



Le génie pour l'industrie

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

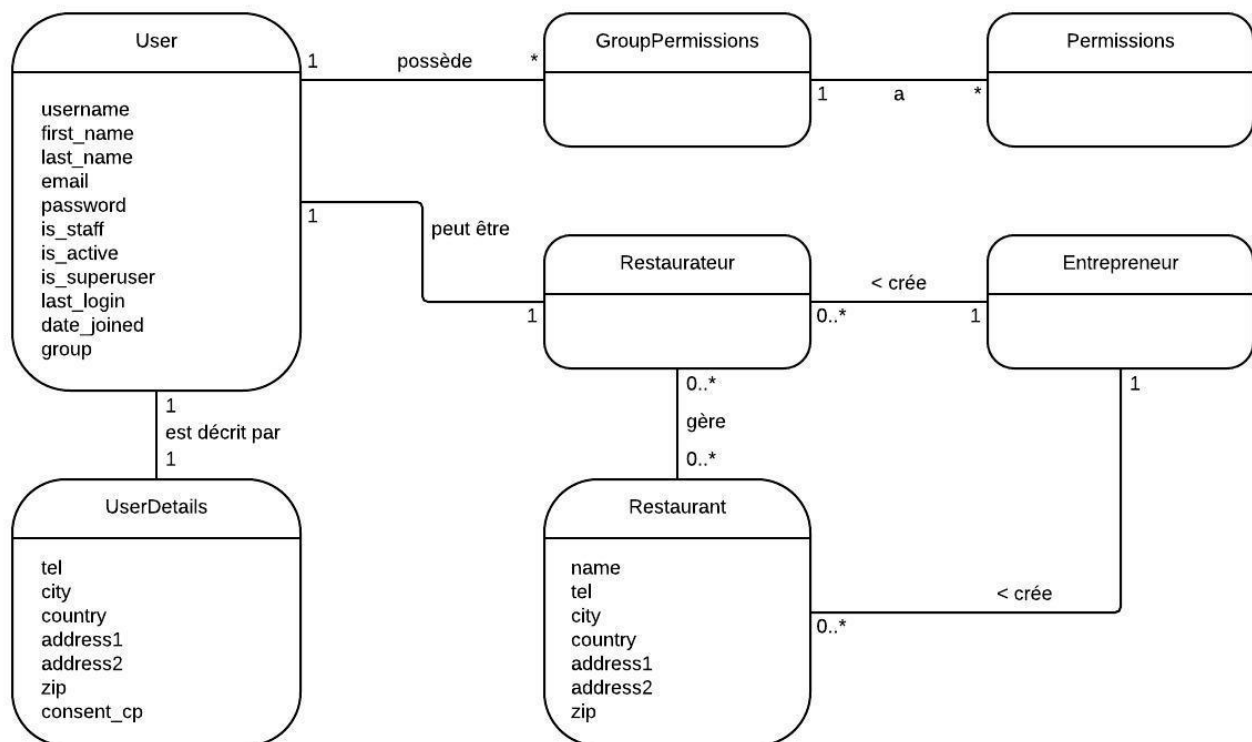
Introduction.....	3
Modèle du domaine	3
Diagrammes de séquence système.....	4
Contrats d'opérations	9
Diagrammes de séquence	12
Conclusion	18

Introduction

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

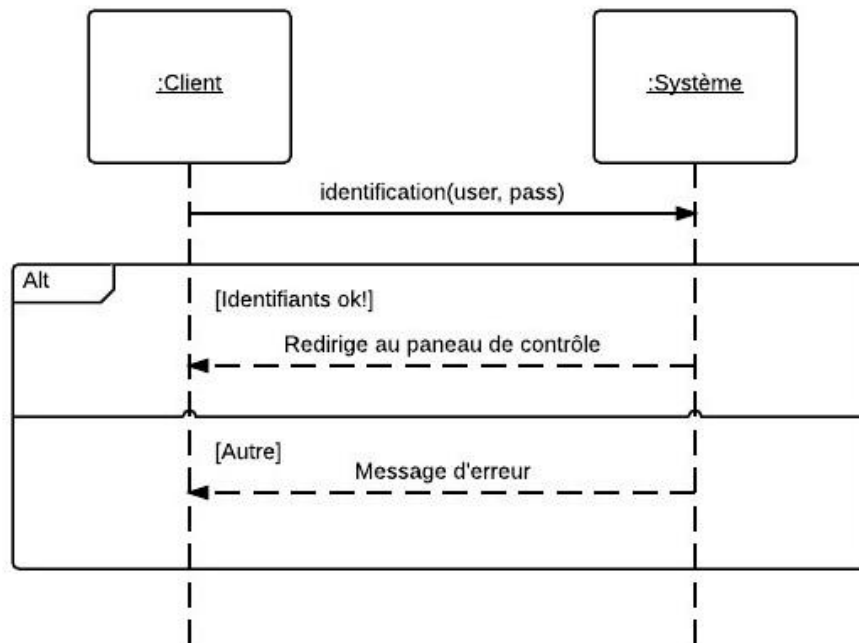
Nous entamons maintenant la deuxième itération qui portera principalement sur la gestion des restaurants et des restaurateurs. Nous devons également corriger le point mentionné dans le plan d'itération pour l'identification.

Modèle du domaine

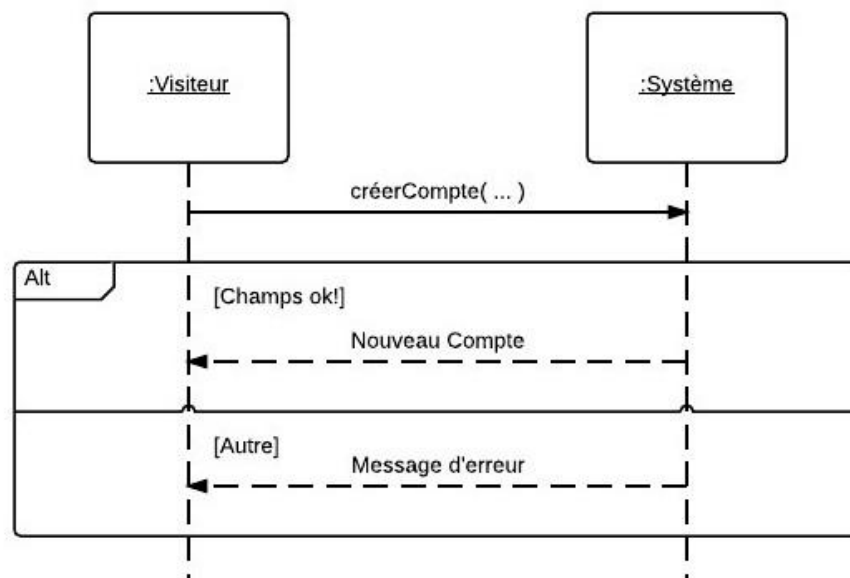


Diagrammes de séquence système

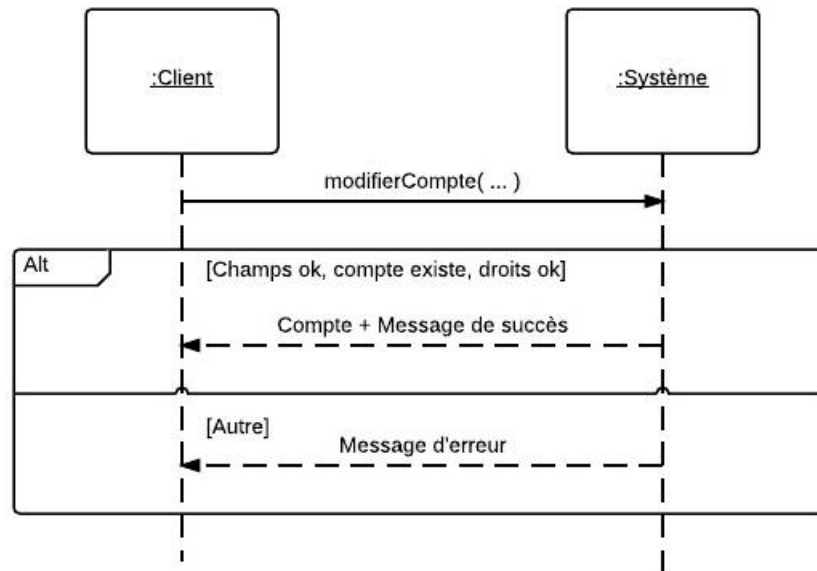
Identification du client au système



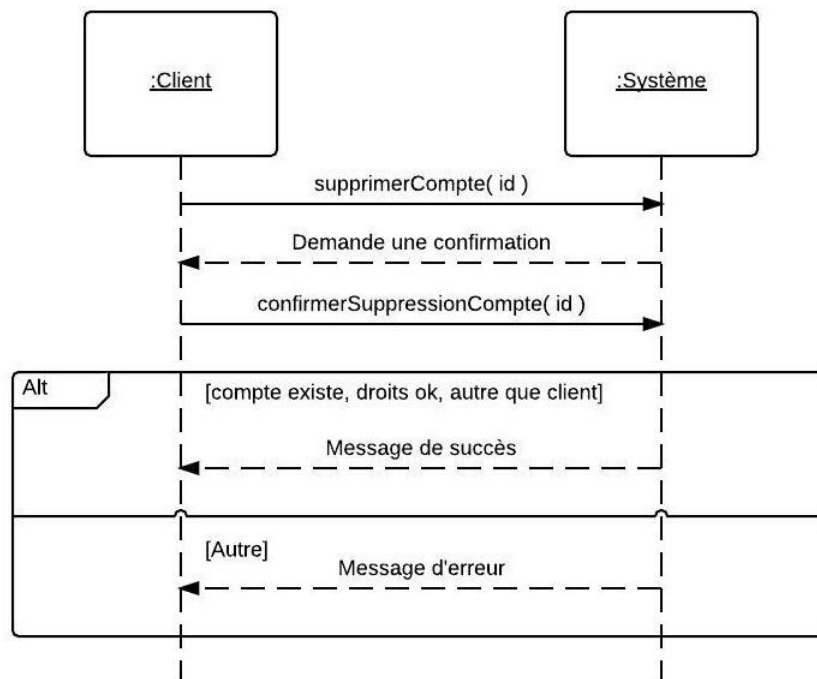
Création d'un compte



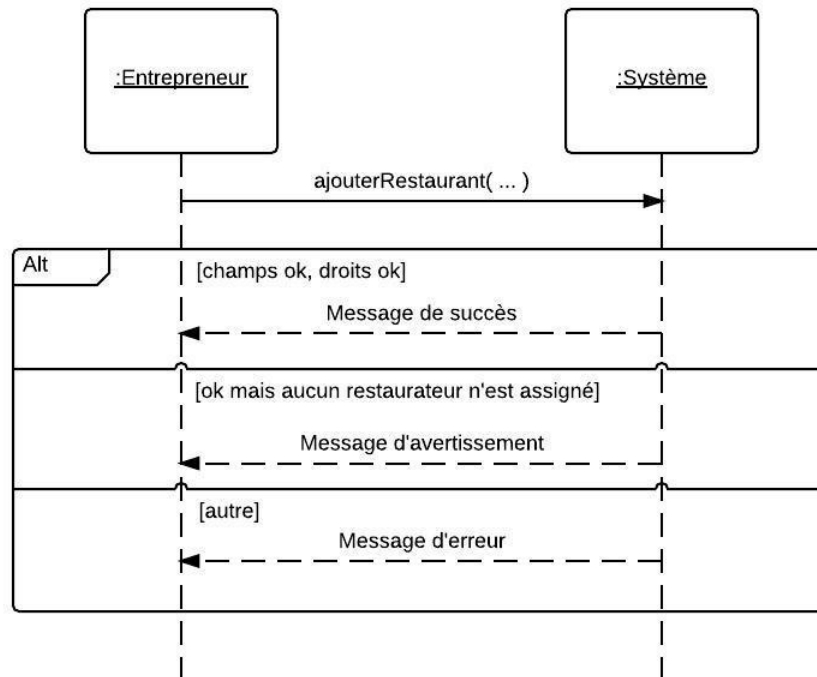
Modification d'un compte



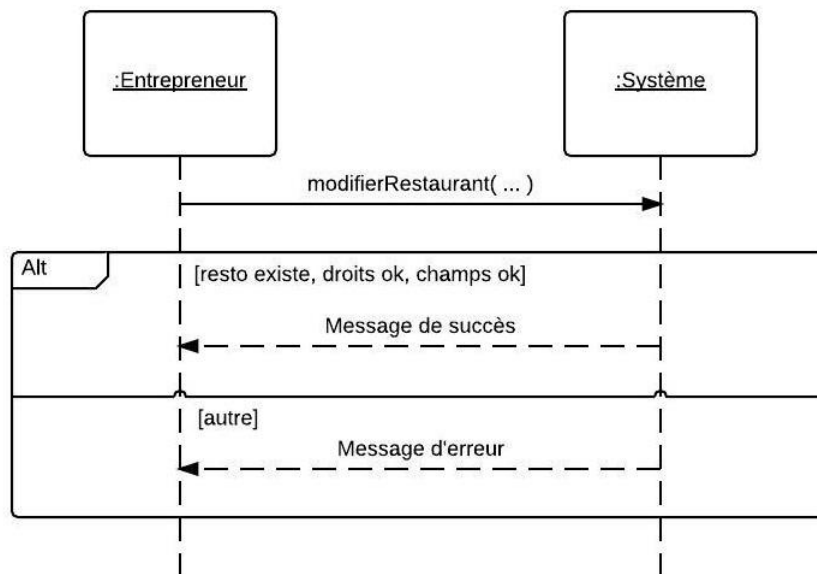
Suppression d'un compte



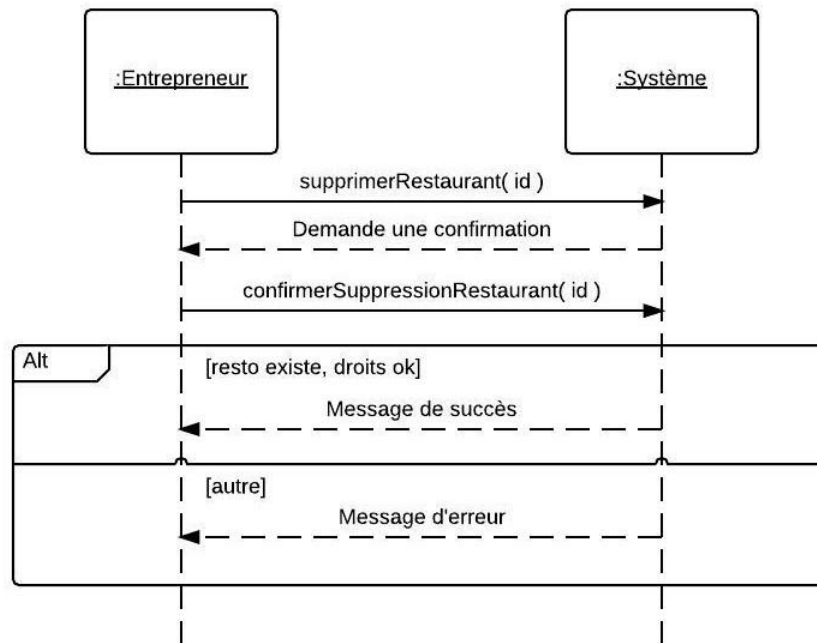
Création d'un restaurant



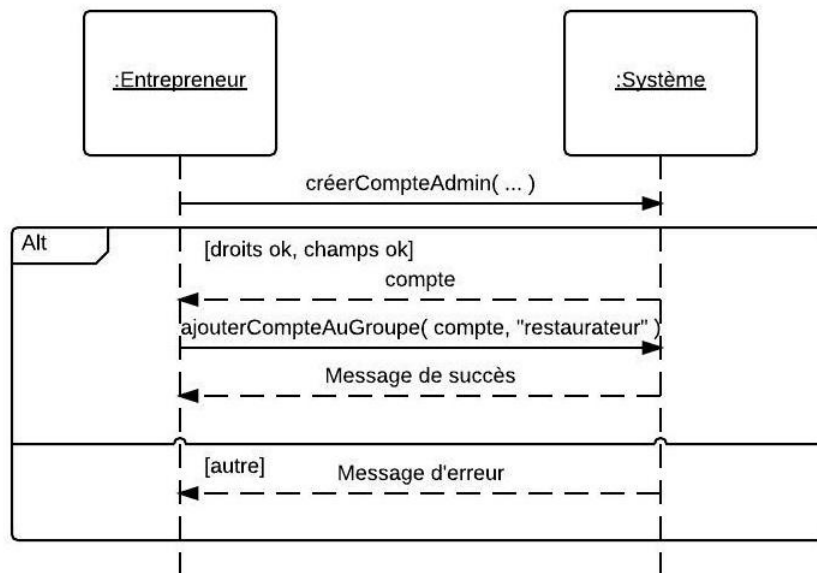
Modification d'un restaurant



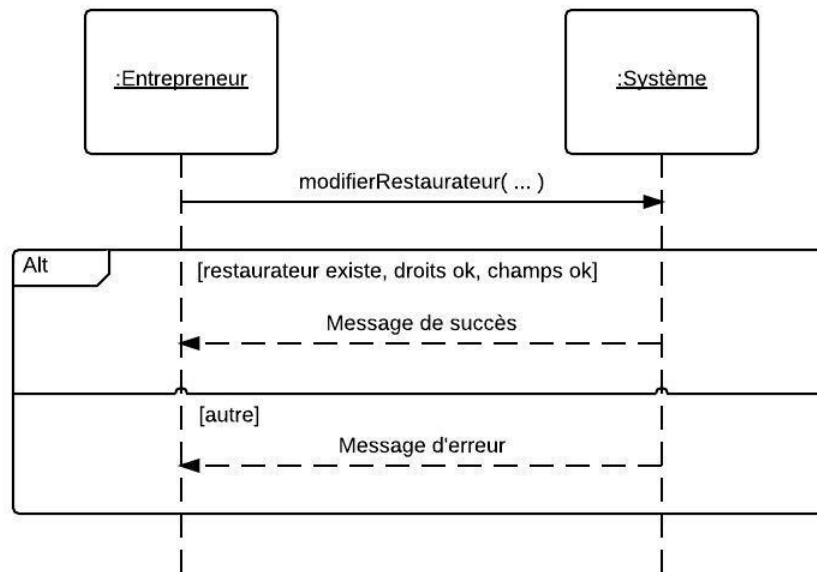
Suppression d'un restaurant



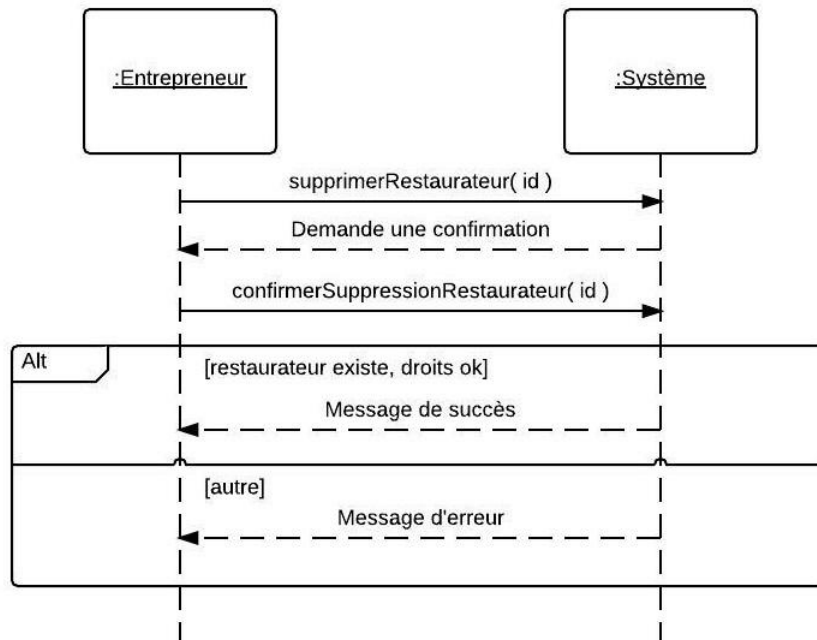
Création d'un restaurateur



Modification d'un restaurateur



Suppression d'un restaurateur



Contrats d'opérations

Opération	identification(user:String, pass:String)
Référence croisée	Identification du client
Préconditions	Le compte existe
Postconditions	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Le compte existe et le client est authentifié
Postconditions	<ul style="list-style-type: none">• Les attributs du User sont modifiés

Opération	supprimerCompte(id:Int)
Référence croisée	Suppression d'un compte
Préconditions	<ul style="list-style-type: none">• Le compte existe
Postconditions	Aucune

Opération	confirmerSupression(id:Int)
Référence croisée	Suppression d'un compte
Préconditions	<ul style="list-style-type: none">• Le User existe• Le client est authentifié
Postconditions	<ul style="list-style-type: none">• L'attribut de User "is_active", est mis à False <p><i>NB. Django déconseille la suppression d'utilisateur pour l'effet de suppression en Cascade des éléments qui y sont associés</i></p>

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	<ul style="list-style-type: none"> • L'entrepreneur est identifié comme super admin
Postconditions	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • L'entrepreneur est identifié comme super admin • Le restaurant existe
Postconditions	<ul style="list-style-type: none"> • Les attributs du Restaurant sont modifiés • Association facultative du Restaurant avec un User du groupe "Restaurateur"

Opération	supprimerRestaurant(id:Integer)
Référence croisée	Suppression d'un restaurant
Préconditions	<ul style="list-style-type: none"> • L'entrepreneur est identifié comme super admin • Le restaurant existe
Postconditions	Aucune

Opération	confirmerSuppressionRestaurant(id:Integer)
Référence croisée	Suppression d'un restaurant
Préconditions	<ul style="list-style-type: none"> • L'entrepreneur est identifié comme super admin • Le restaurant existe
Postconditions	<ul style="list-style-type: none"> • Le Restaurant est supprimé

Opération	créerCompteAdmin(username:String, password1:String, password2:String)
Référence croisée	Création d'un restaurateur
Préconditions	<ul style="list-style-type: none"> • L'entrepreneur est identifié comme super admin
Postconditions	<ul style="list-style-type: none"> • Un User est créé <p><i>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 doit cocher " Statut équipe".</i></p>

Opération	ajouterCompteAuGroupe(id:Integer, groupe:String)
Référence croisée	Création d'un restaurateur
Préconditions	<ul style="list-style-type: none"> • L'entrepreneur est identifié comme super admin • Le User existe
Postconditions	<ul style="list-style-type: none"> • Une association est créée entre l'instance de Groupe "restaurateur" et le User concerné

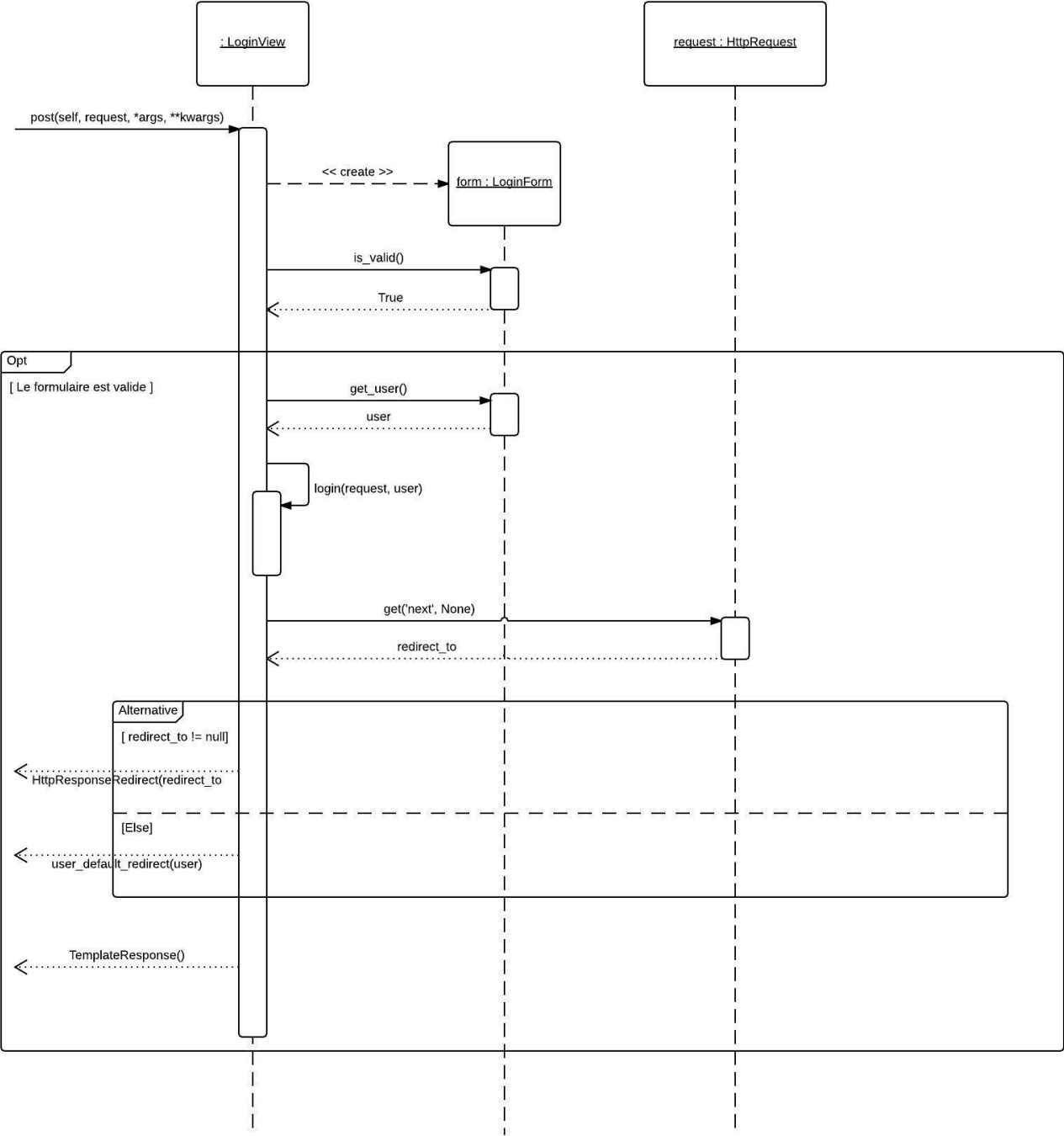
Opération	modifierRestaurateur(id:Integer, username:String, password1:String, password2:String, first_name:String, last_name:String, email:String, is_staff:Boolean, is_active:Boolean)
Référence croisée	Modification d'un restaurateur
Préconditions	<ul style="list-style-type: none"> • L'entrepreneur est identifié comme super admin • Le User existe
Postconditions	<ul style="list-style-type: none"> • Les attributs du User sont modifiés

Opération	supprimerRestaurateur(id:Integer)
Référence croisée	Suppression d'un restaurateur
Préconditions	<ul style="list-style-type: none"> • L'entrepreneur est identifié comme super admin • Le User existe
Postconditions	Aucune

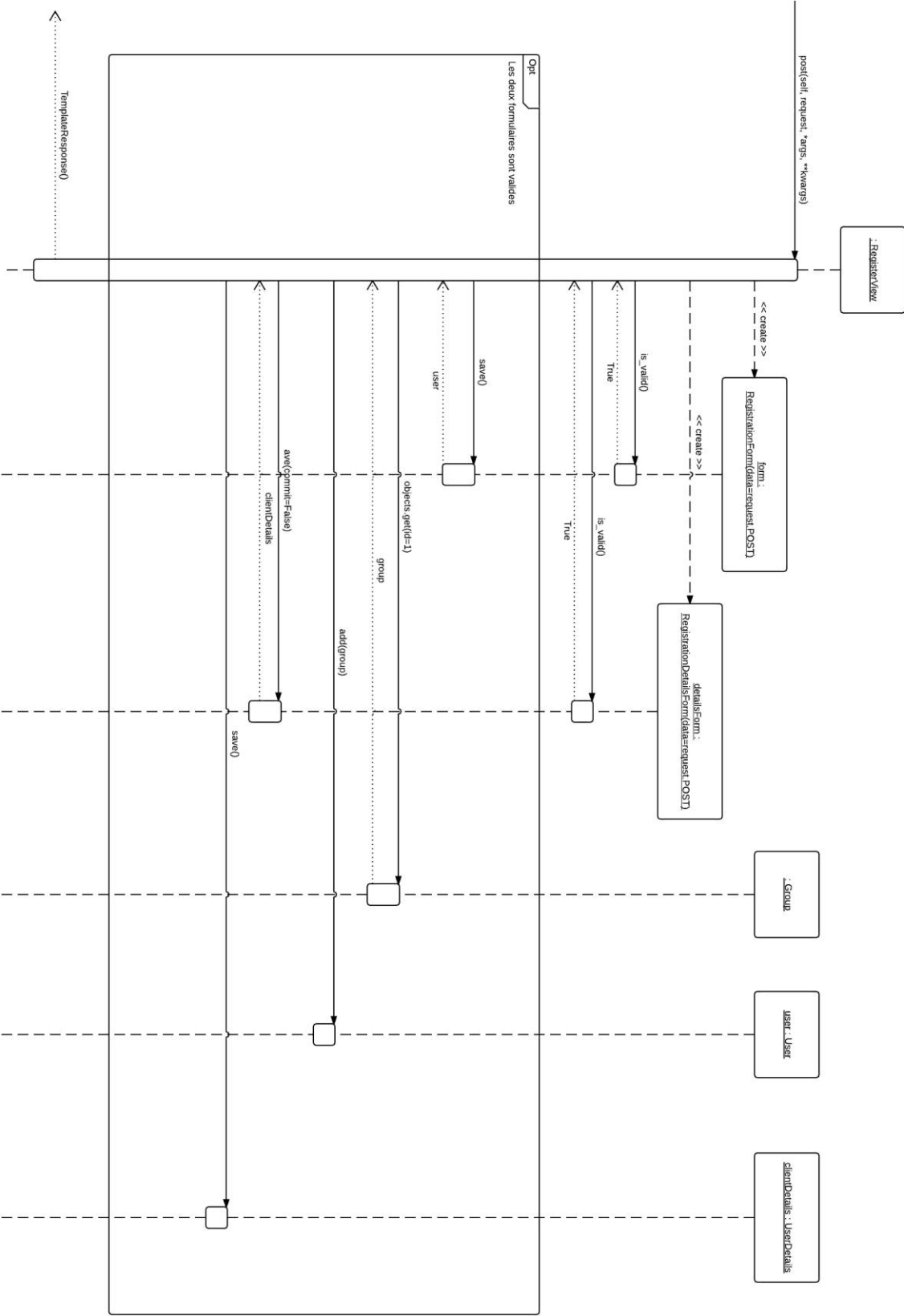
Opération	confirmerSuppressionRestaurateur(id:Integer)
Référence croisée	Suppression d'un restaurateur
Préconditions	<ul style="list-style-type: none"> • L'entrepreneur est identifié comme super admin • Le User existe
Postconditions	<ul style="list-style-type: none"> • Le User est supprimé

Diagrammes de séquence

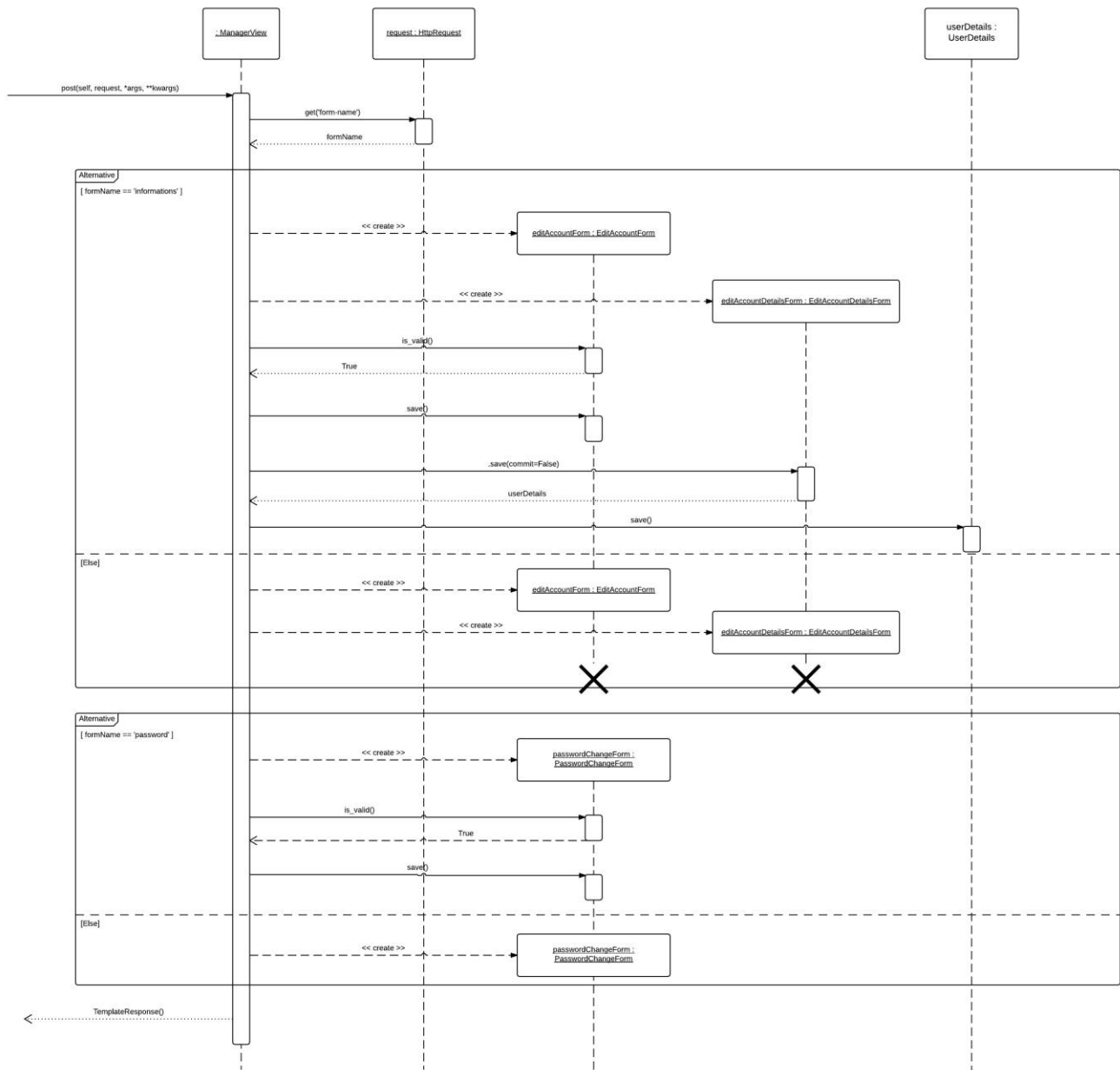
Identification du client



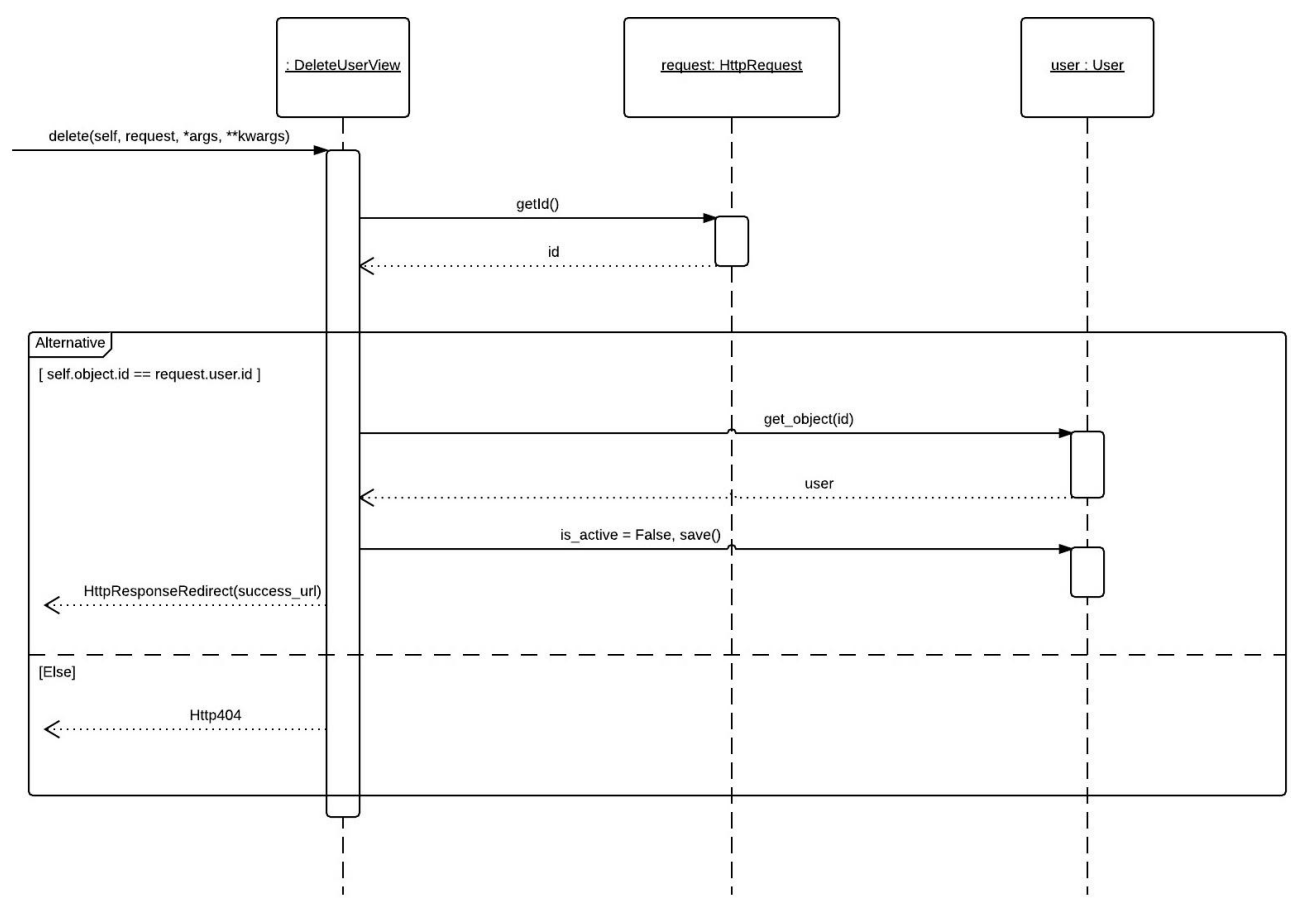
Créer un compte



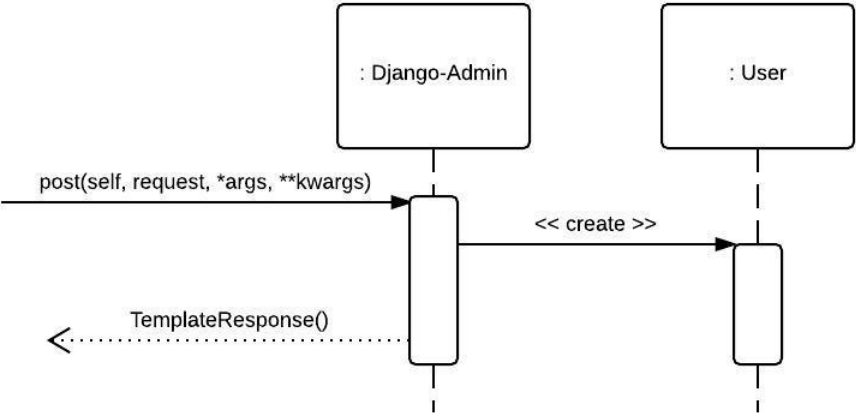
Gérer un compte



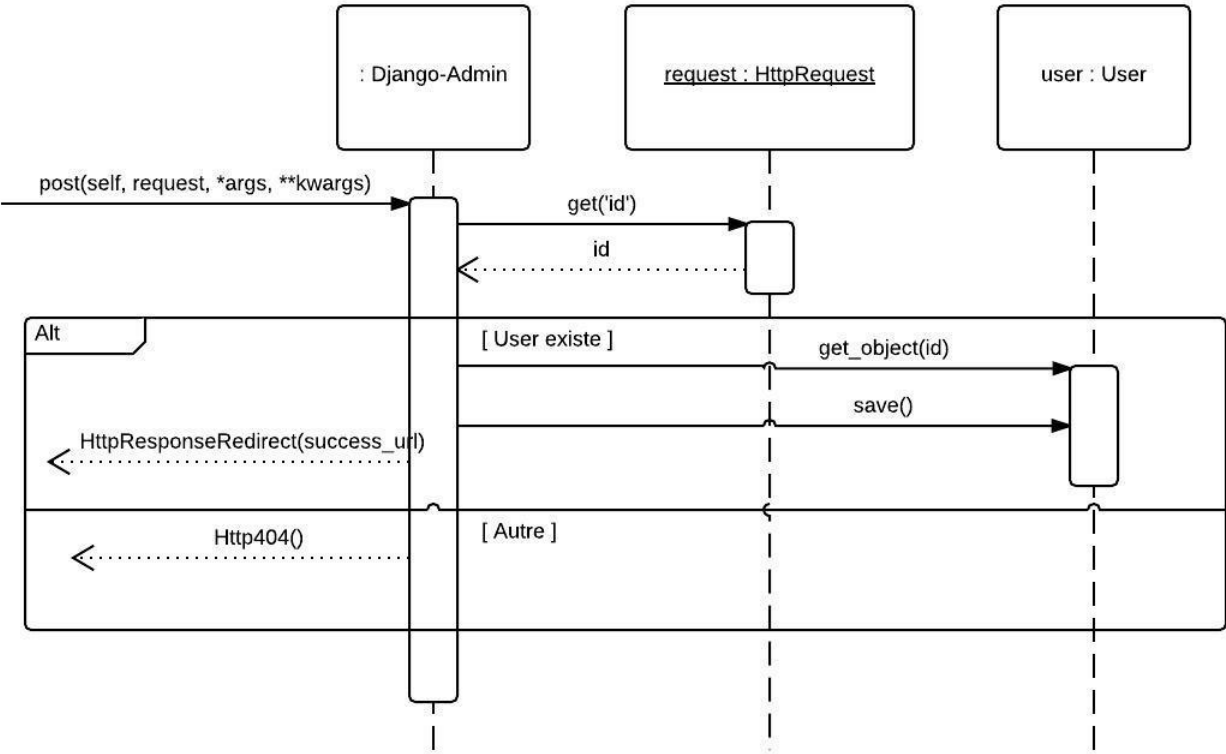
Supprimer un compte



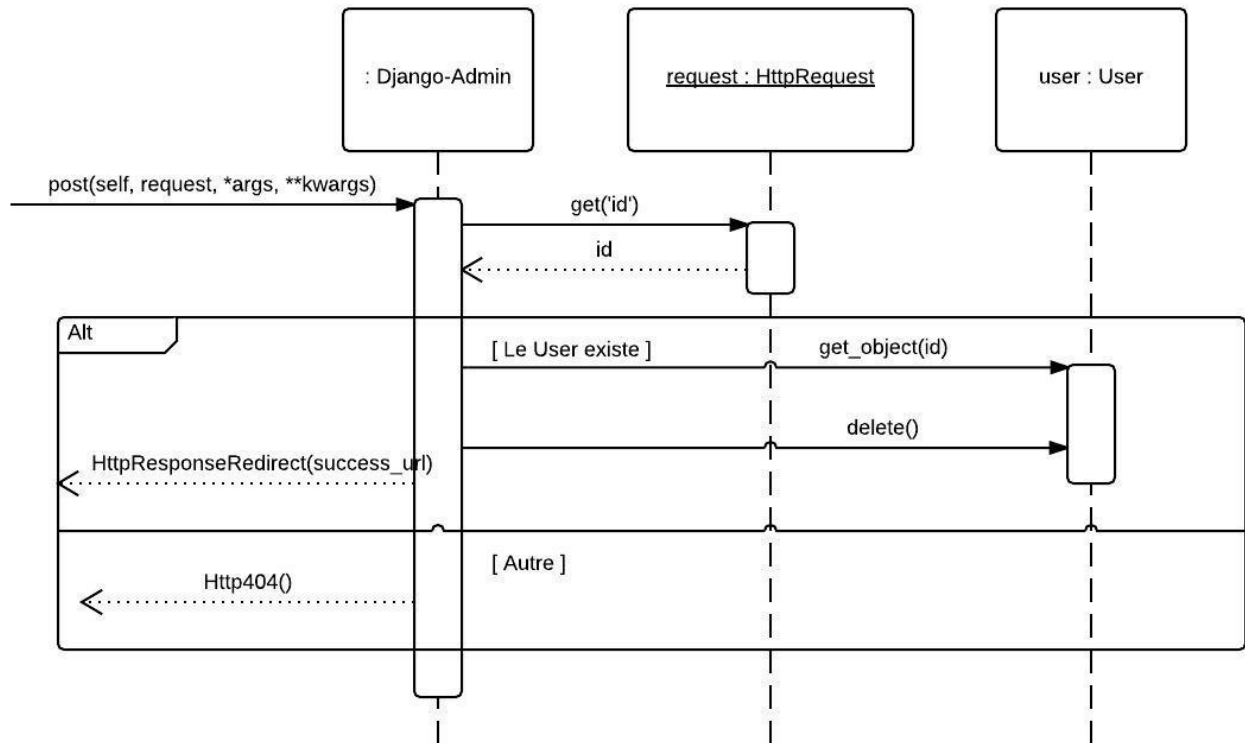
Création d'un compte (admin)



Modification d'un compte (admin)



Suppression d'un compte (admin)



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.