

Rapport d'itération 1

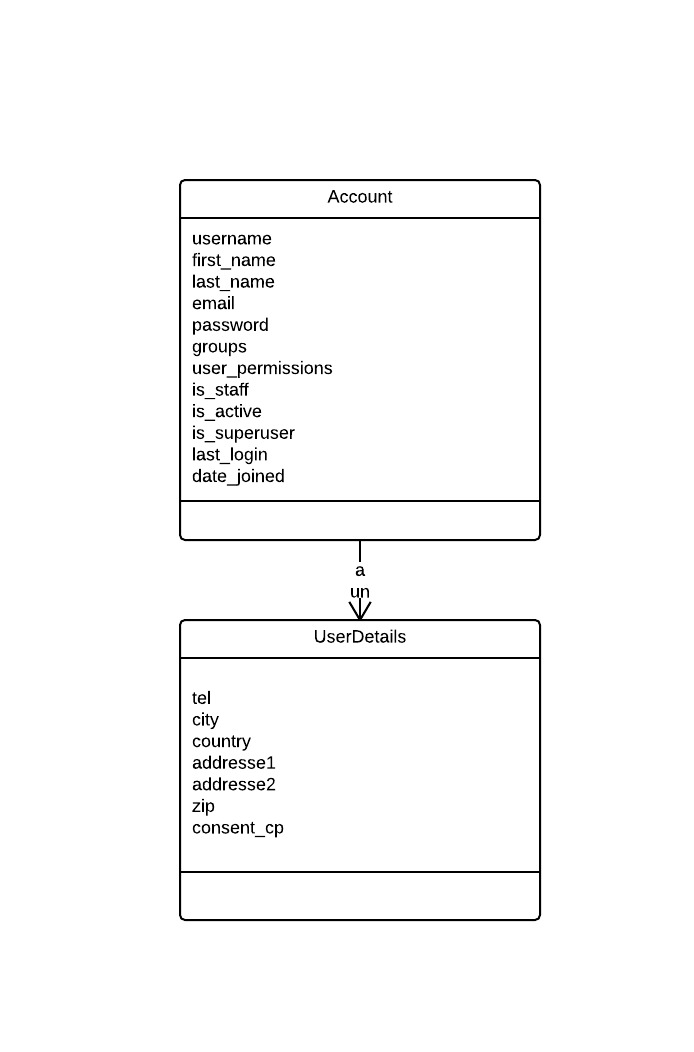
|  |  |
| --- | --- |
| **Laboratoire** | 01 |
| **Étudiants** | Dominic Roberge  André Koolen  Anthony Martin Coallier |
| **Codes permanents** | ROBD03129209  KOOA23039101  MARA19129107 |
| **Cours** | LOG210 |
| **Session** | Été 2015 |
| **Groupe** | 03 |
| **Professeur** | Yvan Ross |
| **Chargé de laboratoire** | Charbonneau |
| **Date** | 25-05-2015 |

# Introduction

Nous sommes à l'air du mobile et du web, prendre du recul maintenant quant à ces technologies est perdre une part de marché face aux compétiteurs. Ceci est vrai même dans la restauration, pour cette raison, nous nous lançons dans la conception d'un logiciel permettant la livraison de nourriture à domicile. Ce logiciel desservira plusieurs restaurants via des comptes restaurateurs, des clients pourront s’y créer un compte, payer et se faire livrer à domicile via le site.

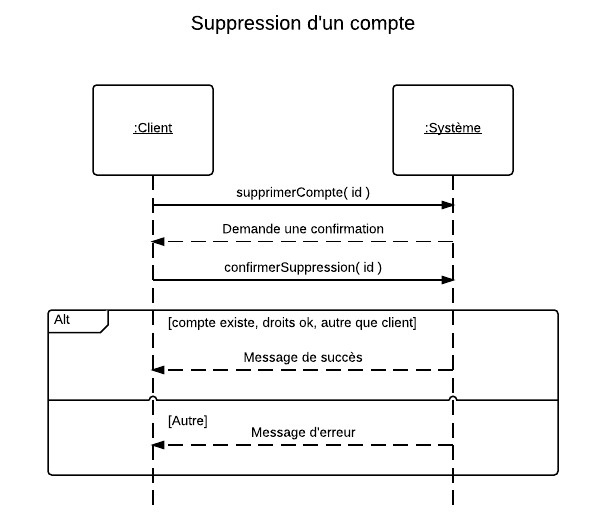
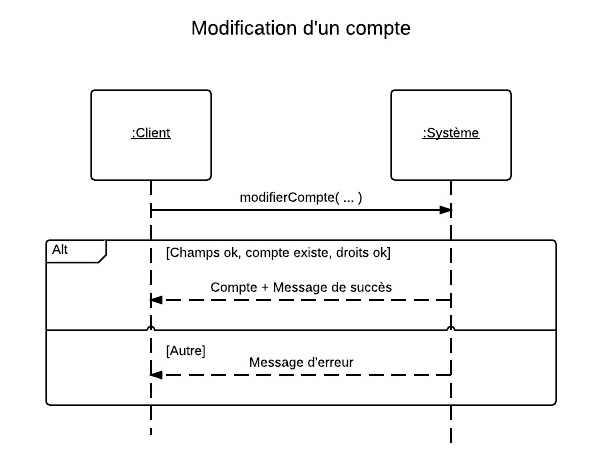
Pour la première itération, nous avons choisi d'implémenter le cadriciel Django pour la gestion de comptes et de droits d'administrations. Nous allons, dans un premier temps, présenter le modèle du domaine, les diagrammes de séquence système, les contrats d'opérations et pour finir les diagrammes de séquences.

# Modèle du domaine



# Diagramme séquence système

# 



# Contrats d'opérations

|  |  |
| --- | --- |
| **Opération** | identification(user:String, pass:String) |
| **Référence croisée** | Identification du client au système |
| **Préconditions** | Le compte existe |
| **Postconditions** | * Création d'un objet Session * Attribut last\_login modifié * Redirection vers la bonne page |

|  |  |
| --- | --- |
| **Opération** | créerCompte(user:String, pass:String, addresses:Liste d'adresses, nom:String, prénom:String, courriel:String) |
| **Référence croisée** | Création d'un compte |
| **Préconditions** | Aucune sauf pour les comptes restaurateurs qui nécessite l'authentification au compte super-admin. |
| **Postconditions** | * Un compte a été créé * Des objets Adresse sont créés * Un courriel à été envoyé à l'adresse du compte indiquée |

|  |  |
| --- | --- |
| **Opération** | modifierCompte(user:String, pass:String, adresses:Liste d'adresses, nom:String, prénom:String, courriel:String) |
| **Référence croisée** | Modification d'un compte |
| **Préconditions** | Le compte existe et le client est authentifié |
| **Postconditions** | * Les attributs du compte sont modifiés * Des objets Adresse sont créés/modifiés/supprimés |

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

|  |  |
| --- | --- |
| **Opération** | confirmerSupression(id:Int) |
| **Référence croisée** | Suppression d'un compte |
| **Préconditions** | Le compte existe |
| **Postconditions** | * L'attribut du compte is\_active est mis à False |

# Diagramme de séquence

# Identification du client

# C:\Users\Anthony\AppData\Local\Microsoft\Windows\INetCache\Content.Word\identification - New Page.jpeg

# Création du compte

# C:\Users\Anthony\AppData\Local\Microsoft\Windows\INetCache\Content.Word\creation - New Page (1).jpeg

# Modification du compte

# C:\Users\Anthony\AppData\Local\Microsoft\Windows\INetCache\Content.Word\modification - New Page.jpeg

# Suppression du compte

# C:\Users\Anthony\AppData\Local\Microsoft\Windows\INetCache\Content.Word\delete - New Page.jpeg

# Conclusion

# Durant cette première itération nous avons couvert toute les fonctionnalités d’authentification/création/modification de compte client tout en mettant en place le cadriciel et le projet de façon générale.

# Comme Django offrait déjà beaucoup de code et de classes pour la gestion des usagers, nous n’avons eu qu’à faire un modèle de données pour les détails tel que ; adresse, téléphone, pays, ville, etc. La gestion des groupes de permissions était déjà inclus dans le CMS intégré de Django ; les nouveau clients auront leur groupe et les restaurateurs le leur.

# La prochaine itération couvrira la gestion des restaurants et des menus dans l’administration du site. Pour finir voici quelques points critiques à clarifier pour celle-ci ;

# 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)