>Description de la fonctionnalité : Cette fonctionnalité permet de modifier sa page de compte. Les champs modifiables seront :</p> <ul style="list-style-type: none"> • Photo de profil • Pseudo
<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>Modifier son compte</p>  <pre> graph TD Start(()) --> Step1[Cliquer sur le bouton "Modifier mon profil"] Step1 --> Step2[Affichage du formulaire pour modification des éléments] Step2 --> Step3[Renseigner les éléments] Step3 --> Step4[Cliquer sur le bouton "Modifier"] Step4 --> End((())) </pre> </div>
Règles de Gestion :
Autres contraintes :
Entrée : Nouveaux éléments à modifier
Sortie : Redirection à la page profil
Priorité : Obligatoire (Must Have)

Gestion des authentifications

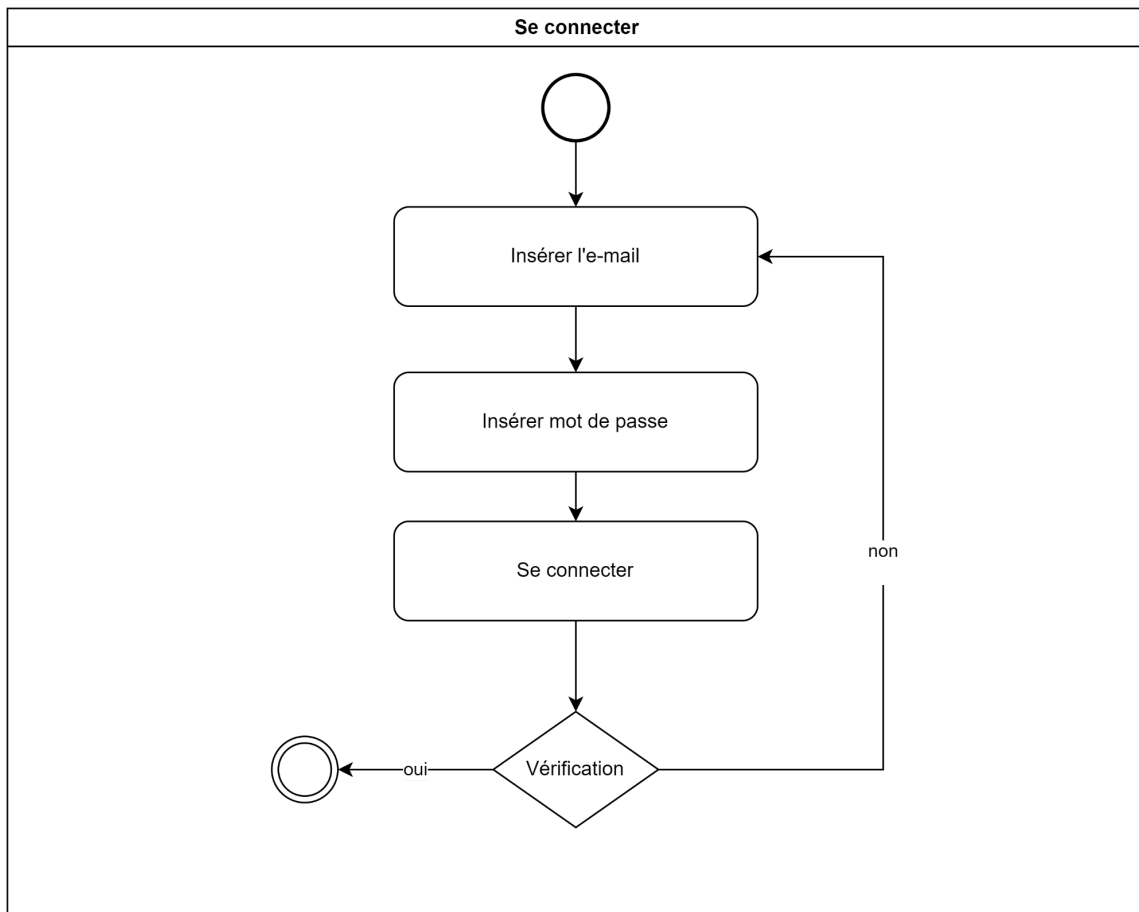
- Se connecter

Nom de la fonctionnalité : Se connecter

Utilisateur : Utilisateur, Administrateur

Description de la fonctionnalité :

Tout utilisateur de l'application devra se connecter.



Règles de gestion : L'e-mail devra respecter le format 'e-mail'
Le mot de passe devra faire au minimum 8 caractères dont 1 chiffre.

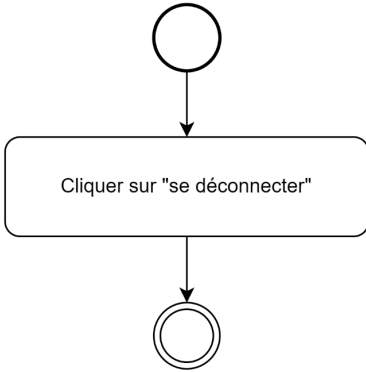
Autres contraintes :

Entrée : E-mail
Mot de passe.

Sortie : Réussite ou échec

Priorité : Obligatoire (Must Have)

- Se déconnecter

Nom de la fonctionnalité : Se déconnecter
Utilisateur : Utilisateur, Administrateur
Description de la fonctionnalité : <p>Cette fonctionnalité permet à tout utilisateur de l'application connecté de se déconnecter.</p> <div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>Se déconnecter</p>  <pre> graph TD Start(()) --> Process[Cliquer sur "se déconnecter"] Process --> End((())) </pre> </div>
Règles de gestion :
Autres contraintes :
Entrée :
Sortie : Réussite ou échec
Priorité : Obligatoire (Must Have)

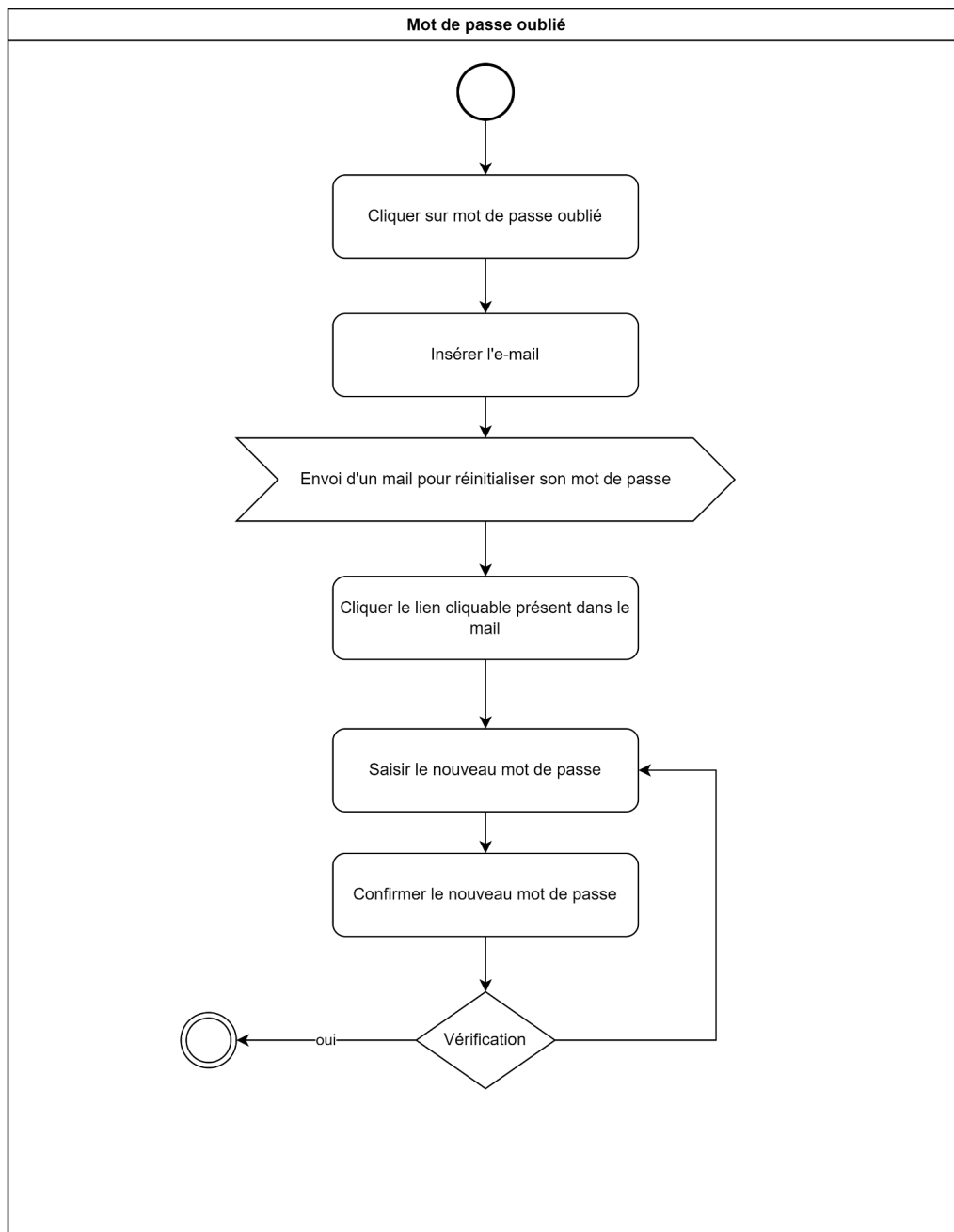
- Changer son mot de passe oublié

Nom de la fonctionnalité : Changer son mot de passe (mot de passe oublié)

Utilisateur : Utilisateur, Administrateur

Description de la fonctionnalité :

Les utilisateurs et administrateurs auront la possibilité de réinitialiser leurs mot de passe en passant par l'adresse mail.



Règles de gestion : Le nouveau mot de passe devra faire au minimum 8 caractères dont 1 chiffre.

Autres contraintes : Le nouveau mot de passe et sa confirmation doivent être les mêmes.

Entrée : Nouveau mot de passe, confirmation du nouveau mot de passe.

Sortie : Réussite ou échec.

Priorité : Bien de l'avoir (Could Have)

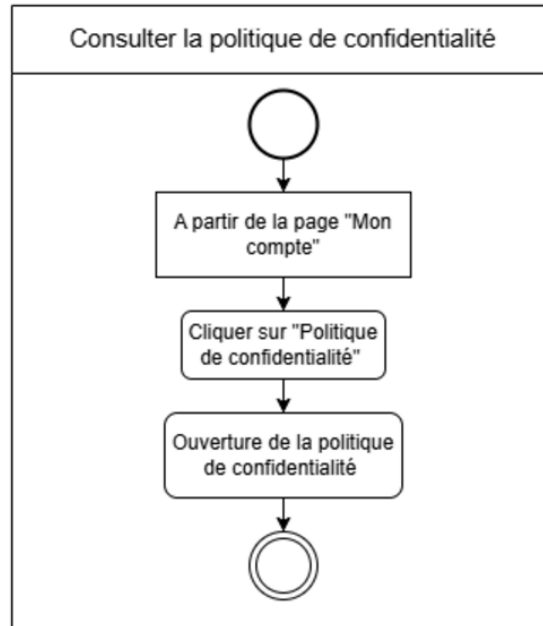
RGPD

- Consulter la politique de confidentialité

Nom de la fonctionnalité : Consulter la politique de confidentialité

Utilisateur : Utilisateur

Description de la fonctionnalité : Un utilisateur pourra accéder à la politique de confidentialité de l'application à partir de la page de son compte



Règles de Gestion :

Autres contraintes :

Entrée :

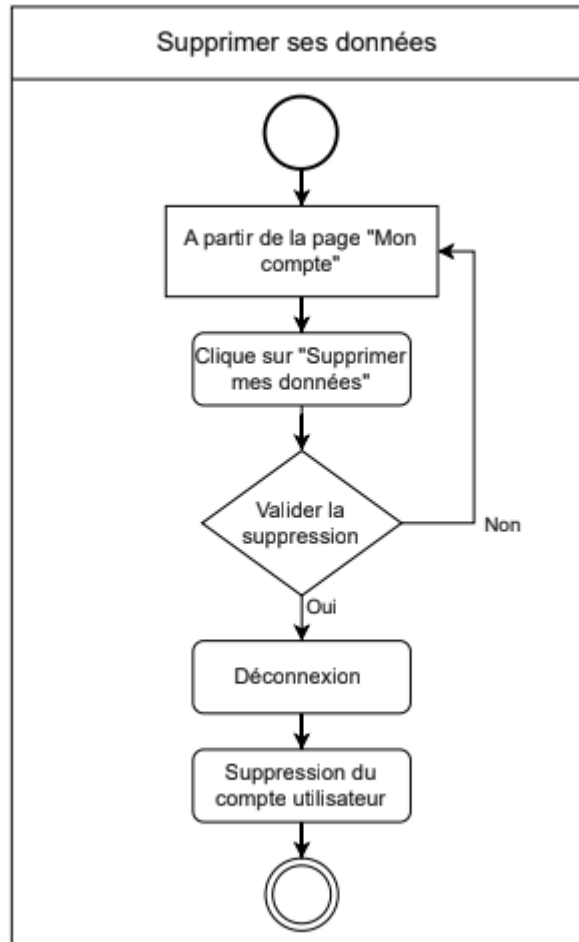
Sortie : Affichage de la politique de confidentialité
Priorité : Ce serait bien (Could Have)

- Supprimer ses données

Nom de la fonctionnalité : Supprimer ses données

Utilisateur : Utilisateur

Description de la fonctionnalité : Un utilisateur peut demander à tout moment de supprimer les données qui lui sont associées via la page de son compte utilisateur



Règles de Gestion : La suppression des données entraîne une suppression totale du compte.

Autres contraintes : Un message indiquant qu'une déconnexion apparaît si l'utilisateur valide la suppression de ses données et qu'aucun retour en arrière ne sera possible

Entrée :

Sortie : Suppression des données et du compte utilisateur

Priorité : Ce serait bien (Could Have)

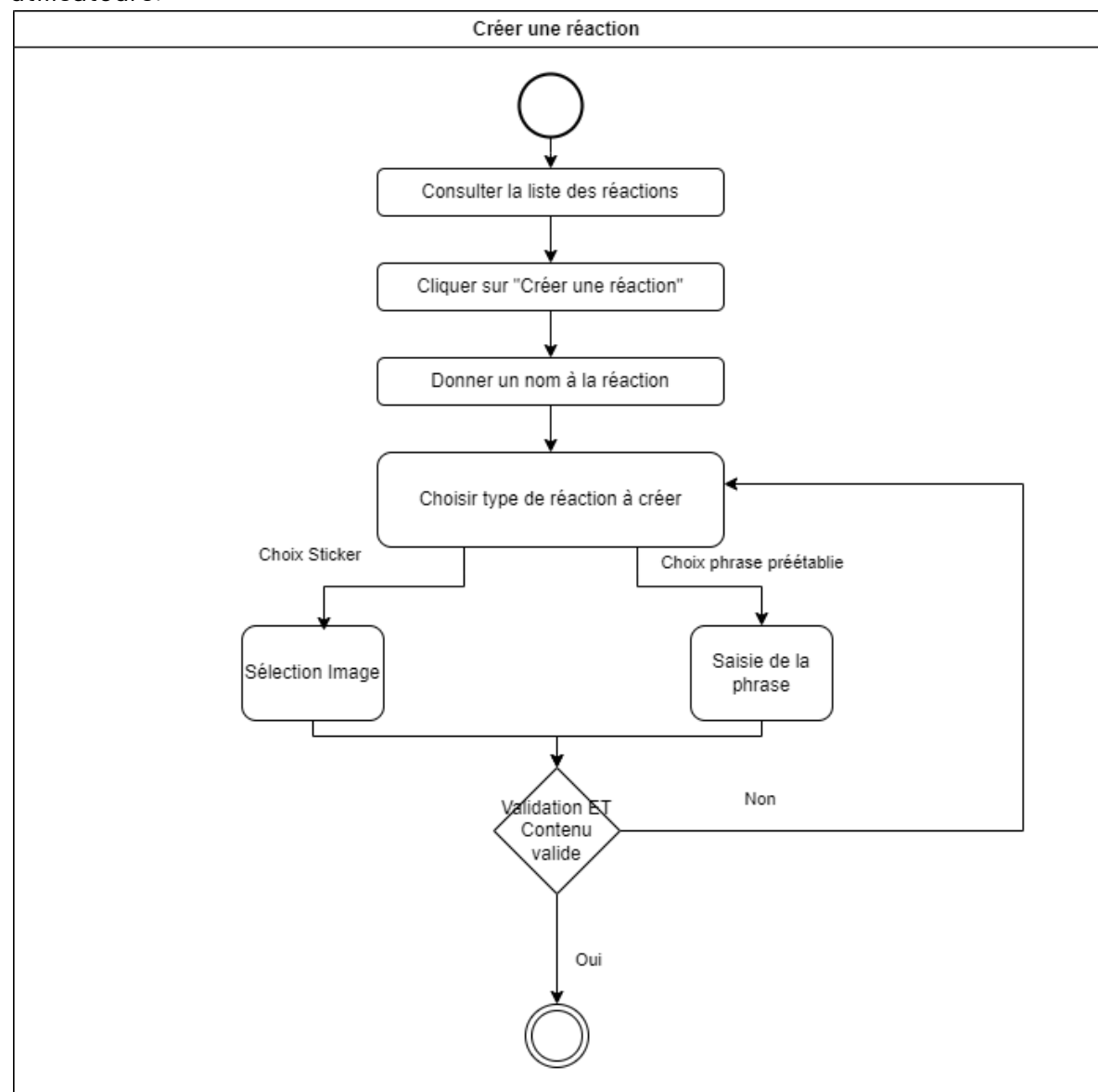
Gestion des réactions

- Créer une réaction

Nom de la fonctionnalité : Créer une réaction

Utilisateur : Administrateur

Description de la fonctionnalité : Permet à l'administrateur de créer une nouvelle réaction (sticker ou phrase préétablie) qui sera disponible à l'usage par les utilisateurs.



Règles de Gestion : Empêcher les injections malveillantes.

Autres contraintes :

Le nom doit respecter une longueur maximale de 85 caractères. La phrase préétablie doit respecter une longueur de 250 caractères. Ils ne doivent pas contenir de termes offensants ou inappropriés.

L'icône doit être de format PNG ou JPEG, avec une taille maximale de 1500 KB et des dimensions de 128*128 px.

Entrée : Entrée textuelle puis image ou entrée textuelle.

Sortie : Réussite ou échec.

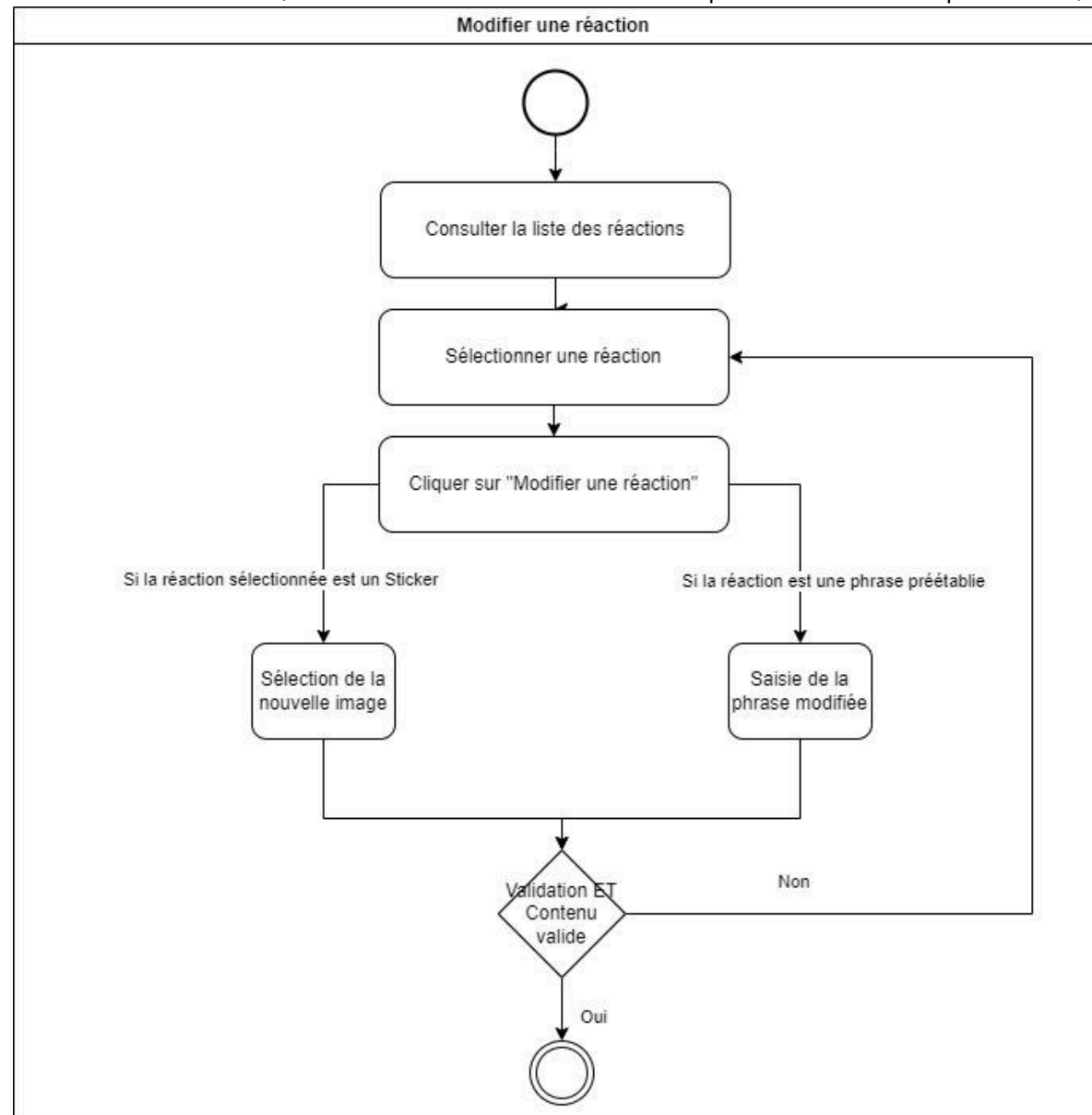
Priorité : Ce serait bien (Could Have)

- Modifier une réaction

Nom de la fonctionnalité : Modifier une réaction

Utilisateur : Administrateur

Description de la fonctionnalité : Permet à l'administrateur de modifier une réaction existante (sticker ou phrase préétablie).



Règles de Gestion : Empêcher les injections malveillantes.

Autres contraintes : Le nom doit respecter une longueur maximale de 85 caractères. La phrase préétablie doit respecter une longueur de 250 caractères. Ils ne doivent pas contenir de termes offensants ou inappropriés. L'icône doit être de format PNG ou JPEG, avec une taille maximale de 1500 KB et des

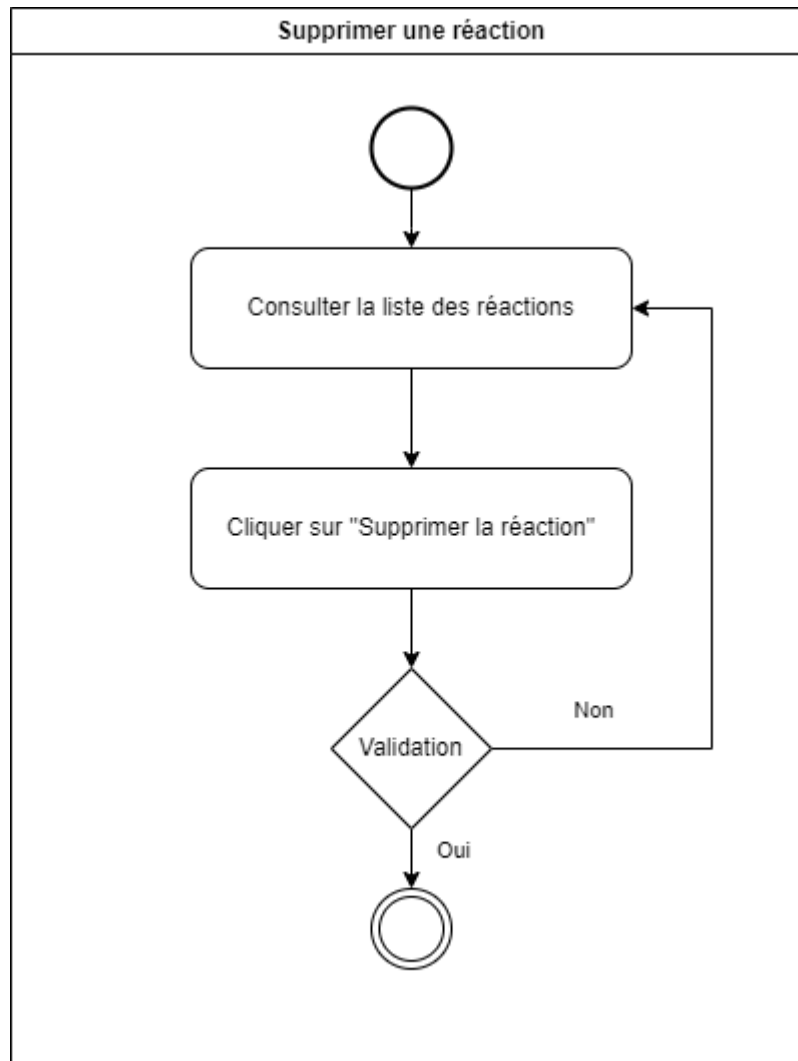
dimensions de 128*128 px.
Entrée : Réaction sélectionnée, nouvelles informations (image ou texte)
Sortie : Réussite ou échec.
Priorité : Ce serait bien (Could Have) -> passe en Must si création de réaction est réalisée

- Supprimer une réaction

Nom de la fonctionnalité : Supprimer une réaction

Utilisateur : Administrateur

Description de la fonctionnalité : Permet à l'administrateur de supprimer une réaction existante.



Règles de Gestion :

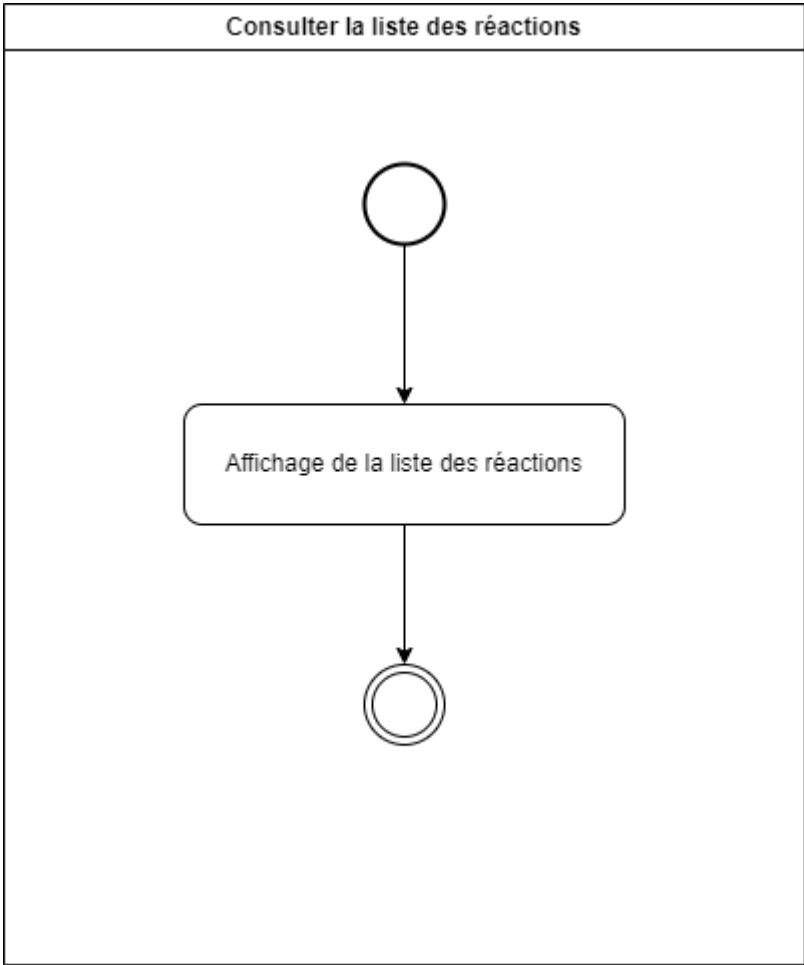
Autres contraintes :

Entrée : Réaction sélectionnée.

Sortie : Réussite ou échec.

Priorité : Ce serait bien (Could Have) -> passe en Must si création de réaction est réalisée

- Consulter liste réaction

Nom de la fonctionnalité : Consulter la liste des réactions
Utilisateur : Administrateur
Description de la fonctionnalité : Permet à l'administrateur d'afficher une liste de toutes les réactions. <div data-bbox="391 521 1197 1482" data-label="Diagram">  <pre> graph TD Start(()) --> Process[Affichage de la liste des réactions] Process --> End((())) </pre> </div>
Règles de Gestion : L'administrateur peut trier la liste par sticker ou phrase préétablie.
Autres contraintes :
Entrée :
Sortie : Liste des réactions, leur titre ET texte ou image.
Priorité : Ce serait bien (Could Have) -> passe en Must si création de réaction est réalisée

Gestion des murs

- Consulter un mur

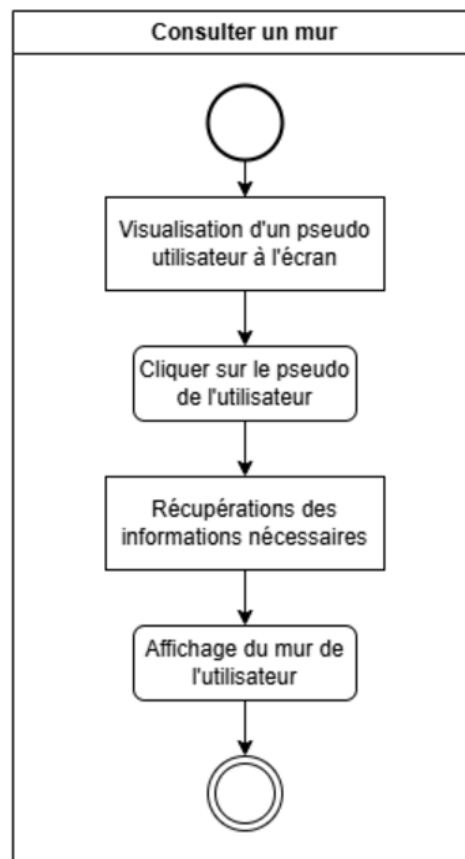
Nom de la fonctionnalité : Consulter un mur

Utilisateur : Utilisateur

Description de la fonctionnalité : Un utilisateur pourra consulter le mur d'un autre utilisateur à partir de n'importe où sur l'application du moment que le pseudo de cet utilisateur est à l'écran.

Sur le mur d'un autre utilisateur les informations suivantes sont disponibles :

- Les informations de profil de l'utilisateur (photo de profil, pseudo, entité, établissement)
- Les statistiques de l'utilisateur
- Le nombre de podiums de l'utilisateur
- Les badges débloqués de l'utilisateur
- Les événements relié à cette utilisateur
- Ma réaction à ces événements
- Les réactions des autres utilisateurs à ces événements



Règles de Gestion :

Autres contraintes :

- Les statistiques s'arrêteront à J-1 car envoi des données du jour une fois par jour à minuit
- Le nombre de podiums est un compteur et si aucun podium n'a été réalisé alors cette donnée ne s'affiche pas
- Les badges débloqués seront visible en couleur
- Seulement les 5 derniers événements de l'utilisateur seront visibles ?

Entrée : Infos utilisateur, statistiques utilisateur, podiums utilisateur, badges utilisateur, événement, réaction

Sortie : Affichage du mur de l'utilisateur

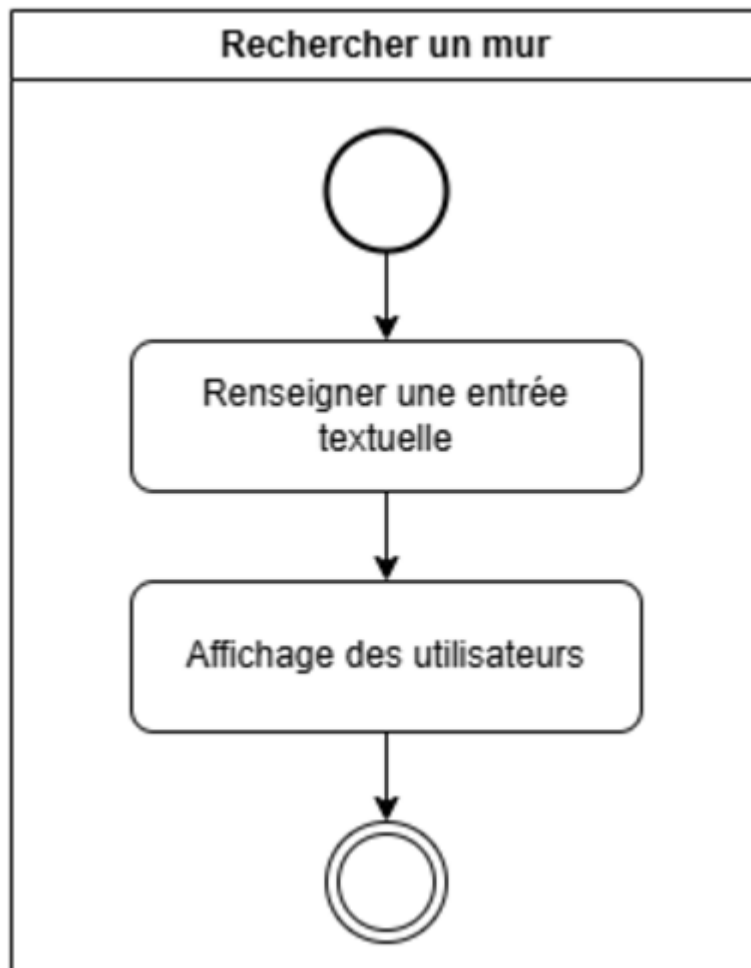
Priorité : Obligatoire (Must Have)

- Rechercher un mur

Nom de la fonctionnalité : Rechercher un mur

Utilisateur : Utilisateur

Description de la fonctionnalité : Permet à tout utilisateur de rechercher le mur d'un autre utilisateur via une entrée dans une barre de recherche textuelle.



Règles de Gestion : Empêcher les injections malveillantes (entrée de certains caractères)

Autres contraintes : L'entrée textuelle sera limité au nombre max de caractère possible pour un pseudo

Entrée : Une chaîne de caractères

Sortie : Une liste des utilisateurs dont une partie du pseudo correspond exactement à la chaîne de caractères d'entrée

Priorité : Obligatoire (Must Have)

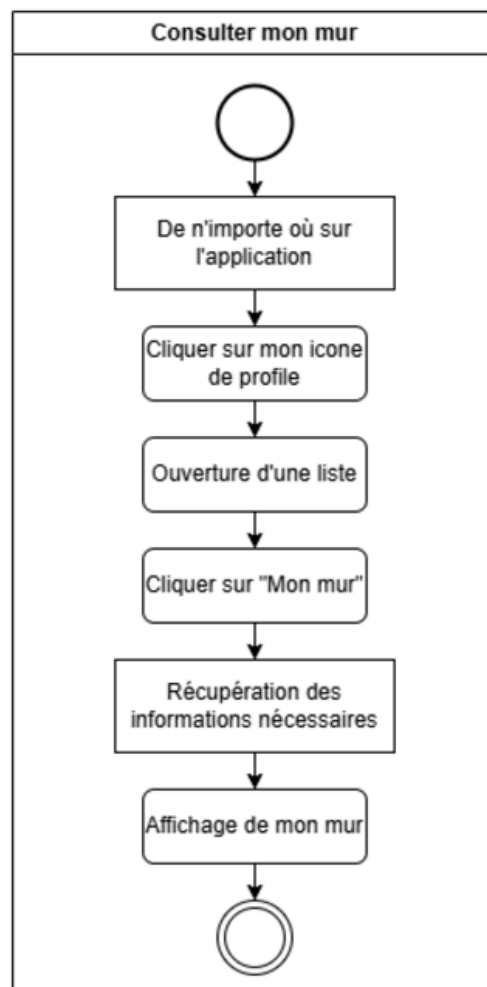
- Consulter mon mur

Nom de la fonctionnalité : Consulter mon mur

Utilisateur : Utilisateur

Description de la fonctionnalité : Un utilisateur pourra consulter son mur depuis n'importe quelle page de l'application. Ce mur possède les informations suivantes :

- Mes informations utilisateur (photo de profil, pseudo, entité, établissement)
- Mon nombre de pas actuel de la journée
- Mes statistiques
- Mon nombre de Podium
- Mes badges débloqués
- Mes badges non débloqués
- Mes événements
- Les réactions des autres utilisateurs à mes événements



Règles de Gestion :

Autres contraintes :

- Le nombre de pas affiché sera les pas actuel du jour récupéré via API
- Les statistiques eux s'arrêteront à J-1 car envoi des données du jour une fois par jour à minuit
- Le nombre de podiums est juste un compteur
- Les badges débloqués seront visible en couleur
- Les badges non débloqués auront leurs visuel final caché
- Seulement les 3 derniers de mes événements seront visibles

Entrée : Informations utilisateur, statistiques utilisateur, nombre de pas du jour, podiums utilisateur, badges utilisateur, réaction

Sortie : Affichage de mon mur

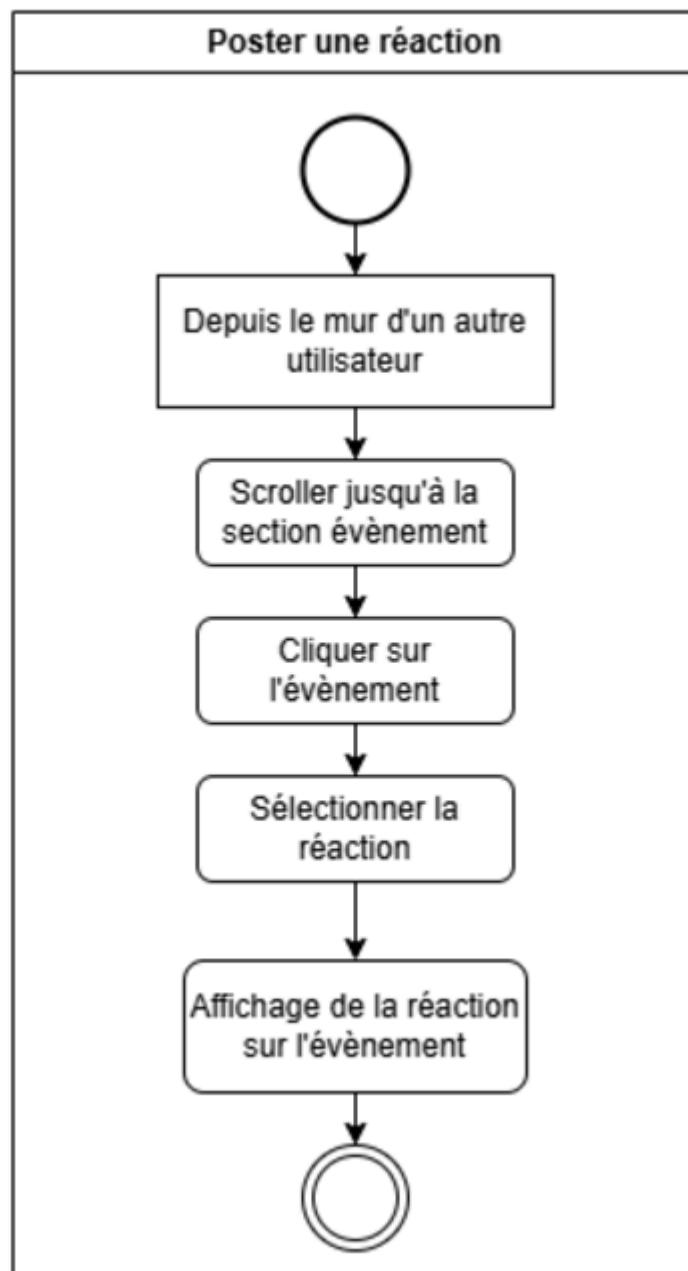
Priorité : Obligatoire (Must Have)

- Poster une réaction

Nom de la fonctionnalité : Poster un réaction

Utilisateur : Utilisateur

Description de la fonctionnalité : Un utilisateur pourra poster une réaction sur n'importe quel événement se trouvant sur un mur d'un autre utilisateur



Règles de Gestion :

Autres contraintes :

- L'utilisateur pourra choisir sa réaction uniquement à travers une liste prédéfinie
- Une seule réaction par événement par utilisateur

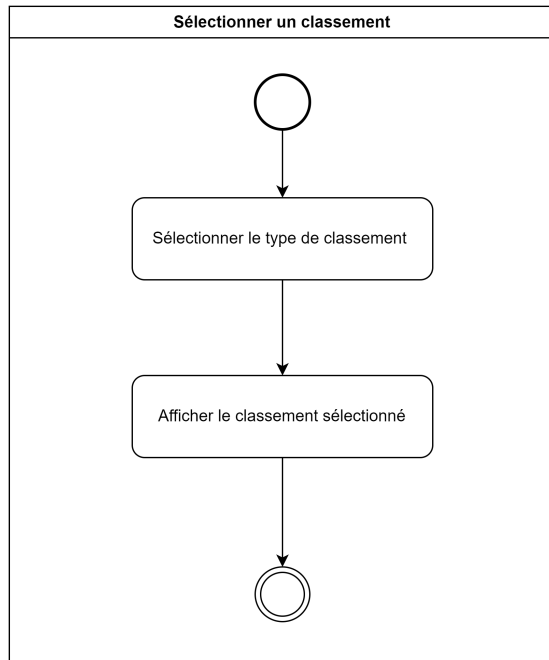
Entrée : Choix de la réaction

Sortie : Affichage de la réaction sur l'événement

Priorité : Bien de l'avoir (Could Have)

Gestion des classements

- Sélectionner un classement

Nom de la fonctionnalité : Sélectionner un classement
Utilisateur : Utilisateur
<p>Description de la fonctionnalité :</p> <p>Les utilisateurs auront accès à une liste déroulante proposant les différents types de classement :</p> <ul style="list-style-type: none"> ● Le classement dans son association ● Le classement des universités entre-elles ● Le classement des associations entre-elles ● Le classement dans son université ● Le classement général de tous les membres de MC <div data-bbox="517 846 1067 1503" data-label="Diagram">  <pre> graph TD Start(()) --> A[Sélectionner le type de classement] A --> B[Afficher le classement sélectionné] B --> End((())) </pre> </div>
<p>Règles de Gestion :</p> <ul style="list-style-type: none"> ● Par défaut, celui-ci affiche le classement général de tous les membres de MC. ● Les classements durent 1 mois, après cela ils doivent être consultés via l'historique et seront réinitialisés.
Autres contraintes :
Entrée : Le type de classement sélectionné
Sortie : Un graphique représentant le classement, la liste des participants ainsi que le nombre de pas.
Priorité : Obligatoire (Must Have)

- Consulter l'historique des classements

Nom de la fonctionnalité : Consulter l'historique des classements

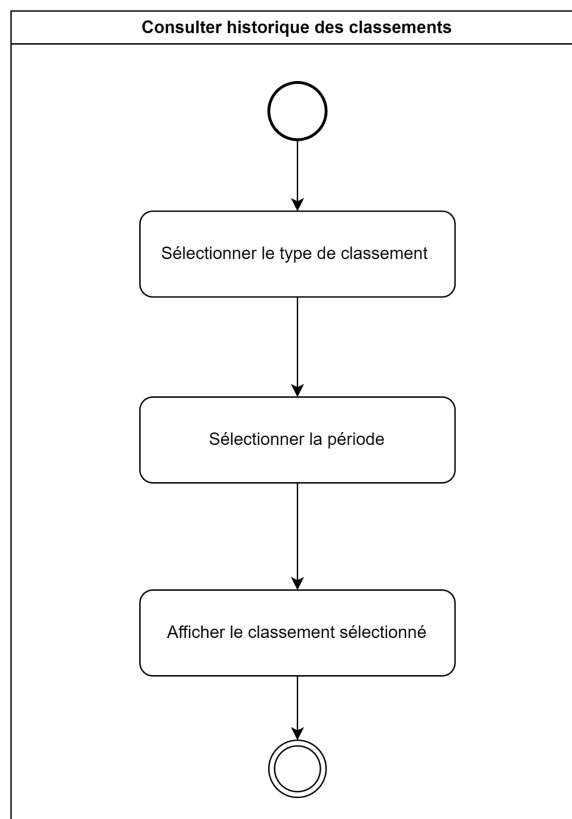
Utilisateur : Utilisateur

Description de la fonctionnalité :

Les utilisateurs auront accès à une liste déroulante proposant les différents types de classement :

- Le classement dans son association
- Le classement des universités entre-elles
- Le classement des associations entre-elles
- Le classement dans son université
- Le classement général de tous les membres de MC

et devront sélectionner la période du classement.



Règles de Gestion :

Autres contraintes : La période doit être antérieure au mois en cours.

Entrée : Le type de classement sélectionné, la période du classement.

Sortie : Un graphique représentant le classement, la liste des participants ainsi que le nombre de pas, et la période de celui-ci.

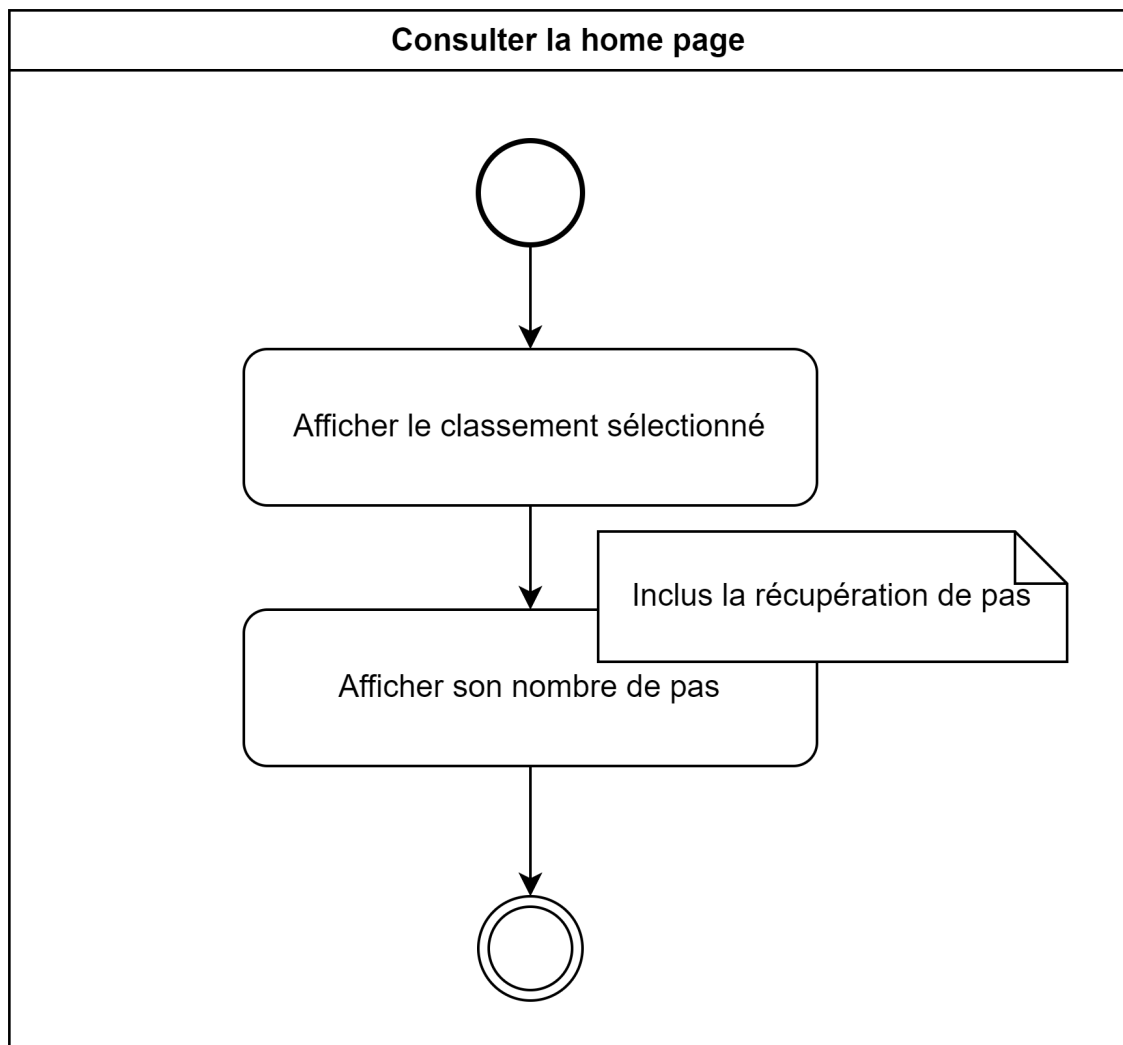
- Consulter la home page

Nom de la fonctionnalité : Consulter la home page

Utilisateur : Utilisateur

Description de la fonctionnalité :

Les utilisateurs auront accès aux classements qu'ils souhaitent consulter ainsi qu'à leurs nombres de pas quotidiens.



Règles de gestion : Pour afficher le nombre de pas quotidien, celui-ci devra être récupéré en amont.

Autres contraintes : On ne récupère pas le nombre de pas des administrateurs.

Entrée : Le type de classement sélectionné.

Sortie : Un graphique représentant le classement, la liste des participants ainsi que

le nombre de pas.
Le nombre de pas effectués .

Priorité : Obligatoire (Must Have)

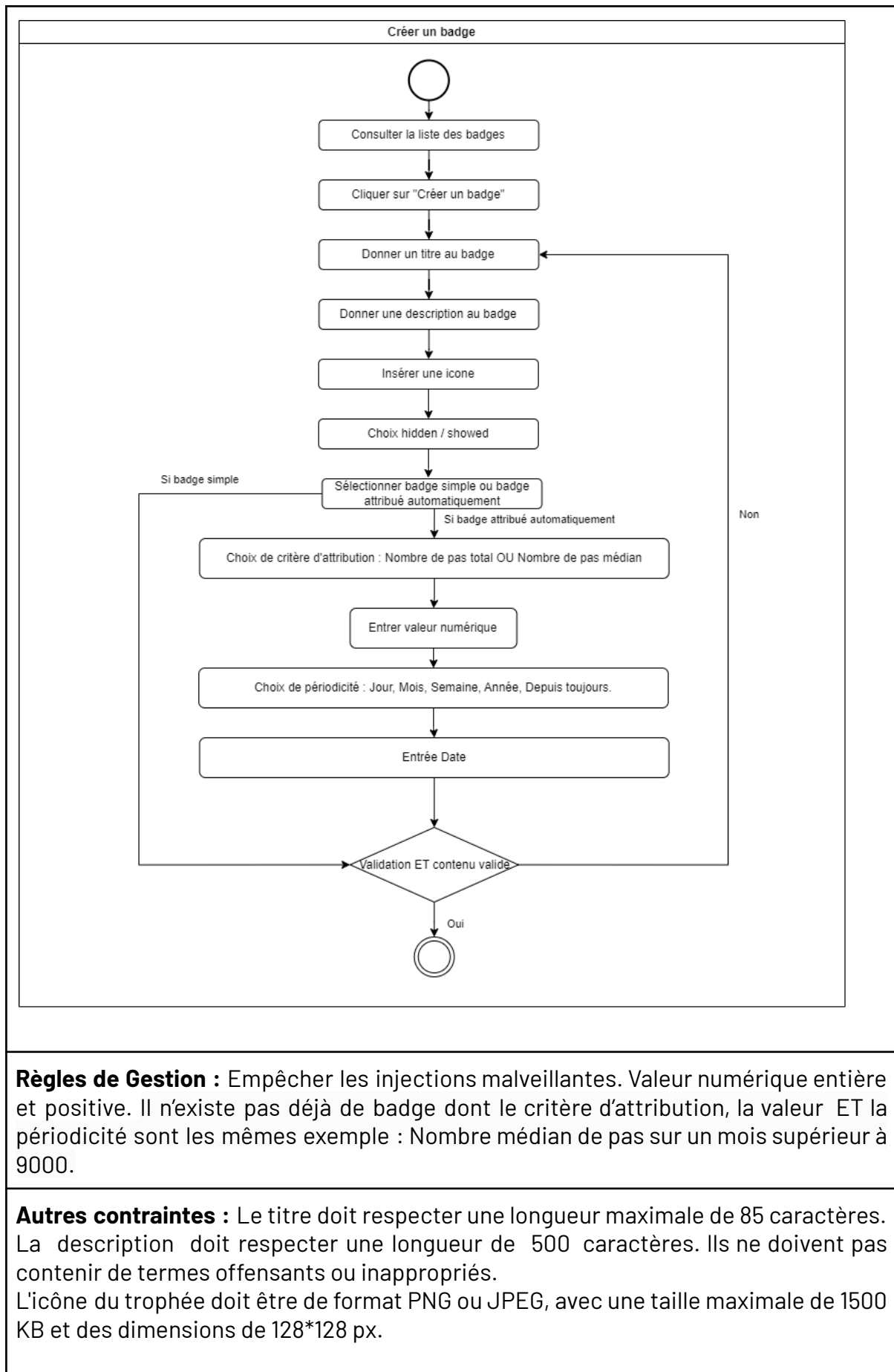
Gestion des badges

- Créer un badge

Nom de la fonctionnalité : Créer un badge

Utilisateur : Administrateur

Description de la fonctionnalité : Permet à l'administrateur de créer un badge pour récompenser les joueurs en fonction de leurs performances (nombre de pas, performances exceptionnelles, etc.).



Règles de Gestion : Empêcher les injections malveillantes. Valeur numérique entière et positive. Il n'existe pas déjà de badge dont le critère d'attribution, la valeur ET la périodicité sont les mêmes exemple : Nombre médian de pas sur un mois supérieur à 9000.

Autres contraintes : Le titre doit respecter une longueur maximale de 85 caractères. La description doit respecter une longueur de 500 caractères. Ils ne doivent pas contenir de termes offensants ou inappropriés. L'icône du trophée doit être de format PNG ou JPEG, avec une taille maximale de 1500 KB et des dimensions de 128*128 px.

Entrée : Titre, explication, icone, hidden/showed, critère (nombre de pas ou nombre médian), nombre entier positif, périodicité, date de début OU Titre, explication, icone, hidden/showed.

Sortie : Réussite ou échec.

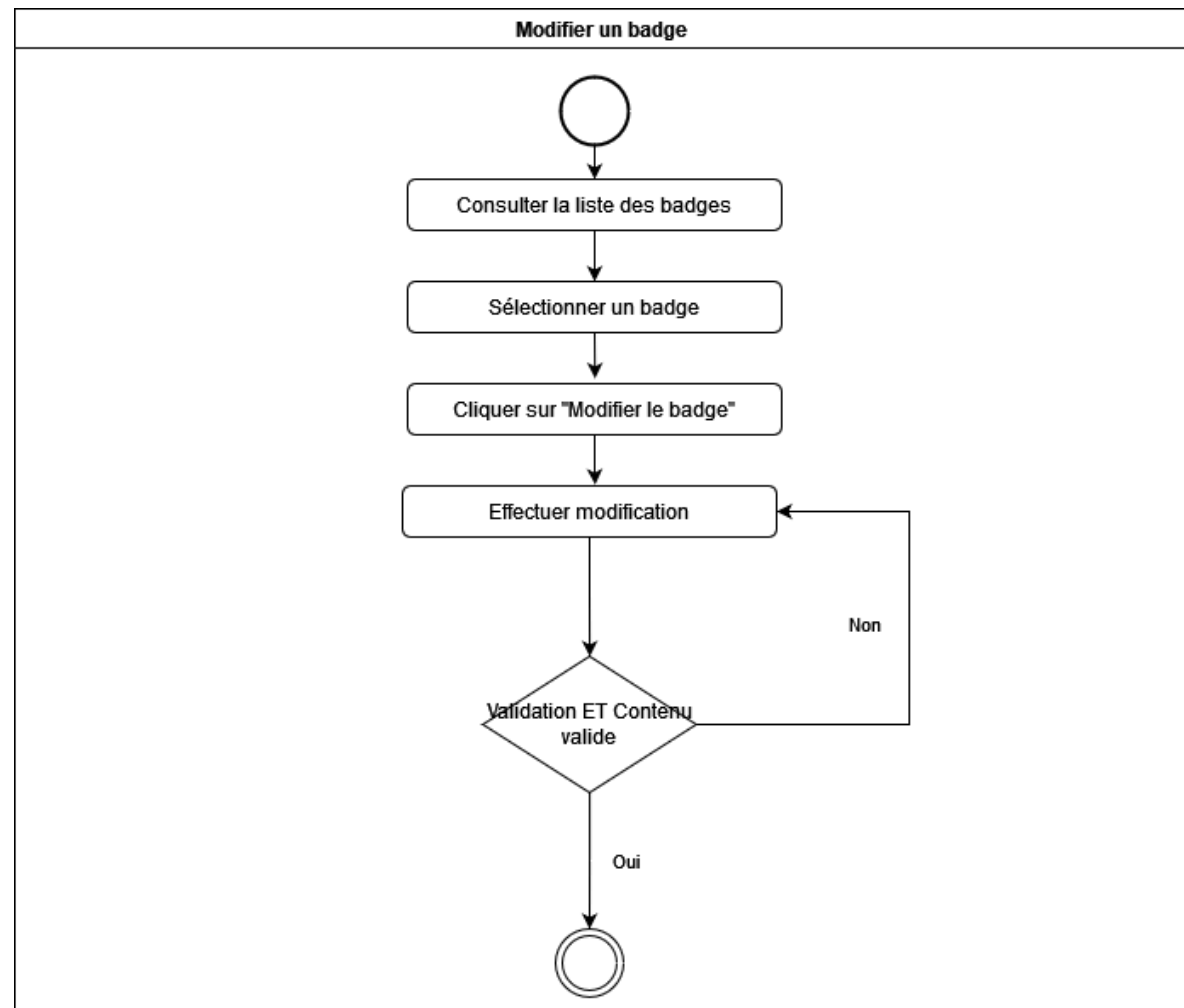
Priorité : Important (Should Have)

- Modifier un badge

Nom de la fonctionnalité : Modifier un badge

Utilisateur : Administrateur

Description de la fonctionnalité : Permet à l'administrateur de modifier un badge existant, y compris son titre, son explication, ses critères de remise, sa périodicité, et son icône.



Règles de Gestion : Empêcher les injections malveillantes. Valeur numérique entrée entière et positive. Il n'existe pas déjà de badge dont le critère d'attribution, la valeur ET la périodicité sont les mêmes exemple : Nombre de pas sur l'année supérieur à 3 500 000.

Autres contraintes : Le titre doit respecter une longueur maximale de 85 caractères. La description doit respecter une longueur de 500 caractères. Ils ne doivent pas contenir de termes offensants ou inappropriés. L'icône du trophée doit être de format PNG ou JPEG, avec une taille maximale de 1500

KB et des dimensions de 128*128 px.

Entrée : Badge sélectionné, nouvelles informations (titre, critère, icône, hidden/showed OU titre, critère, périodicité, nombre entier positif, date de début, icône, hidden/showed).

Sortie : Réussite ou échec.

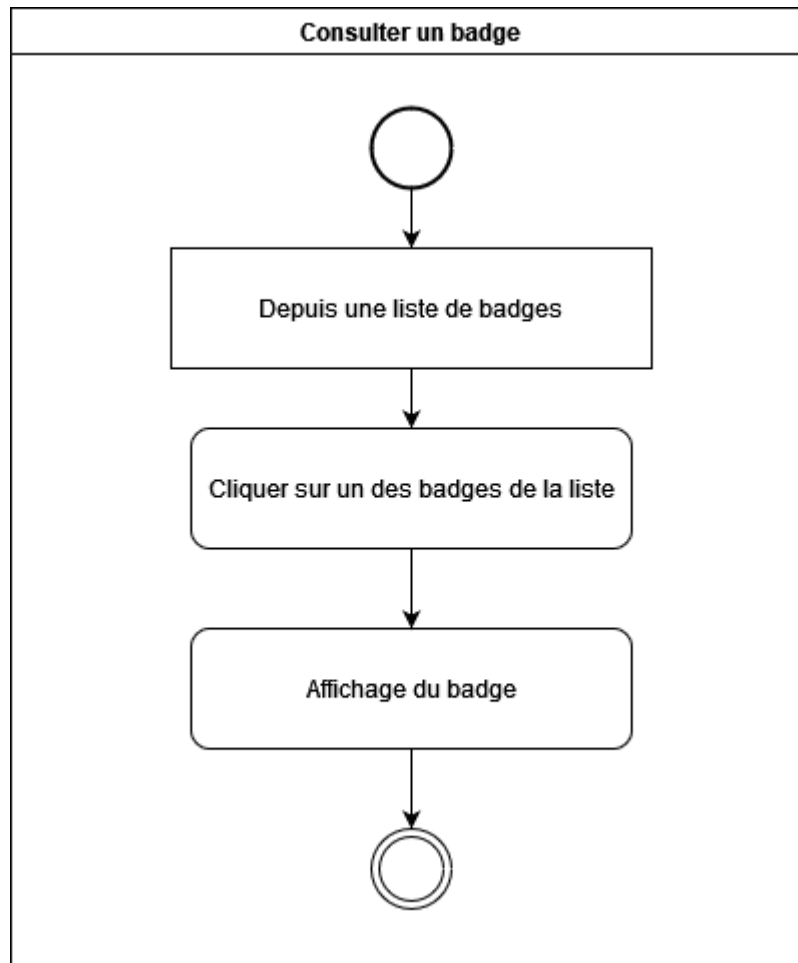
Priorité : Important (Should Have) -> passe en Must si "Créer un badge" est réalisée

- Consulter un badge Administrateur

Nom de la fonctionnalité : Consulter un badge Administrateur

Utilisateur : Administrateur

Description de la fonctionnalité : Permet à l'administrateur de consulter un badge depuis la liste des badges.



Règles de Gestion :

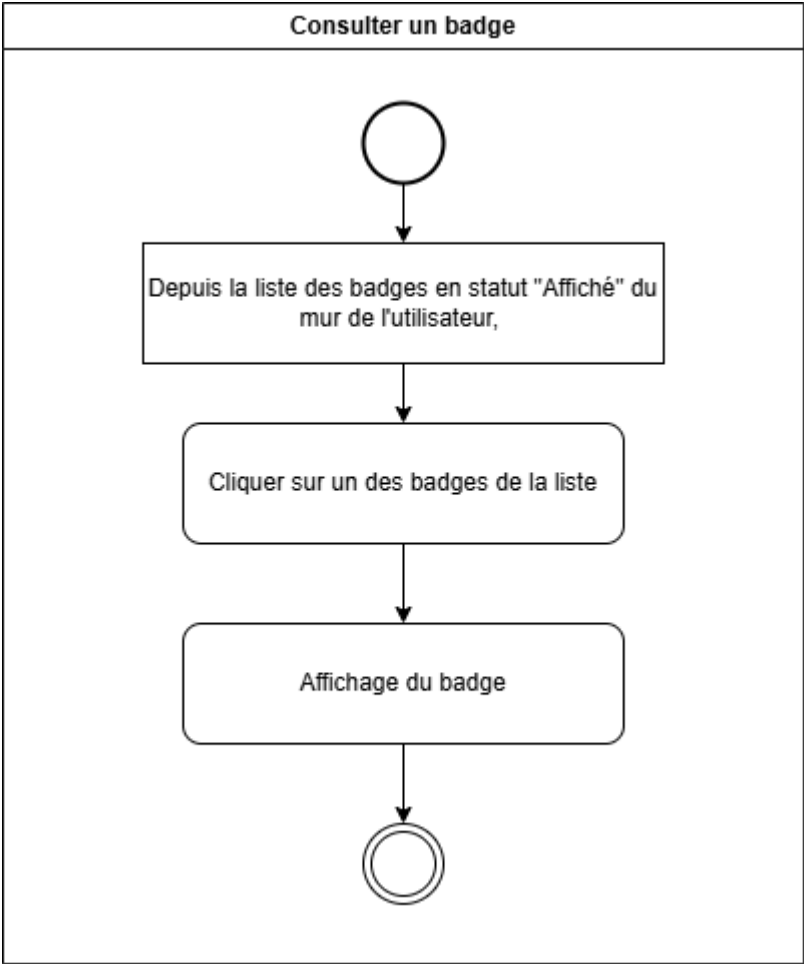
Autres contraintes :

Entrée :

Sortie : Titre, explication, critère d'attribution, nombre, périodicité, icône du badge

Priorité : Important (Should Have) -> passe en Must si "Créer un badge" est réalisée

- Consulter un badge Utilisateur

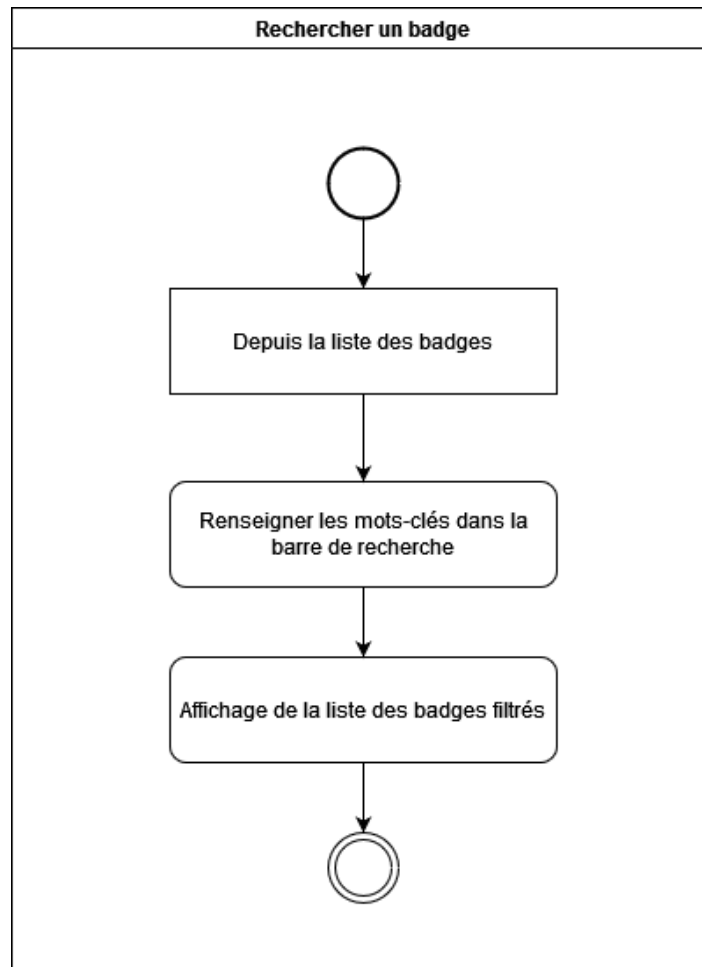
Nom de la fonctionnalité : Consulter un badge Utilisateur
Utilisateur : Utilisateur
<p>Description de la fonctionnalité : Permet à l'utilisateur de consulter un badge depuis son mur.</p> <div data-bbox="392 560 1198 1520">  <pre> graph TD Start(()) --> Context[Depuis la liste des badges en statut "Affiché" du mur de l'utilisateur,] Context --> Action1(Cliquer sur un des badges de la liste) Action1 --> Action2(Affichage du badge) Action2 --> End((())) </pre> </div>
Règles de Gestion : Vérifier que les badges liés à l'utilisateur soient en statut "Affiché"
Autres contraintes :
Entrée :
Sortie : Titre, explication, critère d'attribution, nombre, périodicité, icône du badge
Priorité : Important (Should Have)

- Rechercher un badge

Nom de la fonctionnalité : Rechercher un badge

Utilisateur : Administrateur

Description de la fonctionnalité : Permet à l'administrateur de rechercher un badge via une entrée en barre de recherche depuis la liste des badges.



Règles de Gestion : Empêcher les injections malveillantes, permettre de filtrer sur toutes les colonnes de la liste des badges.

Autres contraintes : Empêcher l'entrée de certains caractères.

Entrée : Un ou plusieurs mots-clés.

Sortie : Liste des badges (titre , image, description) filtrée en fonction de l'entrée.

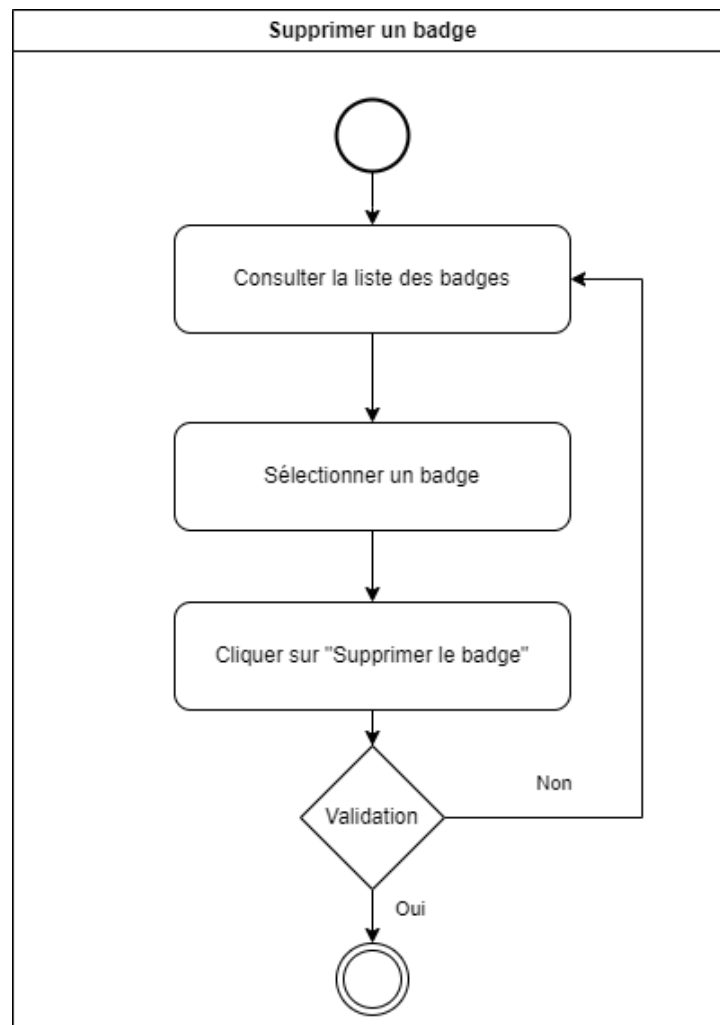
Priorité : Bien de l'avoir (Could Have)

- Supprimer un badge

Nom de la fonctionnalité : Supprimer un badge

Utilisateur : Administrateur

Description de la fonctionnalité : Permet à l'administrateur de supprimer un badge existant.



Règles de Gestion :

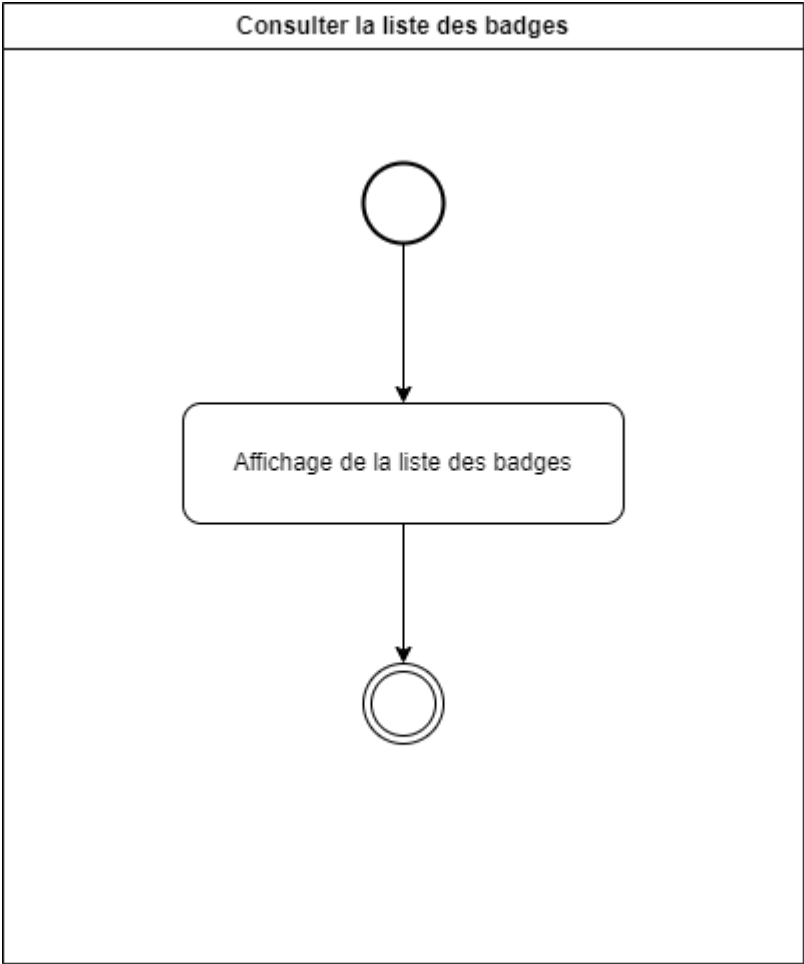
Autres contraintes :

Entrée : Badge sélectionné.

Sortie : Réussite ou échec.

Priorité : Important (Should Have) -> passe en Must si "Créer un badge" est réalisée

- Consulter liste des badges

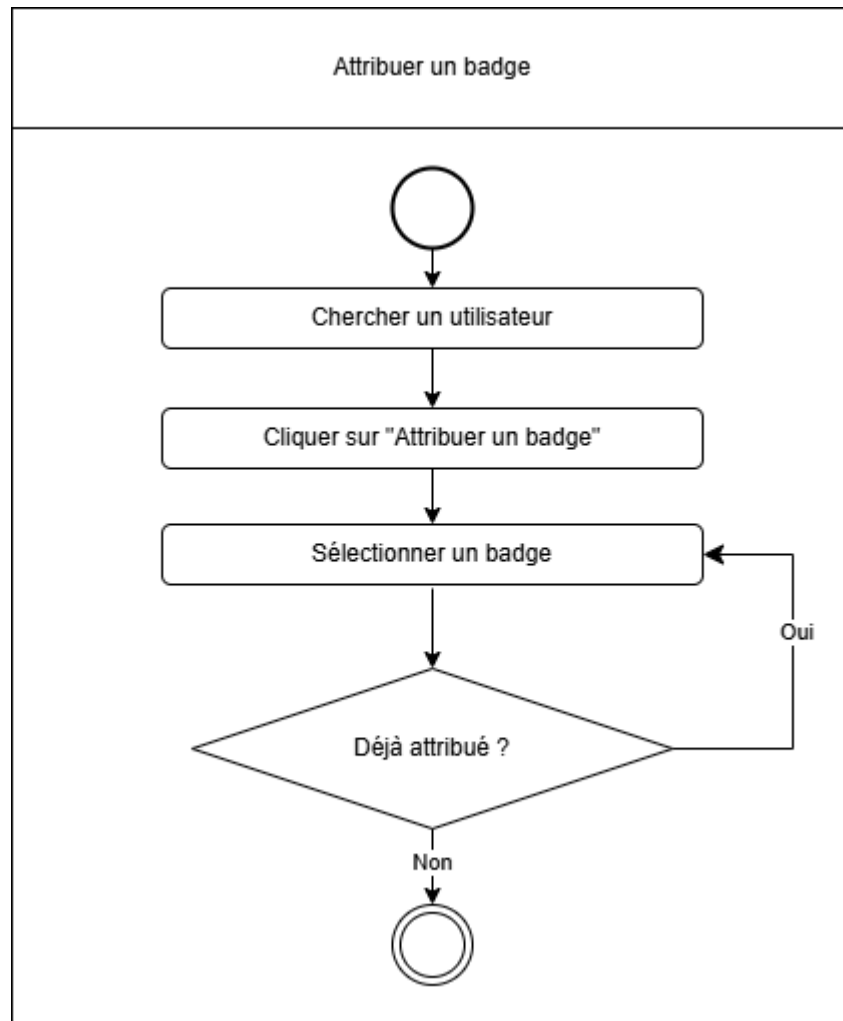
Nom de la fonctionnalité : Consulter la liste des badges
Utilisateur : Administrateur
Description de la fonctionnalité : Permet à l'administrateur d'afficher une liste de tous les badges. <div data-bbox="392 560 1198 1520" data-label="Diagram">  <pre> graph TD Start(()) --> Process[Affichage de la liste des badges] Process --> End((())) </pre> </div>
Règles de Gestion : L'administrateur peut trier la liste par critère ou périodicité.
Autres contraintes :
Entrée :
Sortie : Liste des badges, leur titre, image et description.
Priorité : Important (Should Have) -> passe en Must si "Créer un badge" est réalisée

- Attribuer un badge

Nom de la fonctionnalité : Attribuer Badge

Utilisateur : Administrateur

Description de la fonctionnalité : Cette fonctionnalité permet d'attribuer un badge à un utilisateur depuis la page de consultation des badges



Règles de Gestion : L'identifiant de l'utilisateur et l'identifiant du badge doivent être présent

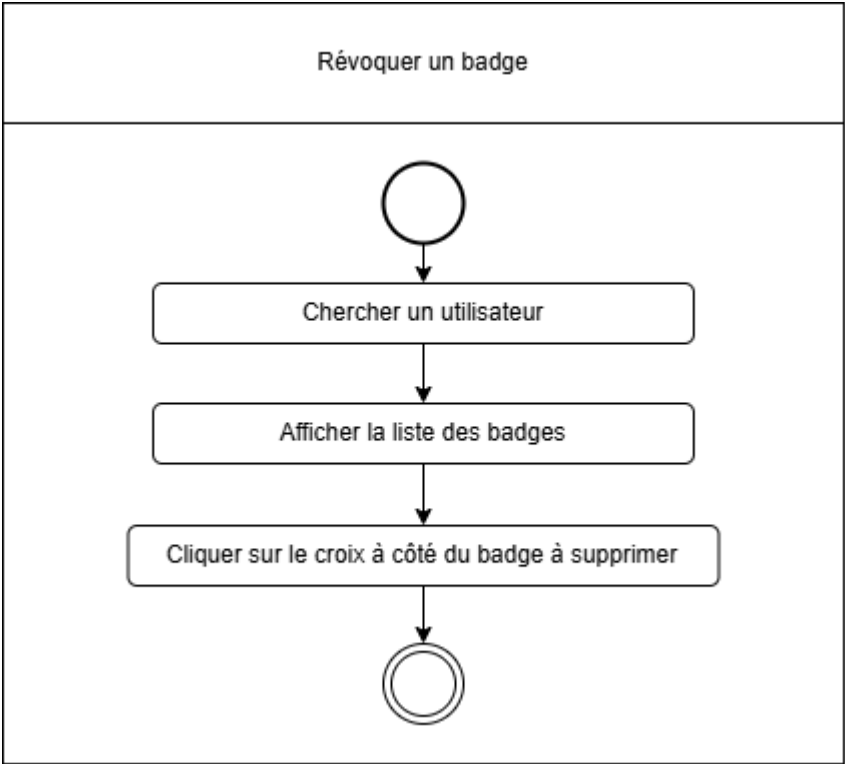
Autres contraintes :

Entrée : Un badge et un utilisateur

Sortie : Réussite ou échec

Priorité : Important (Should Have)

- Révoquer un badge

Nom de la fonctionnalité : Révoquer Badge
Utilisateur : Administrateur
<p>Description de la fonctionnalité : Cette fonctionnalité permet de retirer un badge attribué à un utilisateur</p> <div data-bbox="371 560 1219 1321" data-label="Diagram">  <pre> graph TD Start(()) --> Step1[Chercher un utilisateur] Step1 --> Step2[Afficher la liste des badges] Step2 --> Step3[Cliquer sur le croix à côté du badge à supprimer] Step3 --> End((())) </pre> </div>
Règles de Gestion : L'identifiant de l'utilisateur et l'identifiant du badge doivent être présent
Autres contraintes :
Entrée : Badge de l'utilisateur
Sortie : Réussite ou échec
Priorité : Important (Should Have)

IV. Contraintes

Dans cette section, nous viendrons lister les différentes contraintes que nous devons prendre en compte pour le développement de l'application.

IV.A Contraintes techniques

IV.A.1 Contraintes liées au développement

Voici la liste des contraintes liées au développement prévues :

- Utiliser une technologie capable de faire des déploiements sur Android et IOS.

L'application doit avoir le même code entre la version Android et la version IOS

- Utiliser une technologie bien documentée en ligne pour la reprise du projet si nécessaire.
- Rédiger de la documentation durant le développement et après le développement.

A l'heure actuelle, le framework React native est la solution envisagée.

API permettant d'accéder au nombre de pas et leur accès

	APPLE HEALTH KIT (APPLE)	GOOGLE FIT API (ANDROID)	FITBIT WEB API	SAMSUNG HEALTH HDK
COUT	Gratuit (compter les frais d'inscription au programme développeur)	Gratuit (compter les frais d'inscription au programme développeur)	Gratuit	Gratuit
INSCRIPTION ET CRITÈRES	<ul style="list-style-type: none"> • Inscription programme développeur Apple • L'application doit être liée à la santé ou au fitness • développer la politique de confidentialité sur l'usage des données 	<ul style="list-style-type: none"> • Inscription Google Developer console • Utilisation permissions OAuth 2.0 • Suivre les règles sur la confidentialité et la sécurité des données de Google Play 	<ul style="list-style-type: none"> • Inscription compte développeur Fitbit • Expliquer les données utilisées et pourquoi • Utilisation permissions OAuth 2.0 	<ul style="list-style-type: none"> • Inscription compte développeur Samsung • Expliquer les données utilisées et pourquoi • Utilisation permissions OAuth 2.0
VALIDATION	<ul style="list-style-type: none"> • Validation lors de la soumission de l'application • ~ une semaine 	<ul style="list-style-type: none"> • Validation lors de la soumission de l'application (Play Protect) • ~ une semaine 	<ul style="list-style-type: none"> • Vérification manuelle • ~ une semaine 	<ul style="list-style-type: none"> • Validation lors de la soumission de l'application en plus de la vérification Google • ~ une semaine

ANDROID

Processus de publication sur le playstore android

I. CRÉER UN COMPTE DÉVELOPPEUR GOOGLE

- Frais unique d'inscription : **25 \$**
- Accès à la console Google Play



II. PRÉPARER LA PUBLICATION

- Icône, capture, description claire
- Choix catégorie, balises, restriction d'âge
- Rédaction politiques de confidentialité

III. TESTER ET VÉRIFIER LA CONFORMITÉ

- Tester sur différentes versions d'Android (Changements système de permission, accès API et compatibilité matérielle)
- Respect politiques Google Play (confidentialité, sécurité)



IV. PUBLICATION

- Soumission pour approbation (quelques heures à quelques jours + même process pour les maj)
- Application automatiquement mise sur le Playstore

IOS

Processus de publication sur l'App Store

I. INSCRIPTION AU PROGRAMME DÉVELOPPEUR APPLE

- Frais unique d'inscription : **99 \$**
- Accès à Xcode, testing tools et à l'App Store Connect



II. PRÉPARER LA PUBLICATION AVEC XCODE

- Développement suivant les normes d'Apple avec Xcode
- Compilation sous format IPA
- Préparer fiche d'application :
 - Icône, capture, description claire
 - Choix catégorie, balises, restriction d'âge
 - Rédaction politiques de confidentialité

III. TESTER ET VÉRIFIER LA CONFORMITÉ

- Créer fiche d'application
- Téléverser l'IPA
- Soumettre pour examen



IV. PUBLICATION

- Application automatiquement mise sur l'App Store après validation
- Chaque mise à jour ultérieure sera soumise à un examen similaire

IV.A.2 Contraintes liées à l'infrastructure

MIAGE CONNECTION ne possède ni système d'hébergement des données, ni serveur pour gérer le backend.

L'enveloppe correspond seulement à assez pour accéder aux comptes développeur des stores. C'est pourquoi nous devons trouver une solution permettant d'accueillir le back-end, la base de données ainsi que les API gratuite. Nous nous tournons actuellement vers plusieurs solutions / combo de solutions :

- 1) La solution Firebase (back-end et BDD) avec le plan gratuit "Spark" qui permet :
 - 50 000 lectures par jour
 - 20 000 écriture par jour
 - 20 000 suppressions par jour

Cela nous paraît suffisant pour gérer la masse d'utilisateurs de l'application.

- 2) La solution Supabase pour le back-end avec Firebase en BDD permet aussi de répondre à la problématique en ayant des appels API illimités.
- 3) La solution Vercel pour le back-end avec Supabase en BDD pour la gestion des utilisateurs.

IV.A.3 Contraintes liées à la gestion de projet

Le projet devra être livré avec tous les documents nécessaires à la reprise du projet par un développeur. Tous les mots de passe, procédures particulières et retours d'expérience seront décrits afin d'avoir un livrable le plus complet possible.

IV.B Contraintes d'environnement

IV.B.1 Contraintes humaines

Nous sommes 4 étudiants en alternance, il nous est difficile de prévoir à l'avance le temps que nous pourrions investir dans le projet. Nous estimons en moyenne d'être en mesure de travailler 7h par personne, par semaine.

Ce qui nous fait donc une capacité de 28h à allouer au projet par semaine.

IV.B.2 Contraintes financières

Nous avons un budget de 125 euros pour réaliser le développement, l'hébergement, la publication et la mise en production de l'application sur les deux stores.

IV.B.3 Contraintes juridiques

Conformité au RGPD (Règlement Général sur la Protection des Données)

- Collecte de données :
 - Obtenir le consentement explicite des utilisateurs avant de collecter leurs données personnelles (exemple : données de pas, identifiants de l'utilisateur)
 - Fournir une explication claire et compréhensible sur les finalités de la collecte dans une politique de confidentialité
 - Stockage des données dans l'Espace Économique Européen. Et si utilisation d'hébergeur : doit être conforme aux RGPD
 - Tenue d'un registre des traitements :
 - Type de données collectées
 - Finalités
 - Durées de conservation
 - Mesures de sécurité mises en place
- Minimisation des données :
 - Ne collecter que les données strictement nécessaires pour le fonctionnement de l'application
 - Par exemple : uniquement le nombre de pas ou les données dérivées nécessaires pour le service proposé
- Droit des utilisateurs :
 - Permettre aux utilisateurs d'exercer leurs droits :
 - Accès à leurs données
 - Suppression des données (droit à l'oubli)
 - Portabilité des données (exportation des données)
 - Opposition à la collecte ou au traitement des données
- Durée de conservation :
 - Définir une durée limitée de conservation des données, spécifiée dans la politique de confidentialité
 - Supprimer ou anonymiser les données après cette période
- Sécurisation des données :
 - Chiffrement des données pendant leur transmission (TLS/SSL)
 - Chiffrement des données sensibles sur le serveur. Chiffrement avancé étant donné que les données de pas sont considérées comme des données de santé par le store.

- Mise en place de contrôles d'accès pour les données stockées. + Mise en place d'un journal d'accès aux données
- Politique de confidentialité :
 - Fournir un lien vers une politique de confidentialité au moment de l'installation de l'application
 - Décrire comment les données sont collectées, utilisées, stockées, et partagées dans la politique
- En cas de fuite ou d'accès non autorisé aux données :
 - Informer la CNIL dans les 72 heures suivant la découverte

IV.B.1 Contraintes temporelles

Nous avons 3 milestones universitaires que nous devons respecter :

- Rendre la dernière version en date du cahier des charges pour le 01/12/24 à 23h00
- Rendre la maquette de la solution ainsi qu'un document avec la justification de nos choix ainsi que la justification de notre méthode de conduite de projet pour la phase de développement pour le 22/01/25 à 23h00
- Rendre un bilan sur une prise de recul de notre projet et de notre gestion de projet le 29/06/25 à 23h00

Notre objectif est que ce dernier bilan signe la fin du projet .

IV.B.5 Contraintes d'accessibilités

Voici les contraintes d'accessibilité prévues :

- La solution sera uniquement délivrée en langue française.
- Respect des normes WCAG (Web Content Accessibility Guidelines) pour garantir la lisibilité des textes.
- Nombre d'étapes pour accomplir des tâches fréquentes réduites.
- Messages clairs et concis en cas d'erreur.
- Inclusion d'une section dans la documentation technique pour envisager des adaptations futures liées à l'accessibilité.

V. Identité visuelle

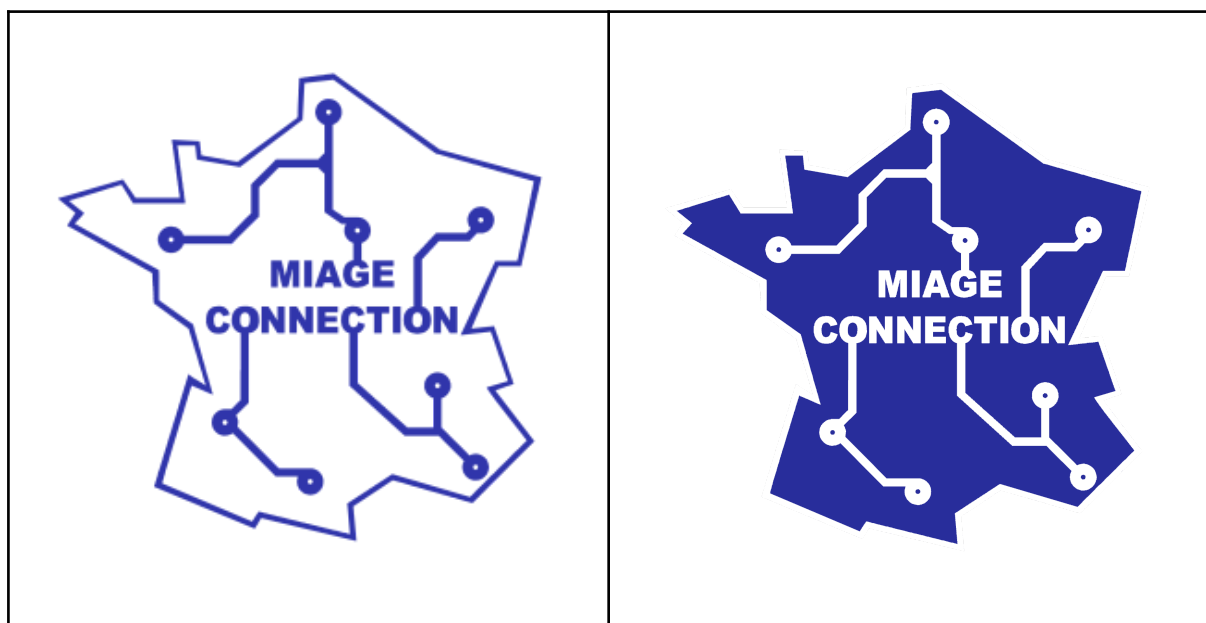
Concernant l'identité visuelle nous avons eu carte blanche, il suffit de produire une application avec une DA cohérente et ergonomique.

Nous allons donc reprendre la couleur du logo MIAGE Connection comme couleur primaire de notre application.

Ce bleu est :  #2b309b

Ensuite, concernant les différents badges, réactions et autres assets nous n'avons pour le moment aucun visuel. Nous créerons des icônes/images génériques qui auront un lien avec de l'activité sportif.

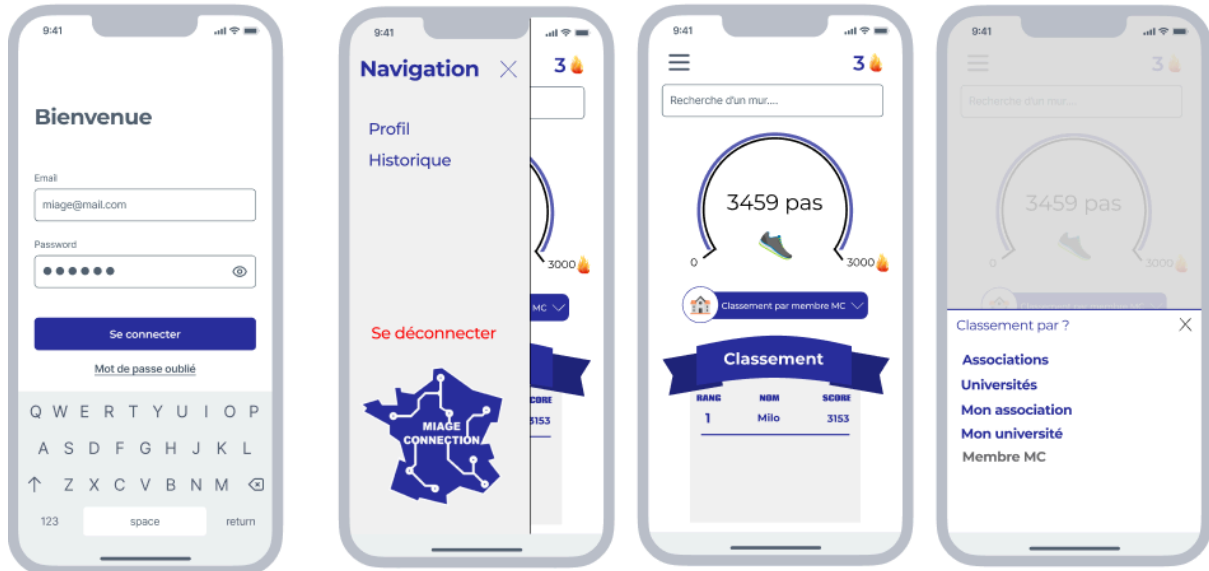
Enfin nous utilisons une technologie qui nous permettra facilement d'avoir une interface agréable pour l'utilisateur.



VI. Wireframe

Voici une évolution de la wireframe présente sur le document "Note de Cadrage". C'est le premier pas vers la maquette de l'application. Une version complète sera livrée prochainement.

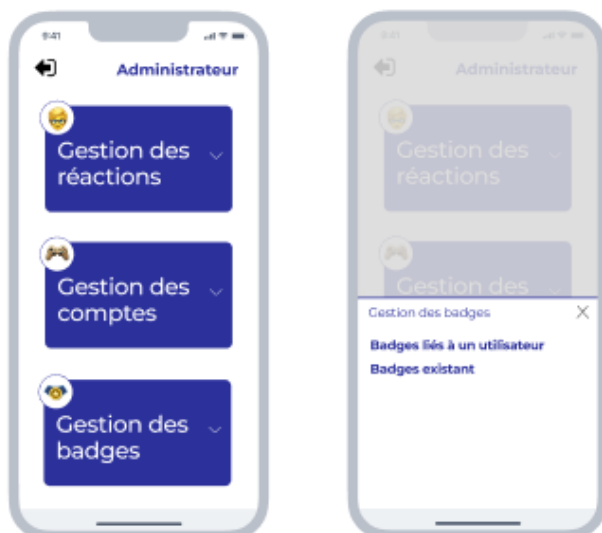
UTILISATEUR



La wireframe de l'utilisateur est composée de plusieurs blocs :

- Le side menu, avec le profil utilisateur et l'historique des classements précédents.
- La home page contient :
 - Le compteur de streak et la recherche des murs
 - Le compteur de pas du jour avec le niveau de progression pour la streak
 - Le classement du moment

ADMIN



La wireframe de l'administrateur est plus minimaliste. Elle est composée de 3 boutons :

- Accéder à la gestion des réactions pour modifier les phrases et stickers.
- La gestion des comptes intègre l'invitation de nouveaux utilisateurs ainsi que la modification de compte utilisateur.
- La gestion des badges où il va pouvoir accéder à une page pour ajouter/retirer un badge à un utilisateur. Il peut aussi accéder à une seconde page afin de créer/modifier/supprimer les badges dans l'application.

Voici un exemple de mur pour un profil.

On peut y retrouver plusieurs blocs :

- Les informations de l'utilisateur, avec les entités rattachées à celui-ci.
- Les événements du compte, composés d'un message contenant le nom du badge ainsi que les réactions des autres utilisateurs.
- La liste des badges de l'utilisateur, avec la possibilité de cliquer dessus afin de voir les informations relatives à celui-ci.
- Les podiums qu'il a accumulés. En fonction des positions de l'utilisateur dans les classements déjà terminés, celui-ci accumule des médailles.
- Les statistiques où sont résumées les données complètes de l'utilisateur.
- Un diagramme linéaire sur l'évolution de l'activité de l'utilisateur.



VII. Risque de cette solution

VII.A Risques post-développement (Run)

Problèmes techniques après la fin du développement

Causes possibles :

- Erreurs de conception ou problématiques non identifiées avant la fin du projet.
- Absence de suivi après la date de fin de projet.

Conséquences : Faible satisfaction et perte d'intérêt pour l'application si des problèmes persistent.

Solutions proposées :

- Faire une structure de projet accessible pour que le projet puisse d'être continué et amélioré par d'autres développeurs

Mises à jour futures obligatoires

- Les applications mobiles peuvent nécessiter des mises à jour régulières pour rester fonctionnelles, notamment dans des cas où il y a de la gestion de données personnelles.
- Nécessité de mise à jour dans le cas de découverte de failles de sécurité, de l'application mais potentiellement aussi de la technologie utilisée.

VII.B Risque Financier

Différence entre le chiffrage initiale et le coût réel

Étant donné notre faible expérience sur ce type de sujet, il se peut que le premier chiffrage lors de cette note de cadrage soit différent du second chiffrage, mais aussi du coût réel.

Causes possibles :

- Oubli d'un élément dans le chiffrage

- Changement des offres des outils que nous voulions utiliser

VII.C Risque de manque de maîtrise d'œuvre

Manque d'expertise technique pour la réalisation de certaines parties du projet.

Causes possibles :

- Technologies ou fonctionnalités nouvelles pour l'équipe (mobile, intégration avec les capteurs mobiles (Android/iOS)).
- La mise en production et déploiement sur les stores nouvelle pour l'équipe.
- Conséquences : Retard, faible qualité, ou impossibilité d'intégrer certaines fonctionnalités comme prévu.

Solutions proposées :

- Cela sera pris en compte dans la planification, en priorisant les fonctionnalités clés du projet.

VIII. Lexique

MIAGE : Méthodes Informatiques Appliquées à la Gestion des Entreprises, une formation universitaire française alliant informatique et gestion.

MIAGE Connection : Fédération nationale des étudiants et diplômés de MIAGE. L'association vise à promouvoir la filière MIAGE et à fédérer les associations MIAGE en France.

MC : Miage Connection.

DCU : Diagramme de Cas d'Utilisation, outil utilisé pour représenter les interactions entre les acteurs et le système.

RGPD : Règlement Général sur la Protection des Données, règlement européen visant à protéger les données personnelles.

API : Application Programming Interface, un ensemble de règles permettant à des logiciels de communiquer entre eux.

Firebase : Plateforme de développement backend proposée par Google, offrant des services tels que l'authentification, la base de données en temps réel, le stockage et les API.

Requête : Action d'envoyer une demande à un serveur ou une API pour récupérer ou manipuler des données. Exemple : requête HTTP (GET, POST, PUT, DELETE) utilisée pour communiquer avec le backend.

Backend : Partie serveur d'une application qui traite la logique métier, gère les bases de données et répond aux requêtes envoyées par le frontend.

BDE : Bureau des Étudiants, une association d'étudiants généralement rattachée à une université ou une école.

Junior Entreprise : Structure étudiante fonctionnant sur le modèle d'un cabinet de conseil, offrant des prestations aux entreprises.

CSV : Comma-Separated Values, un format de fichier texte utilisé pour représenter des données tabulaires.

TLS/SSL : Protocoles de cryptage utilisés pour sécuriser les communications sur internet.

Entité : Regroupe les différentes catégories auxquelles un utilisateur peut être rattaché (BDE, Junior Entreprise, association d'anciens).

Podium : Classement récompensant les meilleures performances dans une compétition donnée.

Événement : Date du déblocage d'un badge

Badge : Récompense visuelle attribuée aux utilisateurs en fonction de leurs performances ou accomplissements dans l'application.

Must Have : Fonctionnalité obligatoire à intégrer dans l'application.

Should Have : Fonctionnalité importante mais non critique.

Could Have : Fonctionnalité optionnelle qui pourrait être implémentée si les ressources et le temps le permettent.

React Native : Framework de développement d'applications mobiles multiplateformes (Android et iOS).

Chiffrement : Technique de sécurité visant à rendre les données illisibles à toute personne non autorisée.

Politique de confidentialité : Document décrivant les pratiques en matière de collecte, d'utilisation, de stockage et de partage des données personnelles.

Espace Économique Européen (EEE) : Zone composée des États membres de l'Union européenne plus quelques pays tiers, où les règles du RGPD s'appliquent.

UI/UX : User Interface / User Experience, désignent respectivement l'apparence de l'interface et l'expérience utilisateur globale.

Accessibilité : Ensemble de pratiques et technologies visant à rendre une application ou un site web utilisable par le plus grand nombre de personnes, y compris celles ayant des limitations physiques, sensorielles ou cognitives.


WCAG (Web Content Accessibility Guidelines) : Normes internationales pour l'accessibilité numérique, visant à garantir que les applications et les sites web sont utilisables par tous, y compris les personnes en situation de handicap.

Audit d'accessibilité : Processus d'évaluation d'une application ou d'un site pour identifier les problèmes d'accessibilité et assurer leur conformité avec les normes (comme WCAG).


Wireframe : Dans la conception Web, un wireframe ou diagramme de wireframe est une représentation visuelle en niveaux de gris de la structure et de la fonctionnalité d'une seule page Web ou d'un écran d'application mobile.

IX. Annexe

Premier échange de document avec le client et premières propositions de solutions :

 Note de cadrage


Second échange de documents avec détail sur les risques et le chiffage :

 Retour note de cadrage

Questionnaire préparatoire à l'interview 3 :  QuestionsPostCC.pdf

Réponses questions par mail du 8/11 :  Réponse Lucas Ricquebourg au 8/11

Troisième échange de documents, des rapides précisions sur les accès aux API propriétaires et les nécessités de compte développeur payants :

 Précisions suite à l'interview 2

Echanges de mail :

<https://drive.google.com/file/d/1oliDyDXlukCVFG3abJFQ1NMNJpXfcWEb/view?usp=sharing>