

**École de technologie supérieure**

*Département de génie logiciel et des technologies de l'information*

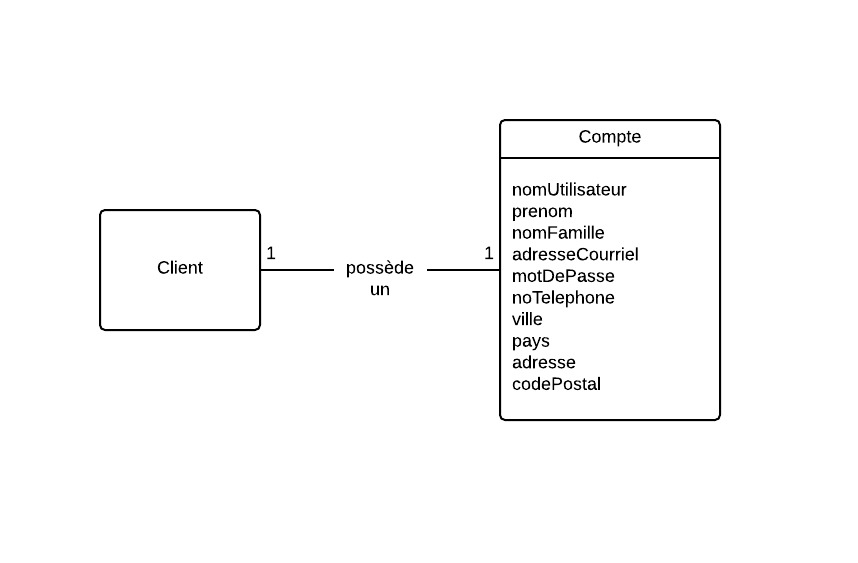
Rapport itération #1

|  |  |
| --- | --- |
| **No du laboratoire** | 01 |
| **Étudiants** | Dominic Roberge  André Koolen  Anthony Martin Coallier |
| **Codes permanents** | ROBD03129209  KOOA23039101  MARA19129107 |
| **Cours** | LOG210 |
| **Session** | Été 2015 |
| **Groupe** | 03 |
| **Professeur** | Ross, Yvan |
| **Chargé de laboratoire** | Charbonneau, Philippe |
| **Date** | 25-05-2015 |

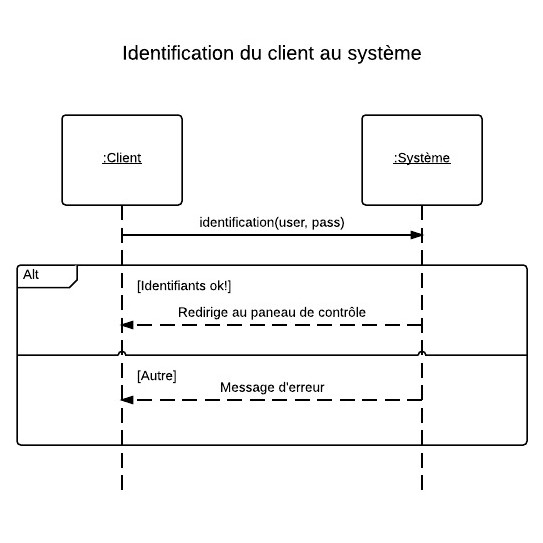
# Introduction

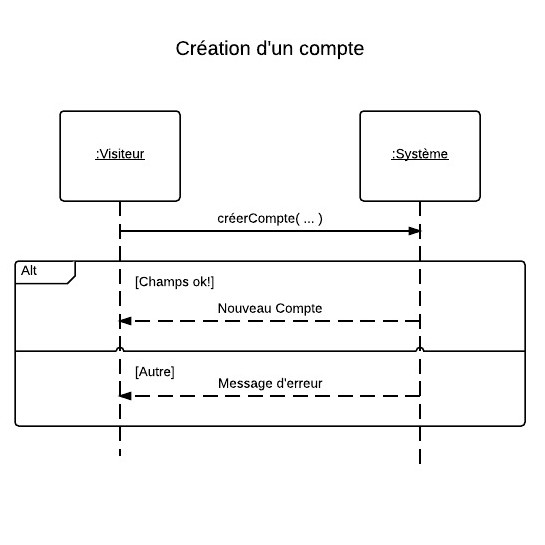
De plus en plus de consommateurs choisissent de faire affaire avec des entreprises qui offrent leurs services par l’intermédiaire du web. Que ce soit avec leurs tablettes, leurs téléphones intelligents ou même leurs portables, ils désirent pouvoir accéder facilement à leurs services favoris de la façon qui leur soit la plus conviviale. Ceci est vrai même dans la restauration. Pour cette raison, un entrepreneur nous a mandatés pour concevoir et implémenter un logiciel permettant la livraison de nourriture à domicile. Notre logiciel devra mettre en contact plusieurs restaurateurs avec des services de livraisons indépendants. Ensuite, les consommateurs n’auront donc plus qu’à commander sur notre site et ils recevront en très peu de temps leurs commandes. Le lien qui existe entre le service de livraison et le restaurant sera donc complètement transparent pour le consommateur. Pour cette première itération, nous avons choisi d'implémenter le « framework » Django pour la gestion de comptes des clients. Nous allons, dans un premier temps, présenter le modèle du domaine suivie des diagrammes de séquence système, des contrats d'opérations et enfin des diagrammes de séquences.

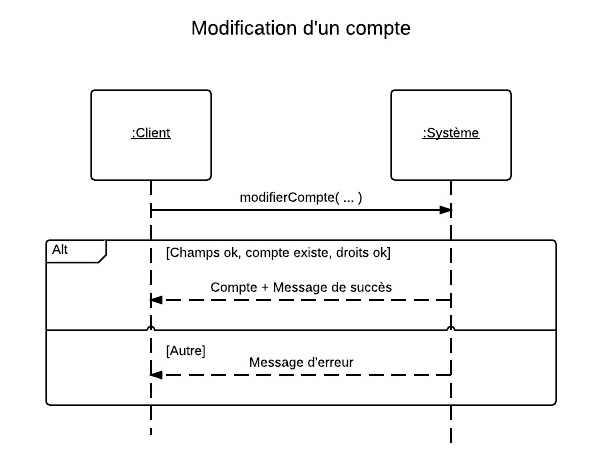
# MDD

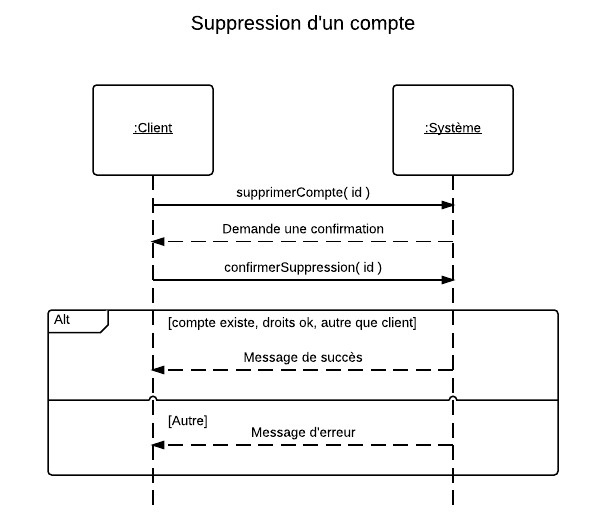


# DSS (Diagrammes de séquences systèmes)









# Contrats d’opérations

## Contrat CO1 : Identification du client au système.

|  |  |
| --- | --- |
| **Opération** | post(self, request, \*args, \*\*kwargs) |
| **Référence croisée** | Identification du client au système |
| **Préconditions** | Le compte existe |
| **Postconditions** | Un objet session a été créé.  L’attribut last\_login du compte a été modifié.  Une requête de redirection a été envoyée. |

## Contrat CO2 : Création d'un compte client

|  |  |
| --- | --- |
| **Opération** | post(self, request, \*args, \*\*kwargs) |
| **Référence croisée** | Création d'un compte |
| **Préconditions** | Aucune |
| **Postconditions** | Une instance de formulaire RegistrationForm form a été créée.  Les attributs de form sont remplis avec les valeurs de la requête request.  Une instance de formulaire RegistrationDetailsForm detailsForm a été créée.  Les attributs de detailsForm sont remplis avec les valeurs de la requête.  Un utilisateur a été créé avec les valeurs de form.  Le groupe 1 a été associé à l’utilisateur.  Les Détails de l’utilisateur ont été créés à partir du formulaire detailsForm.  Une réponse de type TemplateResponse a été retournée. |

## Contrat CO3 : Modification des données d'un compte

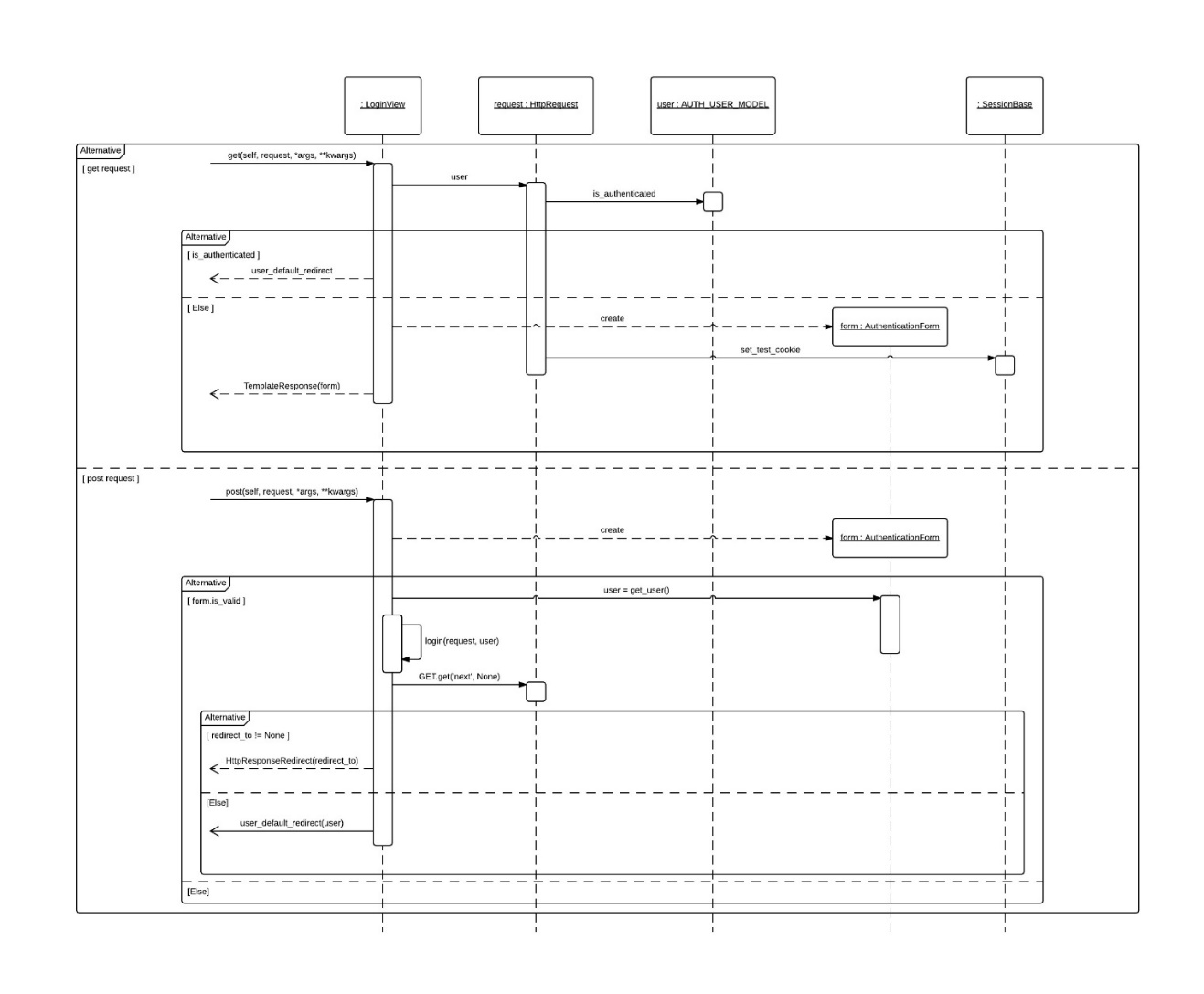
|  |  |
| --- | --- |
| **Opération** | post(self, request, \*args, \*\*kwargs) |
| **Référence croisée** | Modification d'un compte |
| **Préconditions** | Le compte existe et le client est authentifié |
| **Postconditions** | Les données du compte ont été récupérées dans la variable userDetails.  Le nom du formulaire a été récupéré dans la variable formName à partir des données de la requête request.  Si le formulaire est valide, les données sont mises à jour.  Une réponse de type TemplateResponse a été retournée. |

## Contrat CO4 : Suppression d'un compte

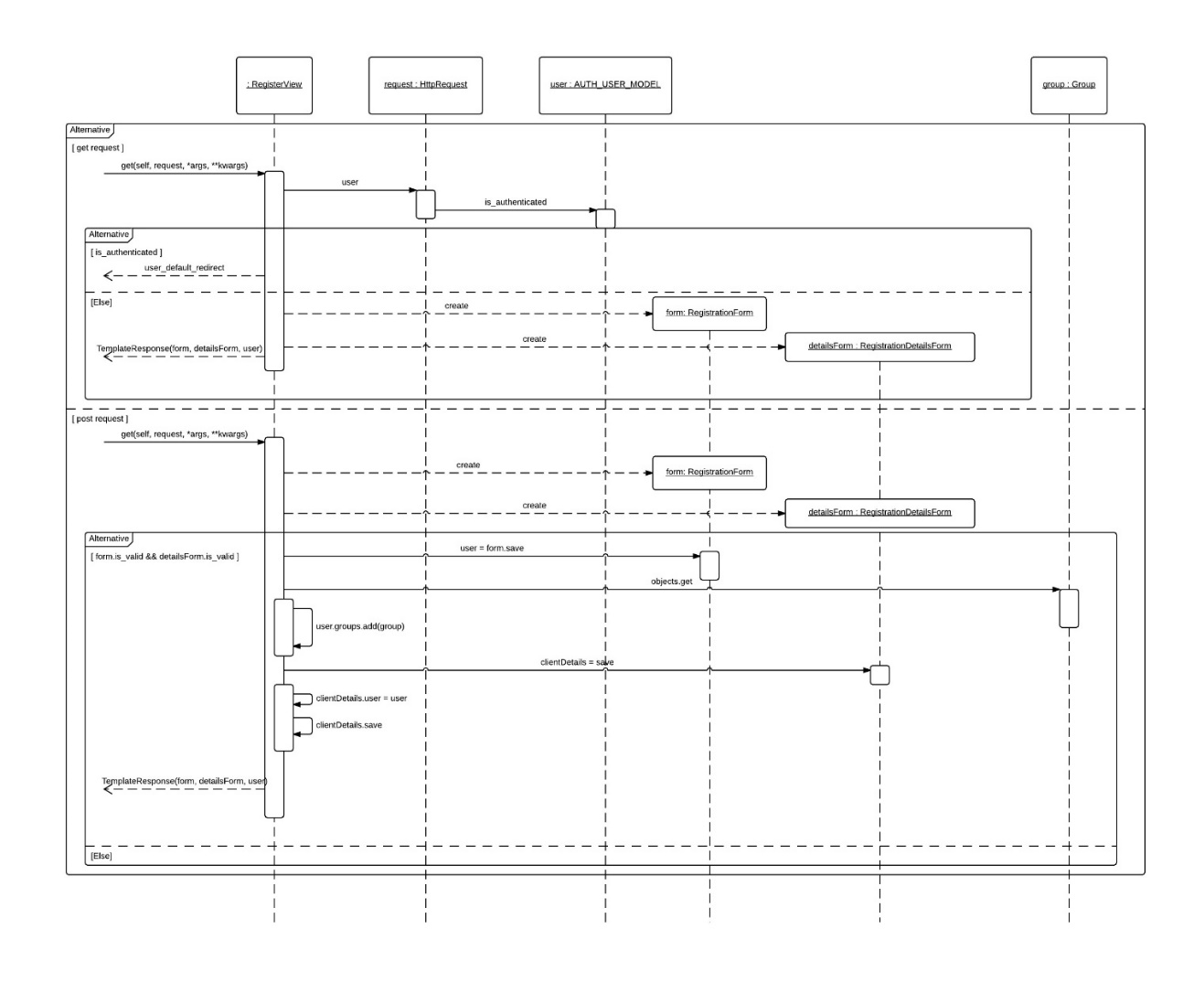
|  |  |
| --- | --- |
| **Opération** | post(self, request, \*args, \*\*kwargs) |
| **Référence croisée** | Suppression d'un compte |
| **Préconditions** | Le compte existe et le client est authentifié |
| **Postconditions** | Un objet objet a été créé.  Objet contient le compte à supprimer.  La méthode delete() a été appelé sur objet.  Le session a été détruite.  Une réponse de type TemplateResponse a été retournée. |

# DS (Diagrammes de séquences)

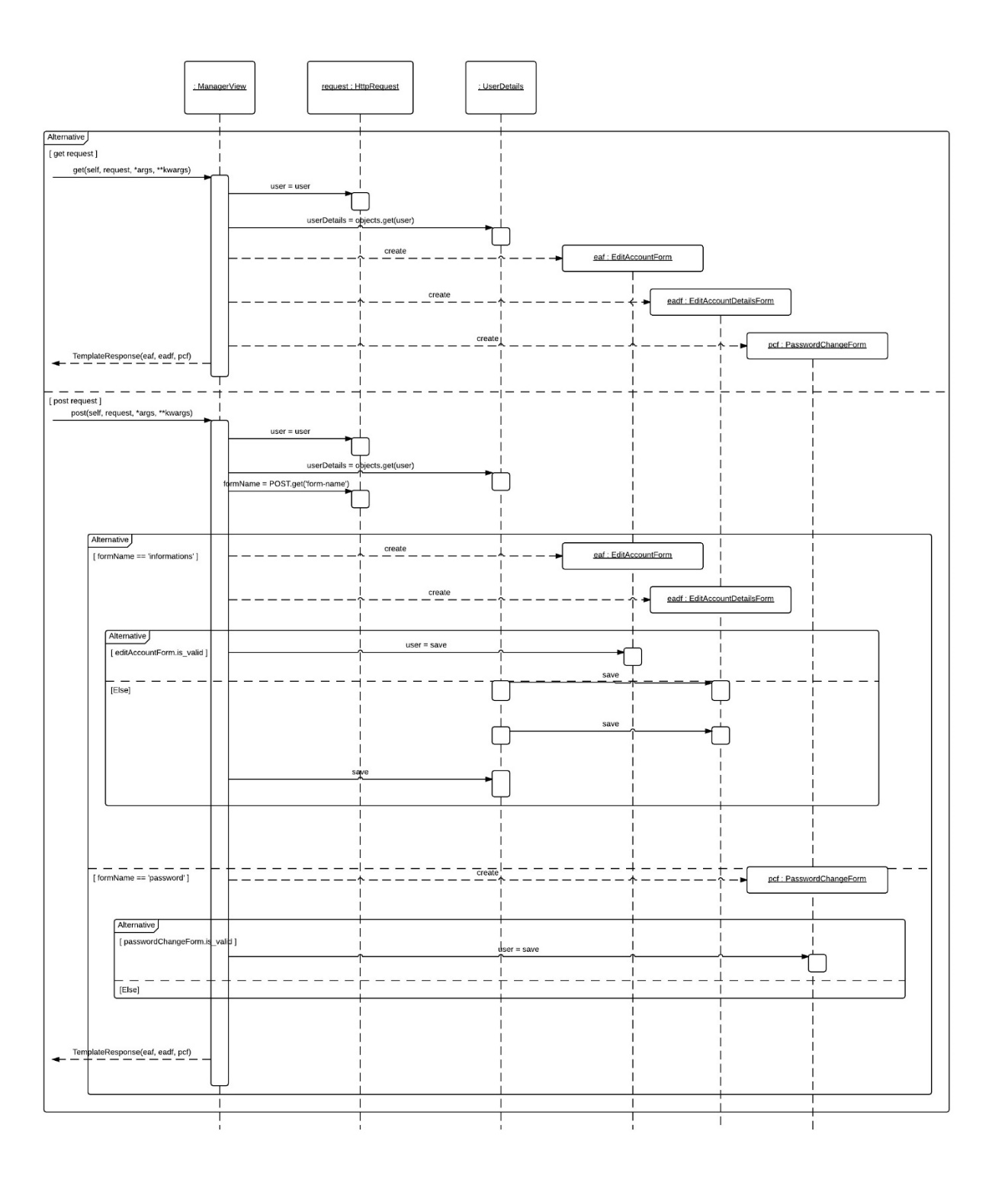
## Identification du client au système



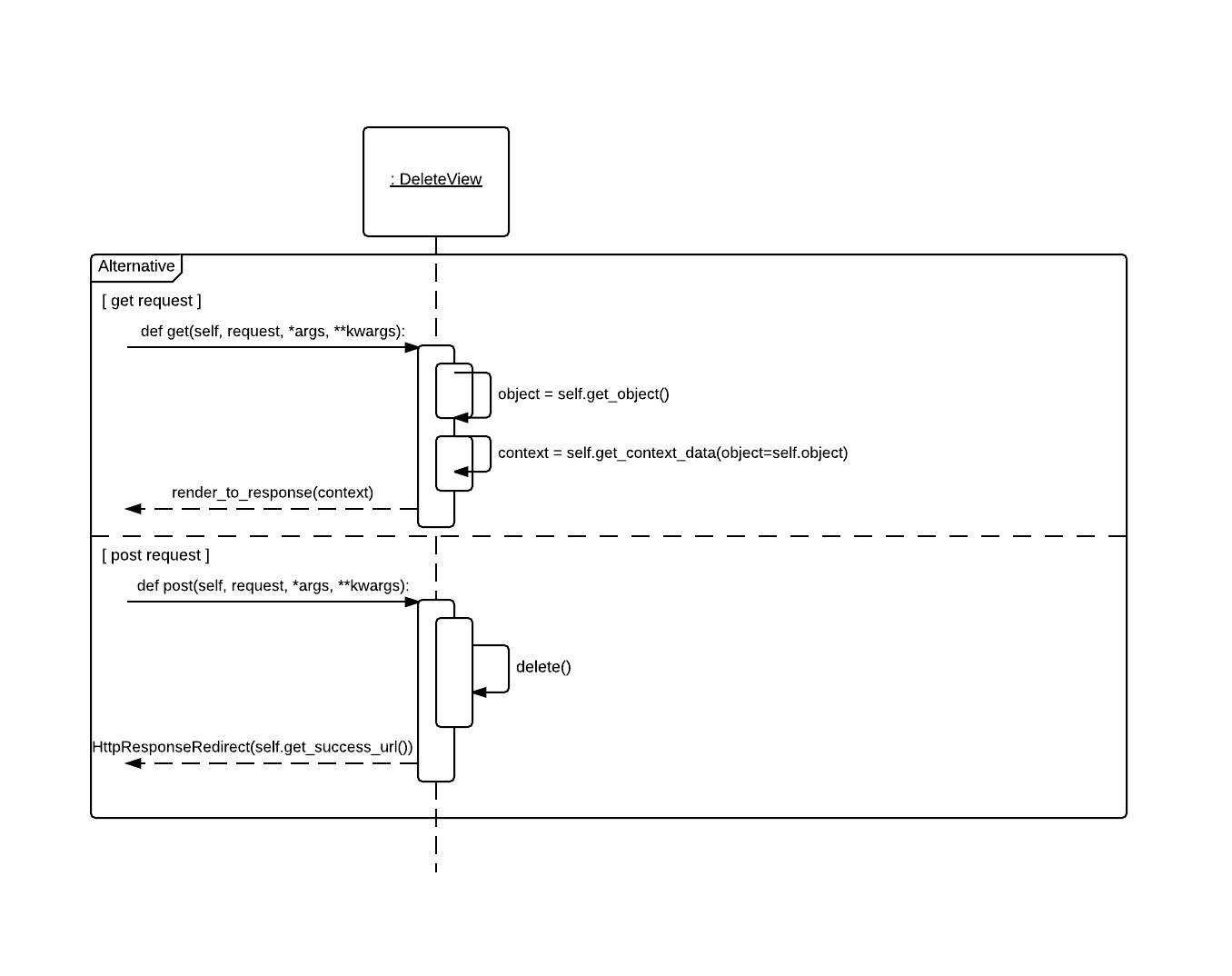
# Création d'un compte client

****

## Modification des données d'un compte

****

## Suppression d'un compte

****

# Conclusion

Durant cette première itération nous avons couvert toutes les fonctionnalités d’authentification, de création, de modification et de suppression d’un compte client. De plus, le « framework » Django a été implémenté. Étant donné que Django comprend déjà un module pour la gestion des usagers, seuls les détails tels que l’adresse, le numéro de téléphone, le pays, la ville, etc. ont dû être ajoutés. La prochaine itération couvrira la gestion des restaurants et des menus dans l’administration du site. Le tout dans le but de bien remplir notre objectif, c’est-à-dire : concevoir et implémenter un logiciel permettant la livraison de nourriture à domicile. Pour finir voici quelques points critiques à clarifier pour notre seconde itération :

* Choix des « templates », couleurs et du contenu par le restaurateur : « sites » de Django, intégration Django-CMS, questions de design, etc.
* Gestion de la devise, des taxes et rabais sur les « produits » (item de la commande)