

Rapport d'itération 2

|  |  |
| --- | --- |
| **Laboratoire** | 02 |
| **Étudiants** | Dominic Roberge  André Koolen  Anthony Martin Coallier |
| **Codes permanents** | ROBD03129209  KOOA23039101  MARA19129107 |
| **Cours** | LOG210 |
| **Session** | Été 2015 |
| **Groupe** | 03 |
| **Professeur** | Yvan Ross |
| **Chargé de laboratoire** | Charbonneau |
| **Date** | 08-06-2015 |

Table of Contents

[Introduction 3](#_Toc421220901)

[Modèle du domaine 3](#_Toc421220902)

[Diagrammes de séquence système 4](#_Toc421220903)

[Contrats d'opérations 9](#_Toc421220904)

[Diagrammes de séquence 12](#_Toc421220905)

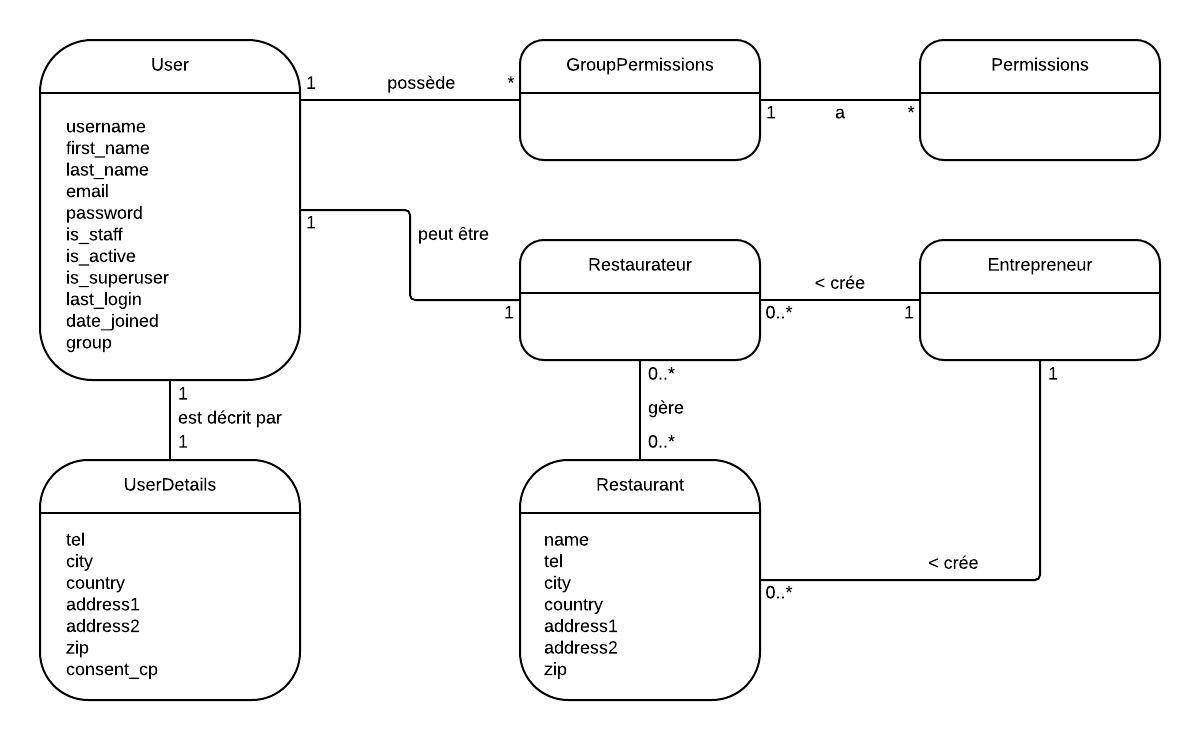
[Conclusion 18](#_Toc421220906)

# Introduction

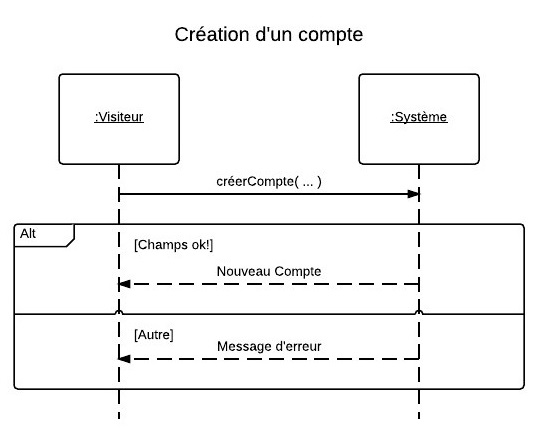
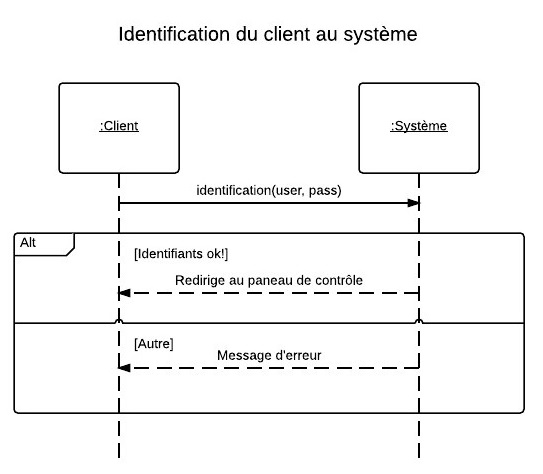
Durant la dernière itération, nous avons installé Django, préparer le projet, appliqué le CSS de Bootstrap3, etc. Quant aux exigences logiciels, nous avons implanté la base de la création et la gestion des comptes.

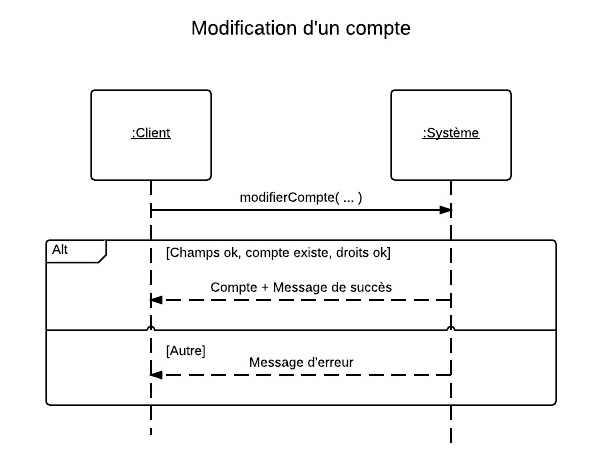
Nous entamons maintenant la deuxième itération qui portera principalement sur la gestion des restaurants et des restaurateurs. Nous devons également corriger le point mentionné dans le plan d'itération pour l'identification.

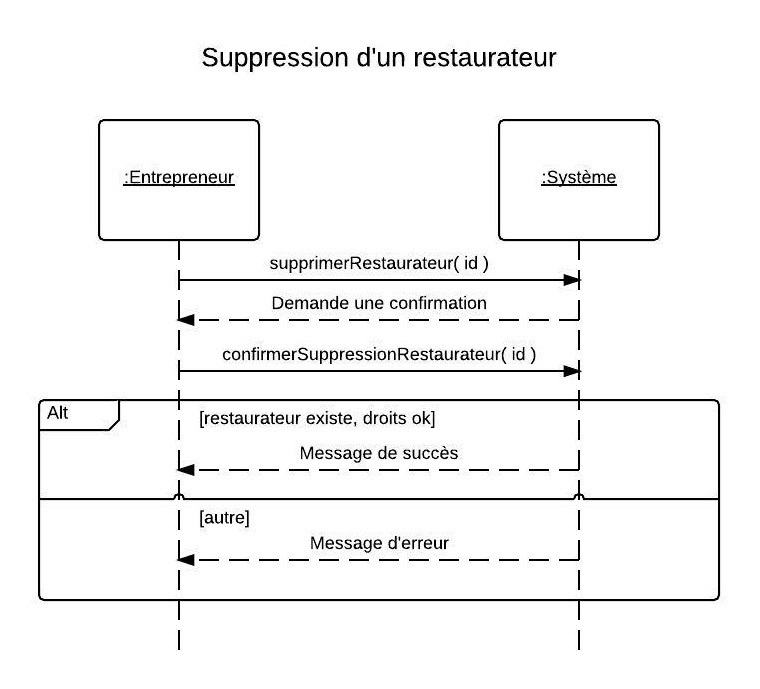
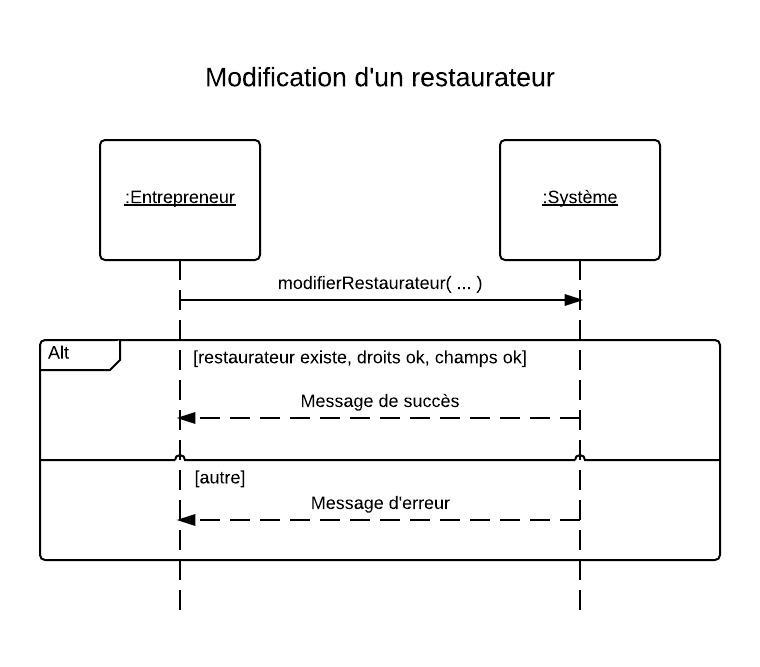
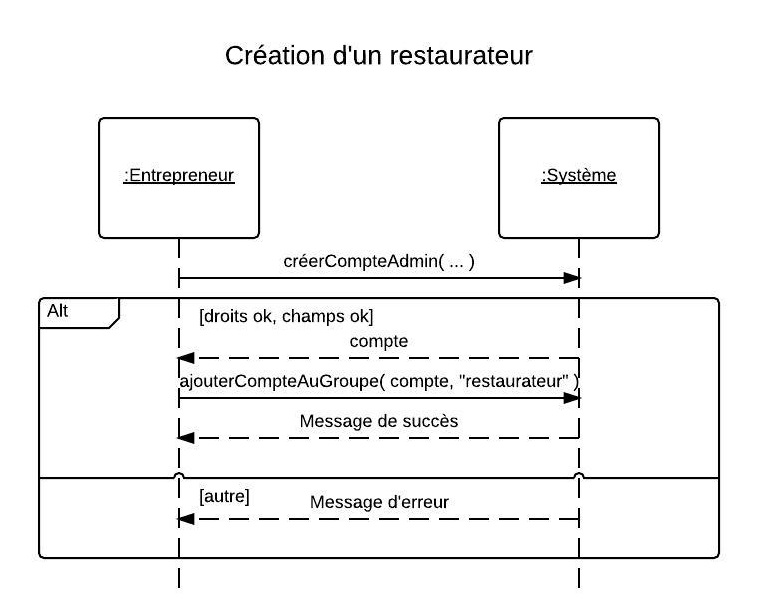
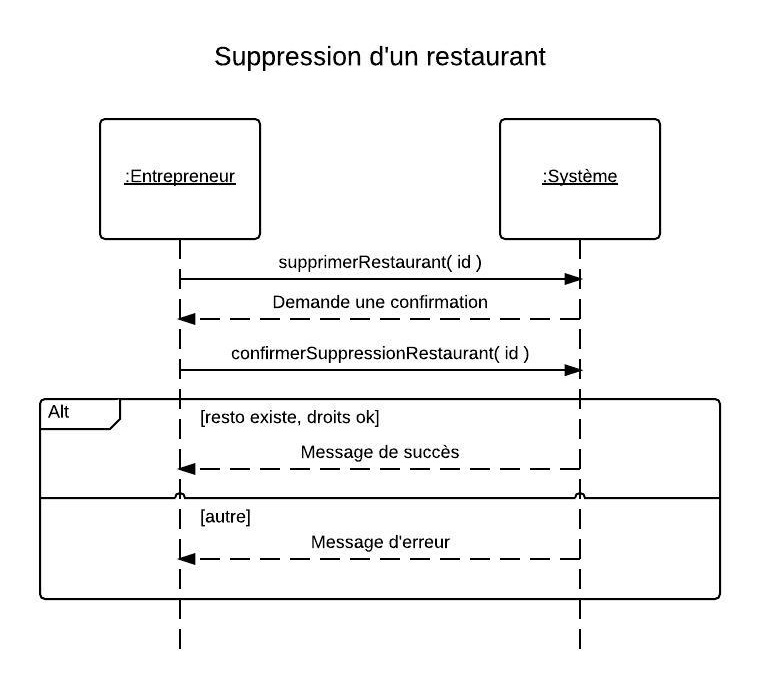
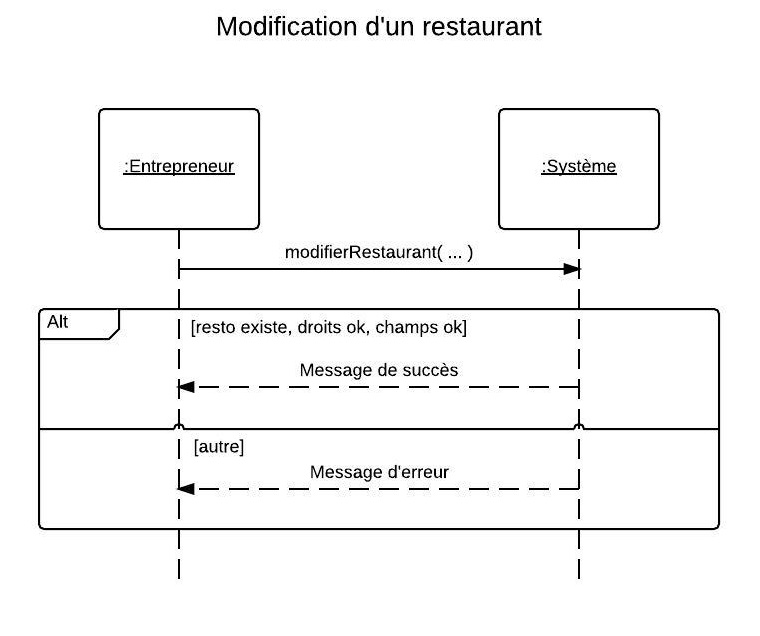
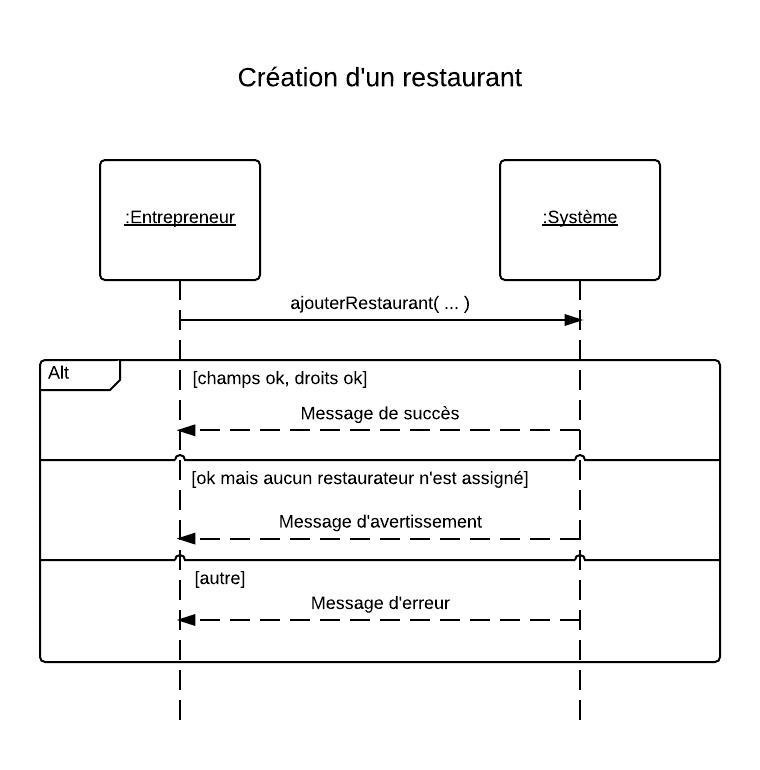
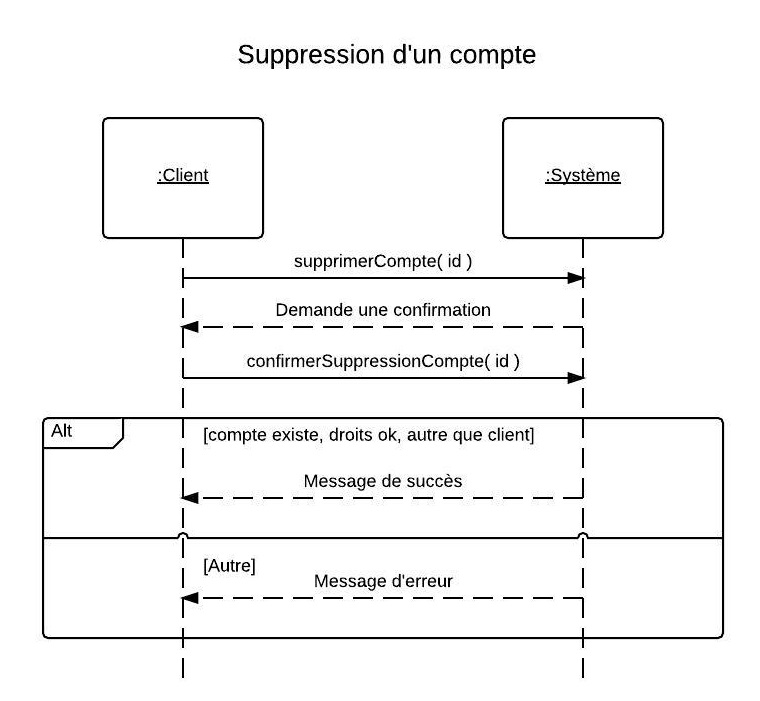
# Modèle du domaine



# Diagrammes de séquence système







# Contrats d'opérations

|  |  |
| --- | --- |
| **Opération** | identification(user:String, pass:String) |
| **Référence croisée** | Identification du client |
| **Préconditions** | Le compte existe |
| **Postconditions** | * Création d'un objet Session * Attribut de l'objet User last\_login modifié |

|  |  |
| --- | --- |
| **Opération** | créerCompte(user:String, pass:String, nom:String, prénom:String, courriel:String, tel:String, city:String, country:String, address1:String, address2:String, zip:String, consent:Boolean) |
| **Référence croisée** | Création d'un compte |
| **Préconditions** | Aucune |
| **Postconditions** | * Un objet User a été créé * Un objet UserDetails a été créé * Une association entre le User et son UserDetails |

|  |  |
| --- | --- |
| **Opération** | modifierCompte(id: integer, user:String, pass:String, nom:String, prénom:String, courriel:String, tel:String, city:String, country:String, address1:String, address2:String, zip:String) |
| **Référence croisée** | Modification d'un compte |
| **Préconditions** | * Le compte existe et le client est authentifié |
| **Postconditions** | * Les attributs du User sont modifiés |

|  |  |
| --- | --- |
| **Opération** | supprimerCompte(id:Int) |
| **Référence croisée** | Suppression d'un compte |
| **Préconditions** | * Le compte existe |
| **Postconditions** | Aucune |

|  |  |
| --- | --- |
| **Opération** | confirmerSupression(id:Int) |
| **Référence croisée** | Suppression d'un compte |
| **Préconditions** | * Le User existe * Le client est authentifié |
| **Postconditions** | * L'attribut de User "is\_active", est mis à False   *NB. Django déconseille la suppression d'utilisateur pour l'effet de suppression en Cascade des éléments qui y sont associés* |

|  |  |
| --- | --- |
| **Opération** | ajouterRestaurant(name:String, tel:String, city:String, country:String, address1:String, address2:String, zip:String, restaurateur:Restaurateur) |
| **Référence croisée** | Création d'un restaurant |
| **Préconditions** | * L'entrepreneur est identifié comme super admin |
| **Postconditions** | * Création d'un objet Restaurant * Association facultative du Restaurant avec un User du groupe "Restaurateur" |

|  |  |
| --- | --- |
| **Opération** | modifierRestaurant(id: integer, name:String, tel:String, city:String, country:String, address1:String, address2:String, zip:String, restaurateur:Restaurateur) |
| **Référence croisée** | Modification d'un restaurant |
| **Préconditions** | * L'entrepreneur est identifié comme super admin * Le restaurant existe |
| **Postconditions** | * Les attributs du Restaurent sont modifiés * Association facultative du Restaurant avec un User du groupe "Restaurateur" |

|  |  |
| --- | --- |
| **Opération** | supprimerRestaurant(id:Integer) |
| **Référence croisée** | Suppression d'un restaurant |
| **Préconditions** | * L'entrepreneur est identifié comme super admin * Le restaurent existe |
| **Postconditions** | Aucune |

|  |  |
| --- | --- |
| **Opération** | confirmerSuppressionRestaurant(id:Integer) |
| **Référence croisée** | Suppression d'un restaurant |
| **Préconditions** | * L'entrepreneur est identifié comme super admin * Le restaurent existe |
| **Postconditions** | * Le Restaurant est supprimé |

|  |  |
| --- | --- |
| **Opération** | créerCompteAdmin(username:String, password1:String, password2:String) |
| **Référence croisée** | Création d'un restaurateur |
| **Préconditions** | * L'entrepreneur est identifié comme super admin |
| **Postconditions** | * Un User est créé   *NB. Le module Admin de Django n'associe pas les droits d'accès à l'utilisateur par défaut. L'entrepreneur est redirigé vers la modification du compte (opération* ***modifierRestaurateur****) et dois cocher "* *Statut équipe".* |

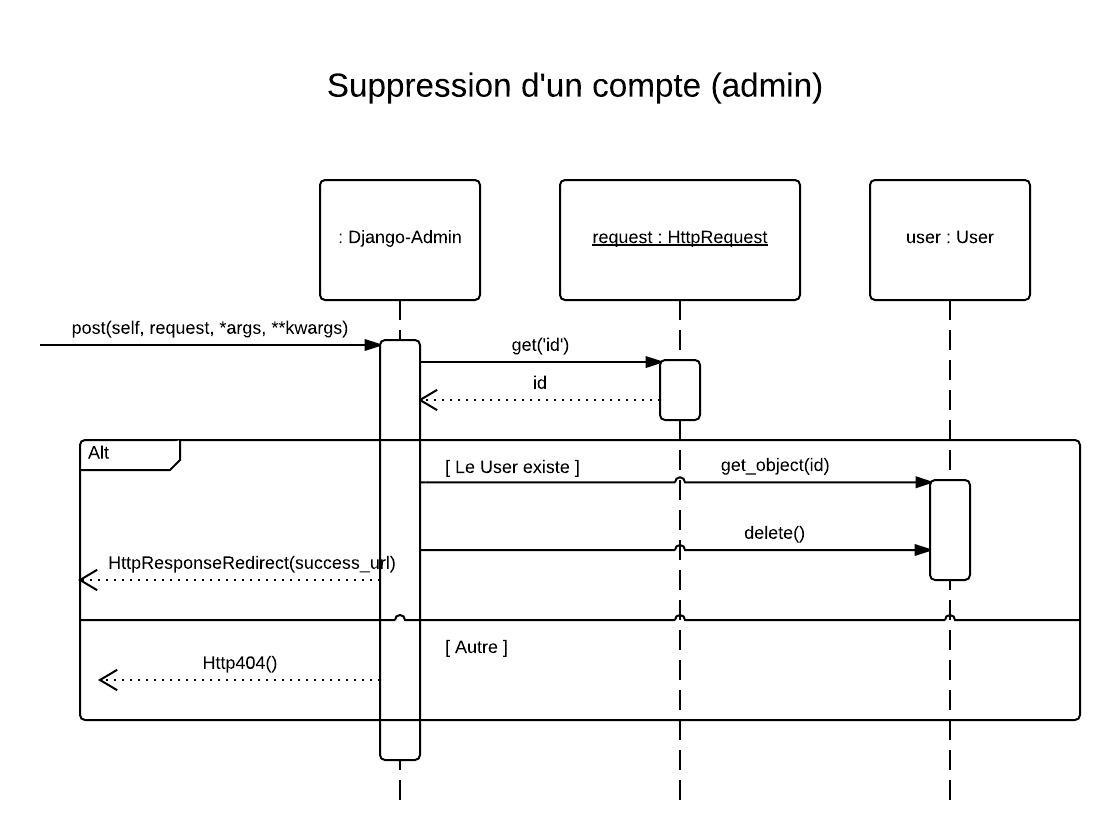
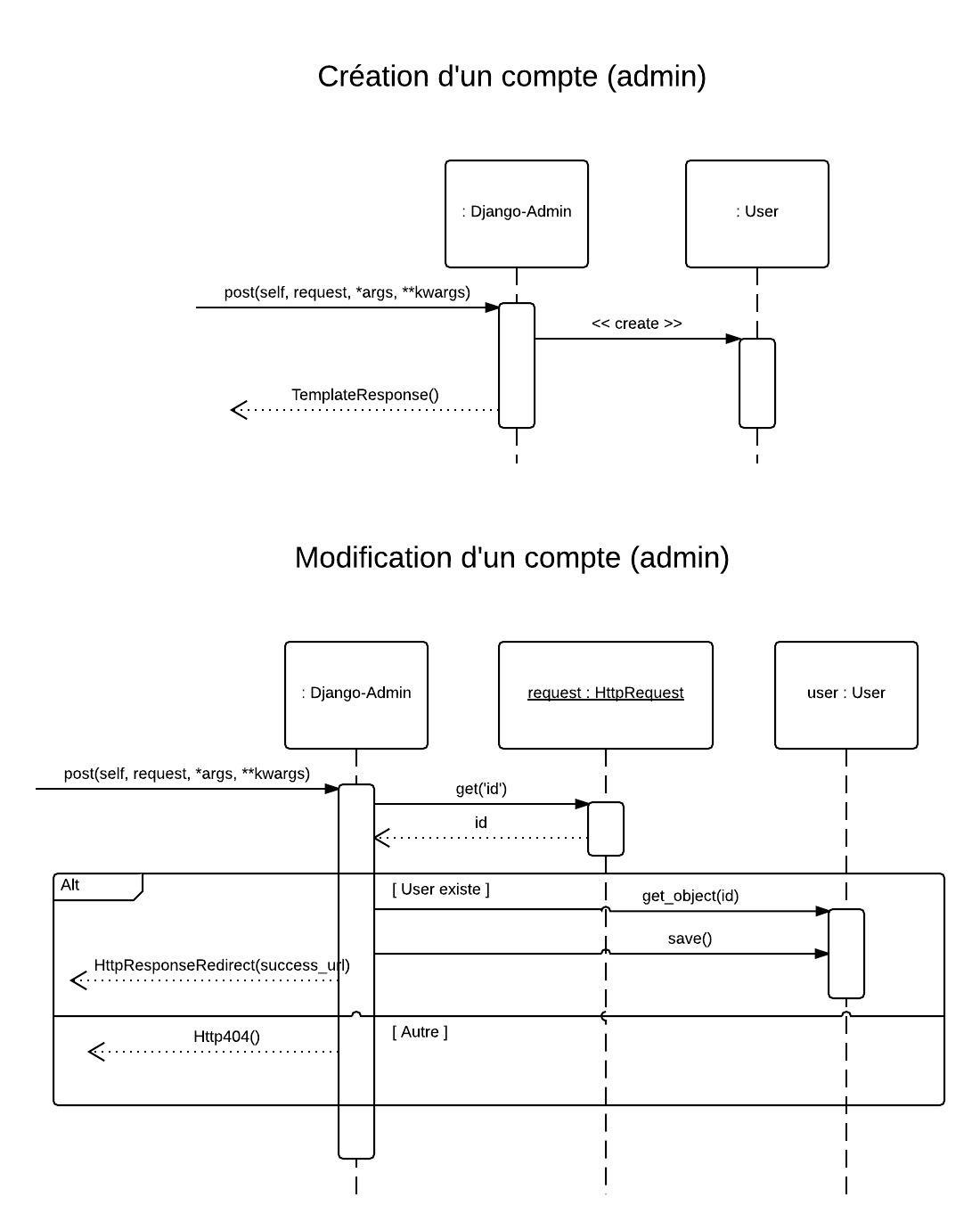
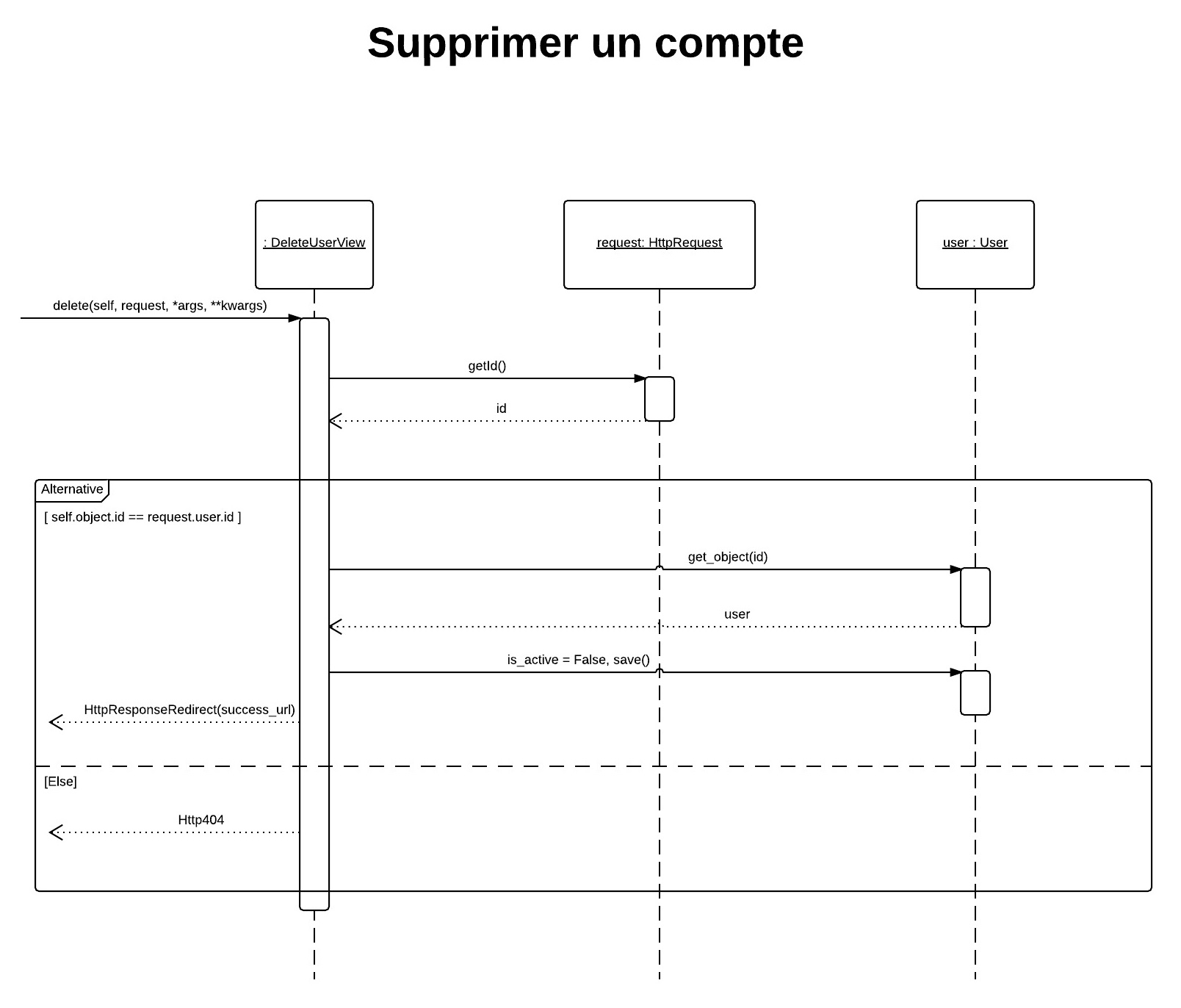
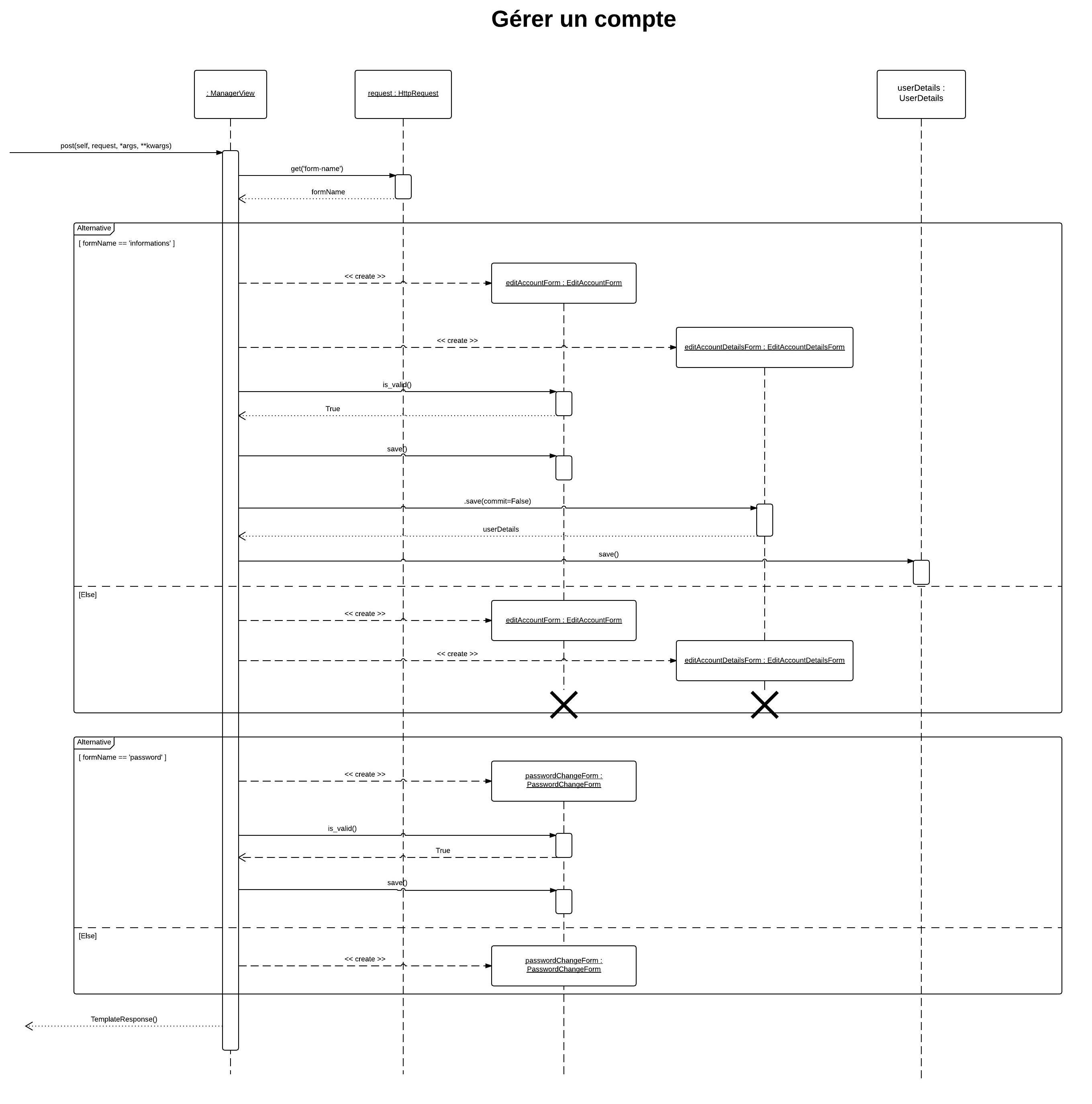
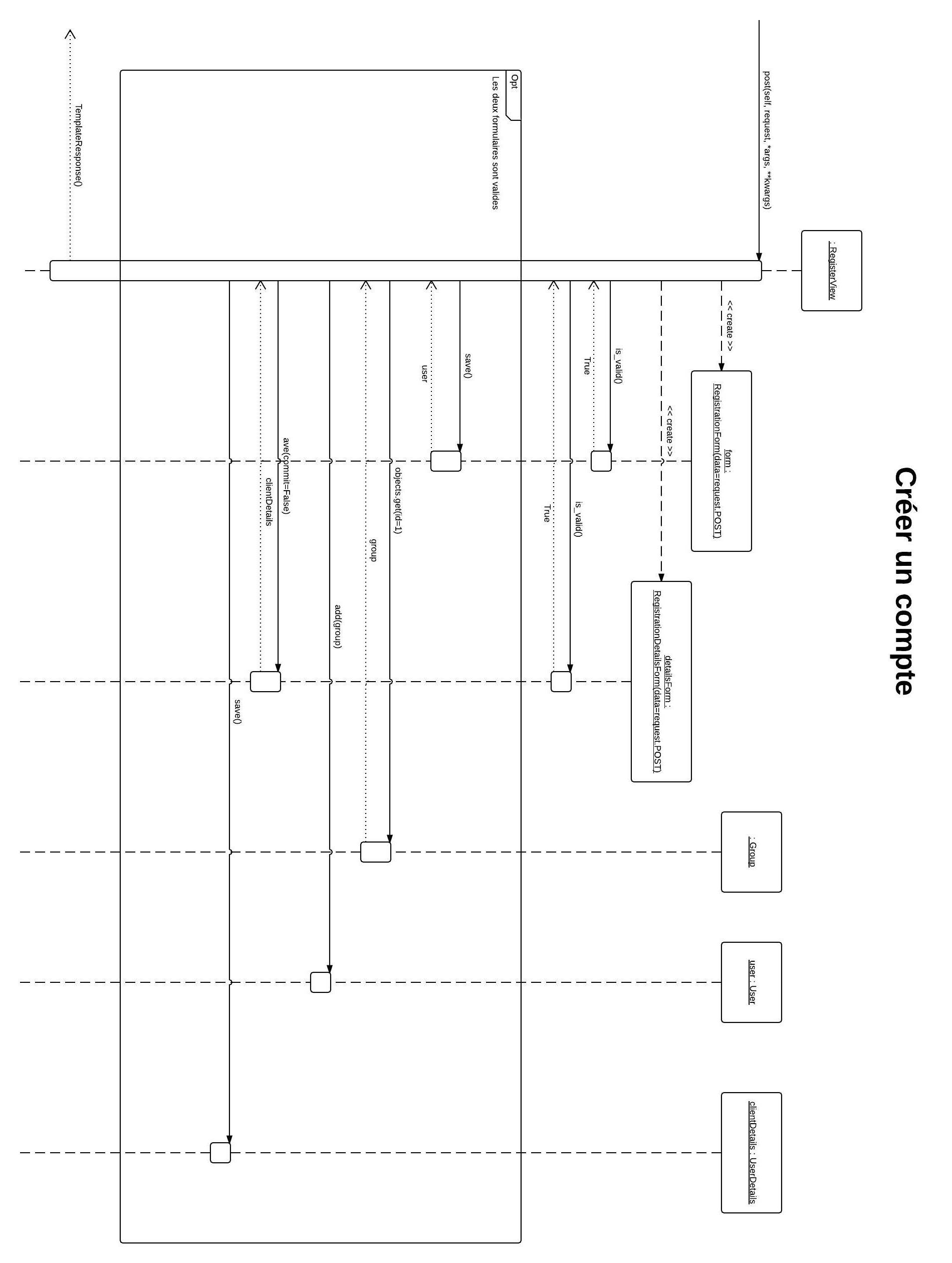
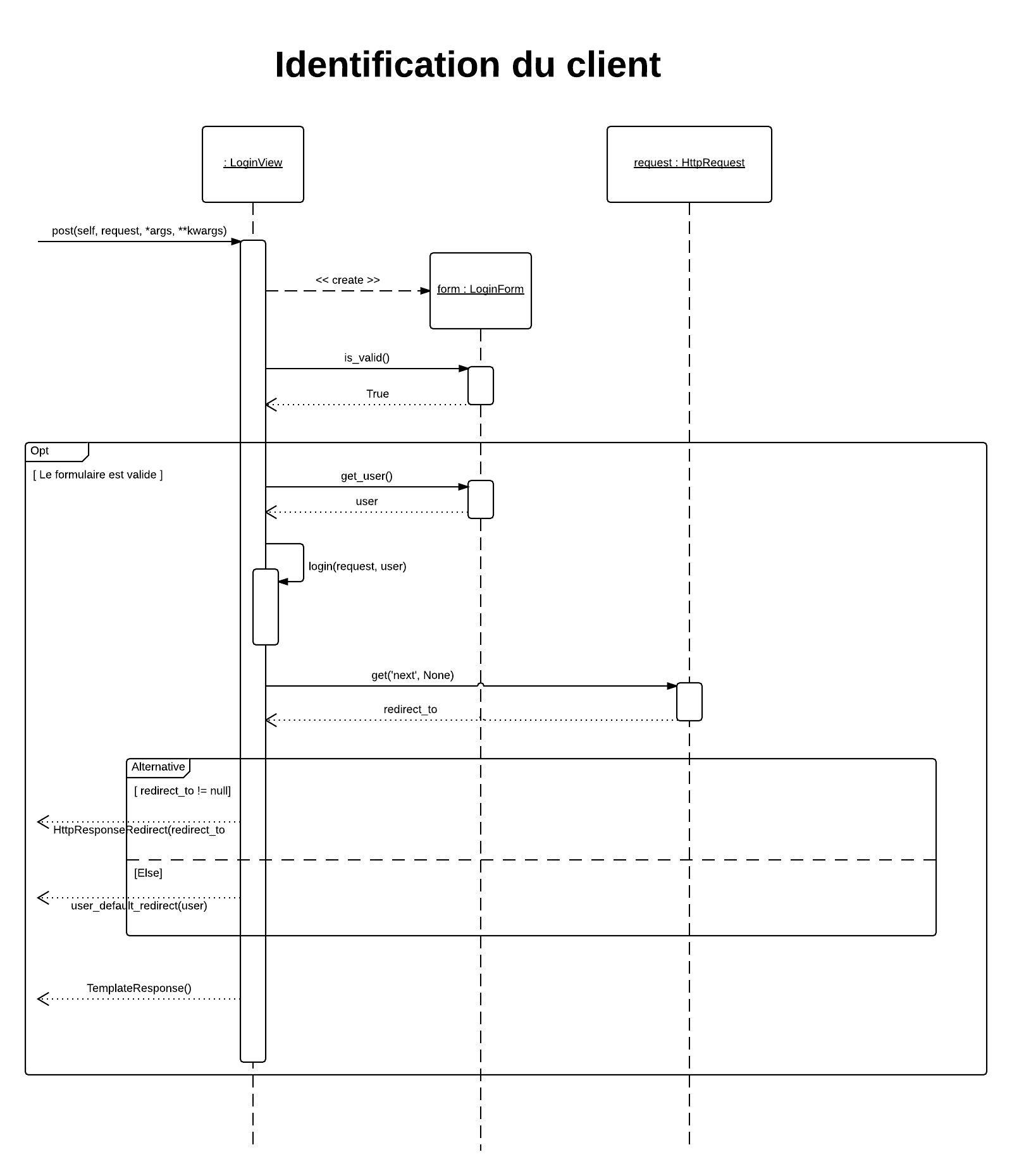
|  |  |
| --- | --- |
| **Opération** | ajouterCompteAuGroupe(id:Integer, groupe:String) |
| **Référence croisée** | Création d'un restaurateur |
| **Préconditions** | * L'entrepreneur est identifié comme super admin * Le User existe |
| **Postconditions** | * Une association est créée entre l'instance de Groupe "restaurateur" et le User concerné |

|  |  |
| --- | --- |
| **Opération** | modifierRestaurateur(id:Integer, username:String, password1:String, password2:String, first\_name:String, last\_name:String, email:String, is\_staff:Boolean, is\_active:Boolean) |
| **Référence croisée** | Modification d'un restaurateur |
| **Préconditions** | * L'entrepreneur est identifié comme super admin * Le User existe |
| **Postconditions** | * Les attributs du User sont modifiés |

|  |  |
| --- | --- |
| **Opération** | supprimerRestaurateur(id:Integer) |
| **Référence croisée** | Suppression d'un restaurateur |
| **Préconditions** | * L'entrepreneur est identifié comme super admin * Le User existe |
| **Postconditions** | Aucune |

|  |  |
| --- | --- |
| **Opération** | confirmerSuppressionRestaurateur(id:Integer) |
| **Référence croisée** | Suppression d'un restaurateur |
| **Préconditions** | * L'entrepreneur est identifié comme super admin * Le User existe |
| **Postconditions** | * Le User est supprimé |

# Diagrammes de séquence



# Conclusion

Durant cette itération nous avons donc implémenter l'ajout de restaurant et décrit les artéfact pour la création d'un restaurateur, qui est en fait déjà codé par le module Admin de Django. Nous avons également entamé le travail sur le design *front-end* du produit.

La prochaine itération qui commence à partir du 8 juin traitera de la gestion des menus dans le module pour les restaurants ainsi que d'autres points qui seront décidés lors de la planification.