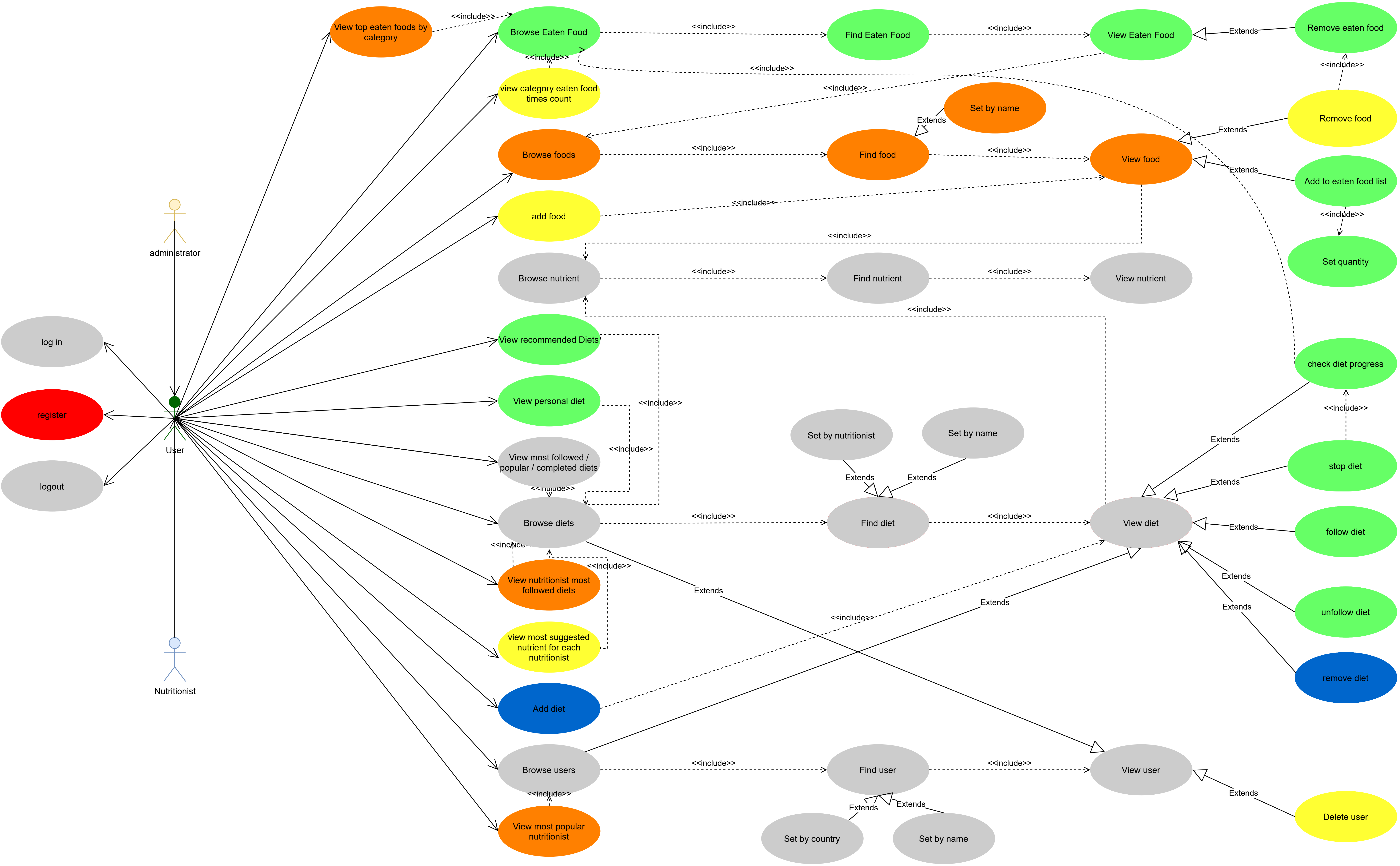
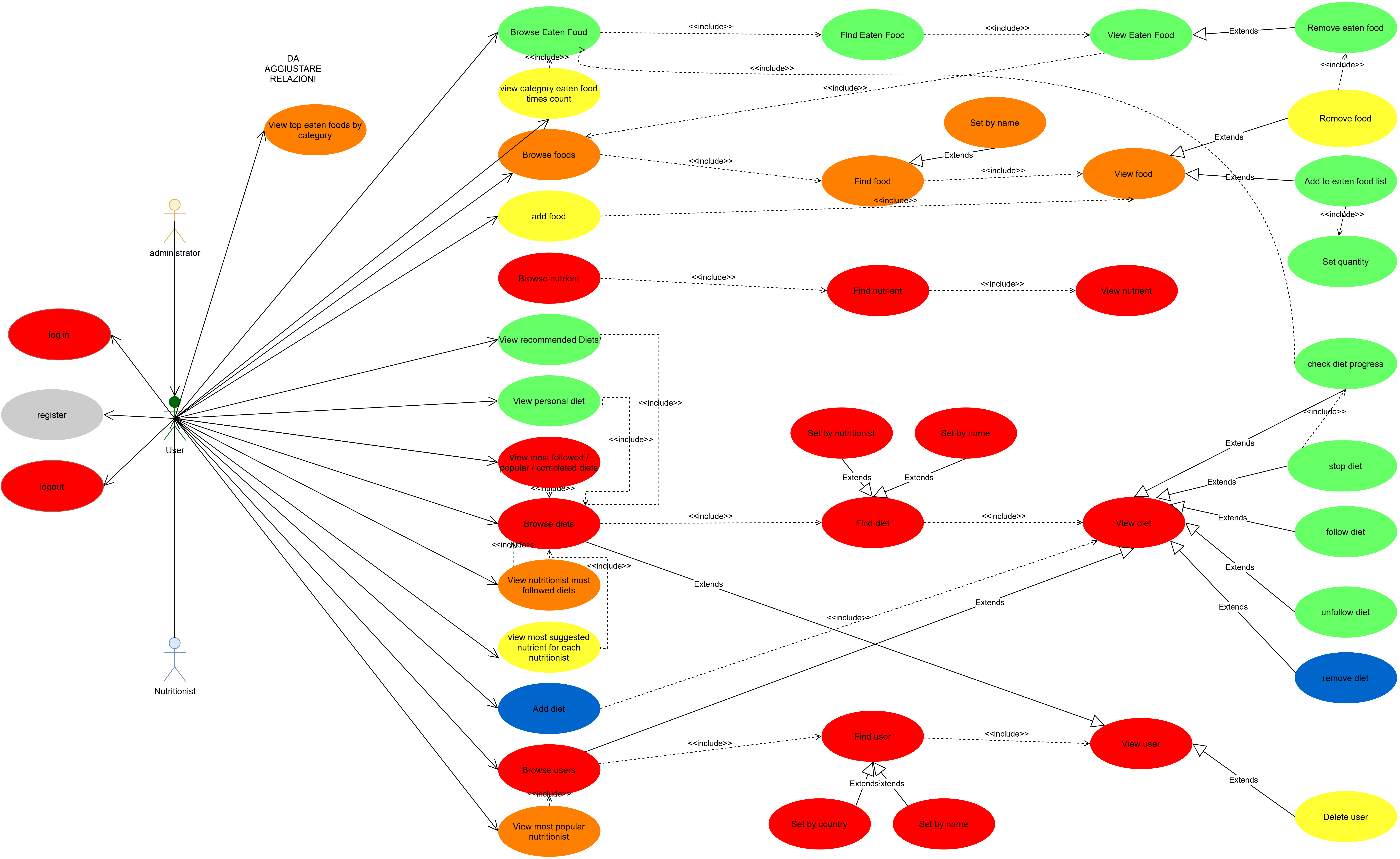
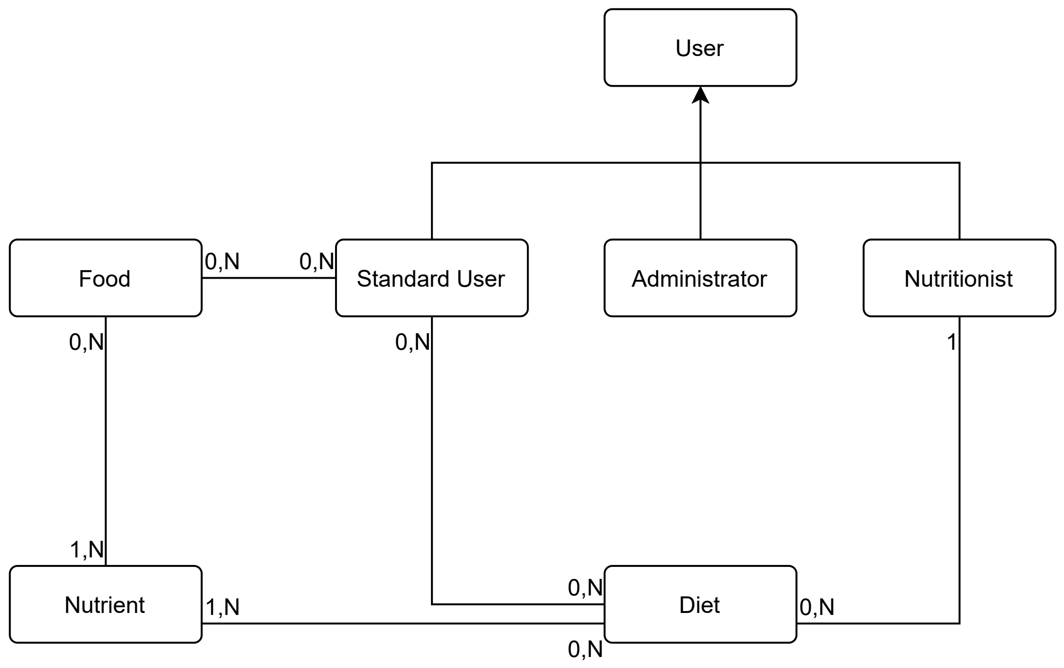
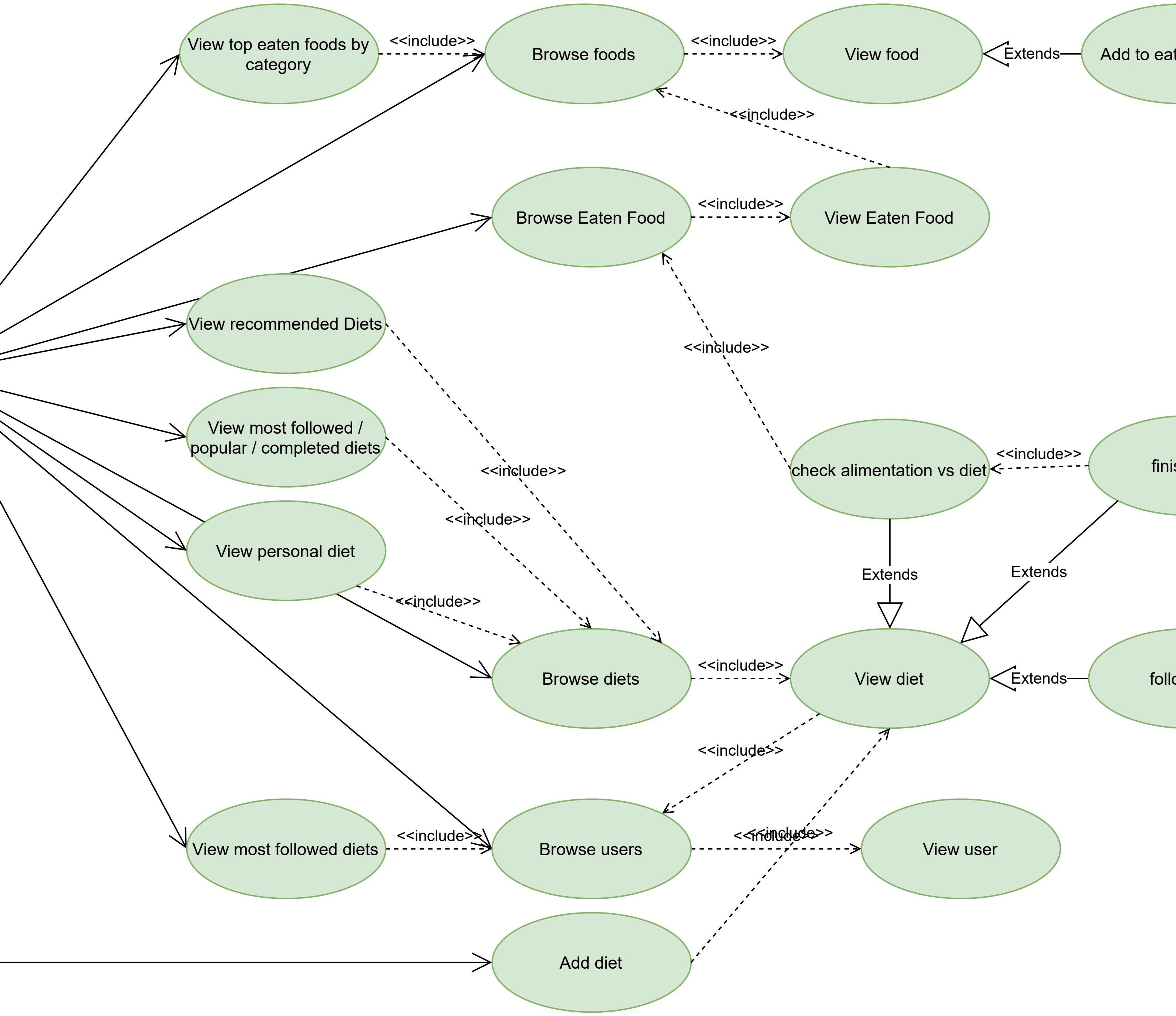


administrator
User
Nutritionist
log in
register
logout
View top eaten foods by category
Browse Eaten Food
Find Eaten Food
View Eaten Food
Remove eaten food
view category eaten food times count
Set by name
Browse foods
Find food
View food
Remove food
Add to eaten food list
Set quantity
add food
Browse nutrient
Find nutrient
View nutrient
View recommended Diets
View personal diet
View most followed / popular / completed diets
check diet progress
stop diet
Set by nutritionist
Set by name
Browse diets
Find diet
View diet
follow diet
unfollow diet
remove diet
View nutritionist most followed diets
view most suggested nutrient for each nutritionist
Add diet
Browse users
Find user
View user
Delete user
View most popular nutritionist
Set by country
Set by name
<<include>>
Extends

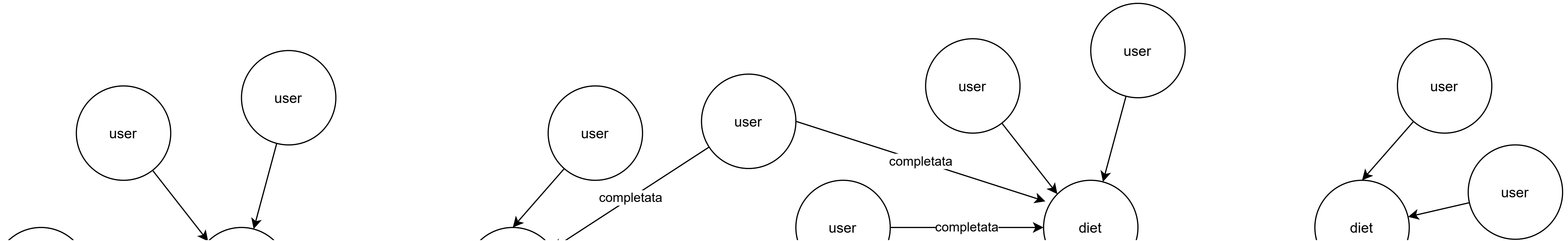

DA
AGGIUSTARE
RELAZIONI
administrator
User
Nutritionist
log in
register
logout
View top eaten foods by category
Browse Eaten Food
view category eaten food times count
Browse foods
add food
Browse nutrient
View recommended Diets
View personal diet
View most followed / popular / completed diets
Browse diets
View nutritionist most followed diets
view most suggested nutrient for each nutritionist
Add diet
Browse users
View most popular nutritionist
Find Eaten Food
View Eaten Food
Remove eaten food
Set by name
Find food
View food
Remove food
Add to eaten food list
Set quantity
Find nutrient
View nutrient
check diet progress
Set by nutritionist
Set by name
Find diet
View diet
stop diet
follow diet
unfollow diet
remove diet
Find user
View user
Set by country
Set by name
Delete user
<<include>>
Extends

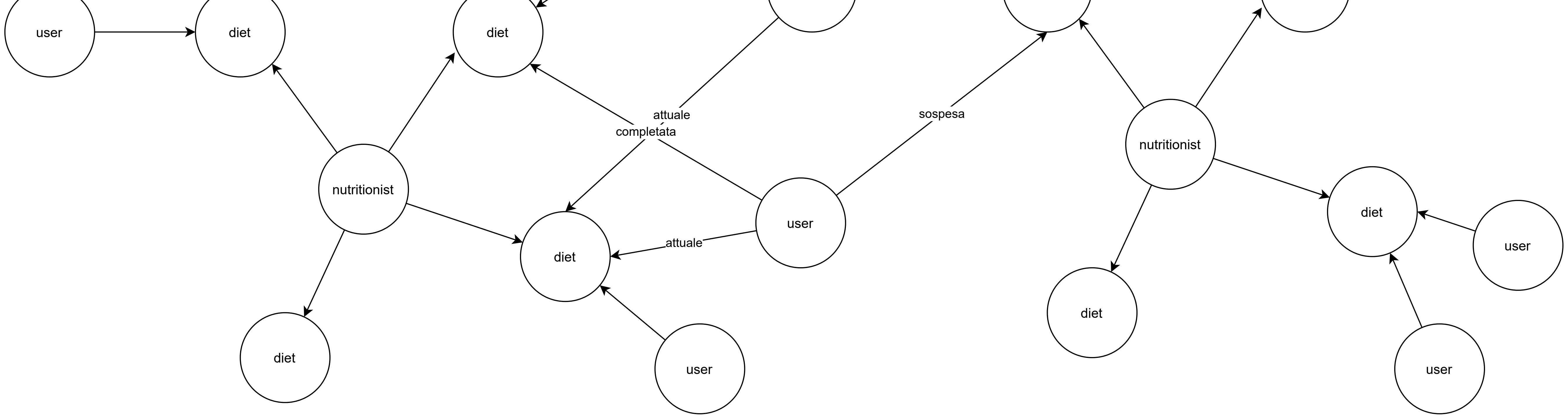

User
Food
Standard User
Administrator
Nutritionist
Nutrient
Diet
0,N
0,N
0,N
0,N
1
1,N
1,N
0,N
0,N
0,N

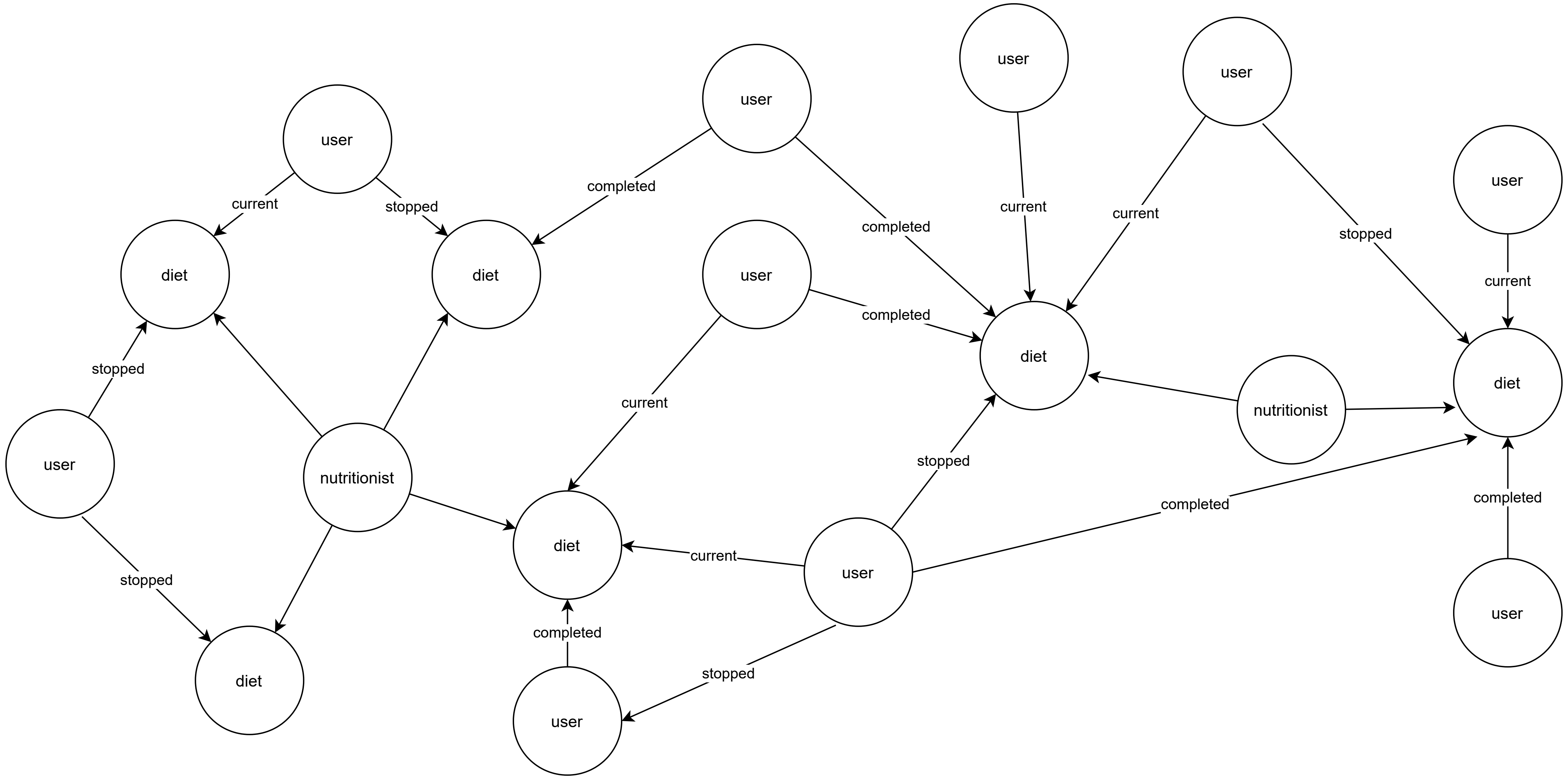
administrator
User
Nutritionist

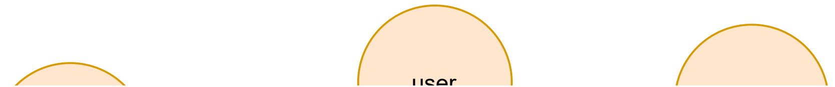

View top eaten foods by category
<<include>>
Browse foods
<<include>>
View food
Extends
Add to eat
<<include>>
Browse Eaten Food
<<include>>
View Eaten Food
View recommended Diets
<<include>>
View most followed / popular / completed diets
<<include>>
<<include>>
check alimentation vs diet
<<include>>
finis
View personal diet
<<include>>
Extends
Extends
Browse diets
<<include>>
View diet
Extends
follo
<<include>>
View most followed diets
<<include>>
Browse users
<<include>>
<<include>>
View user
Add diet

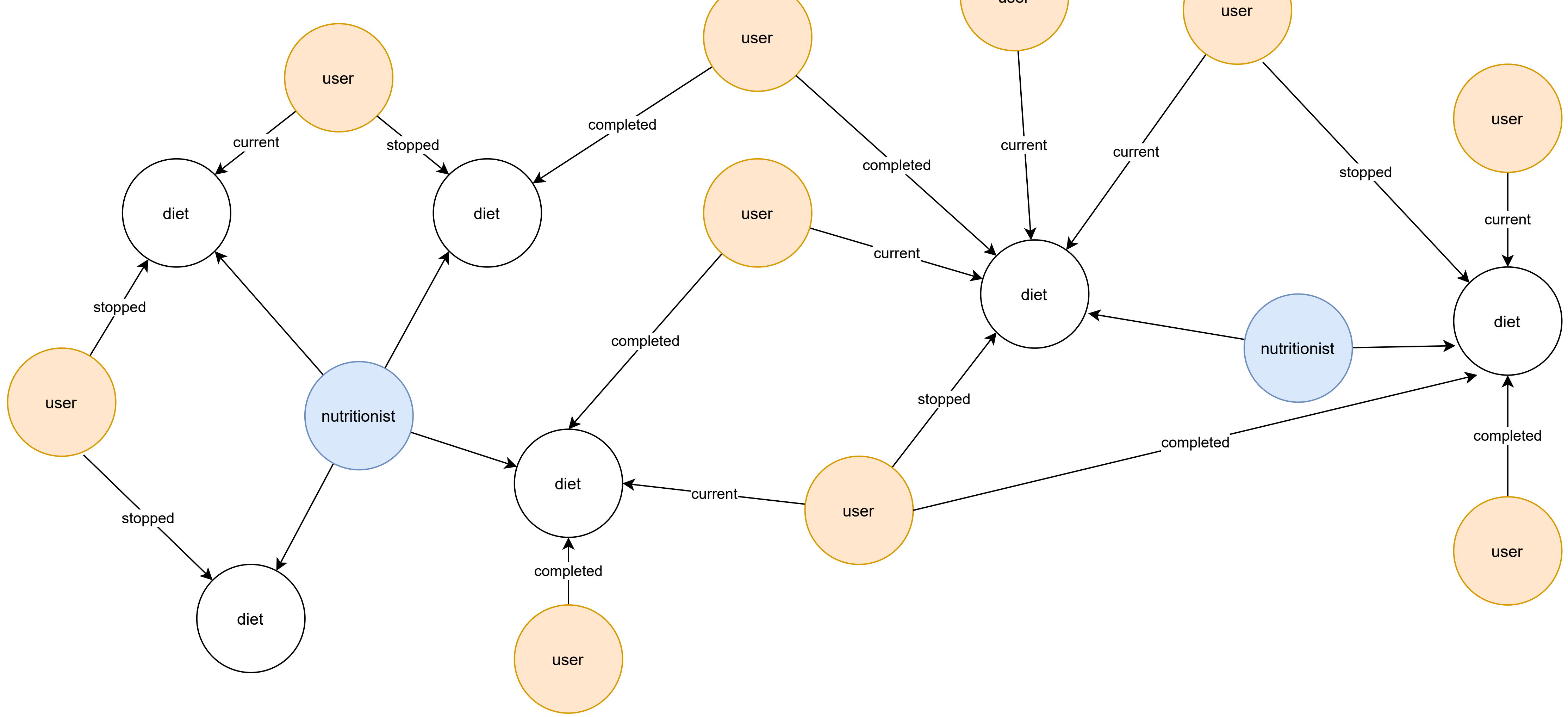
ten food list
sh diet
ow diet

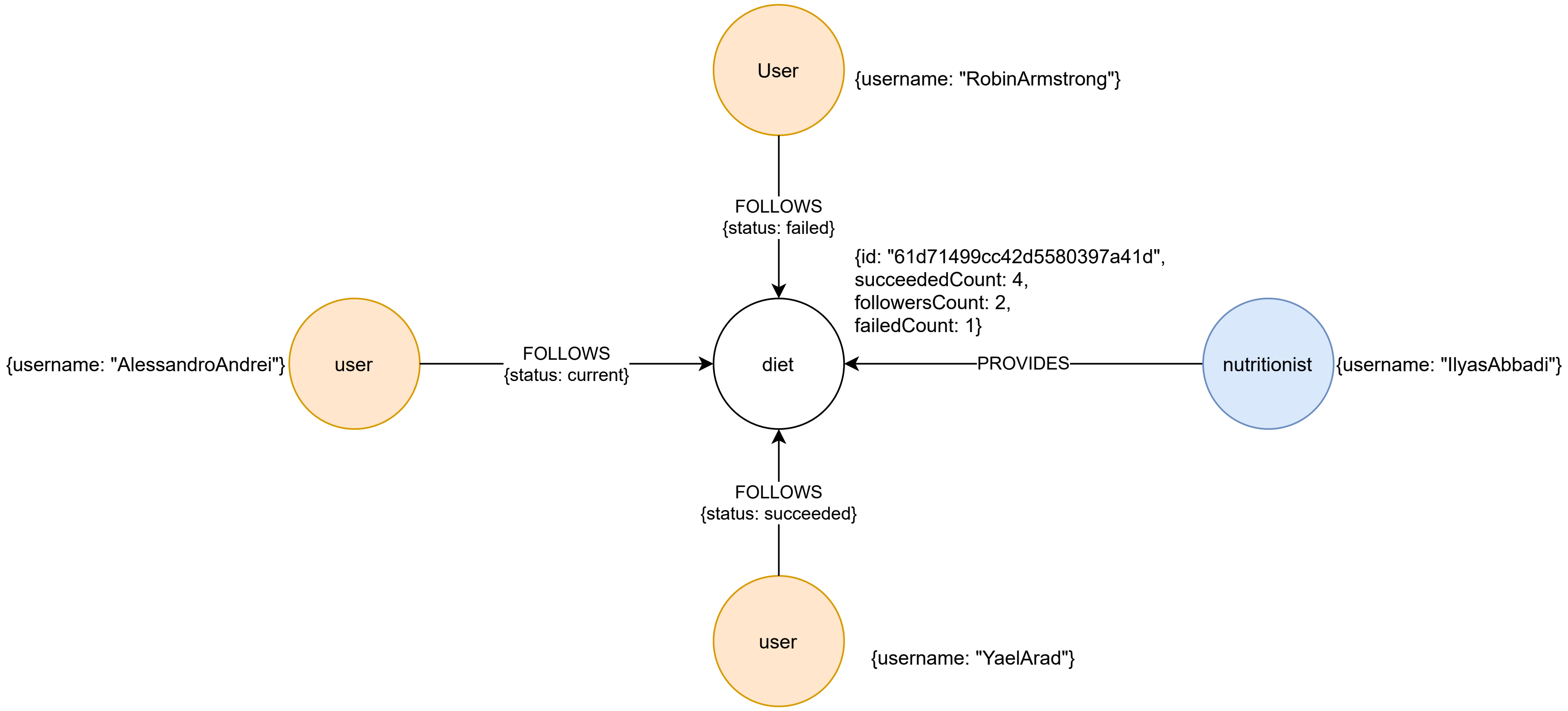

user
user
user
user
completata
user
user
user
completata
user
completata
diet
user
diet
user

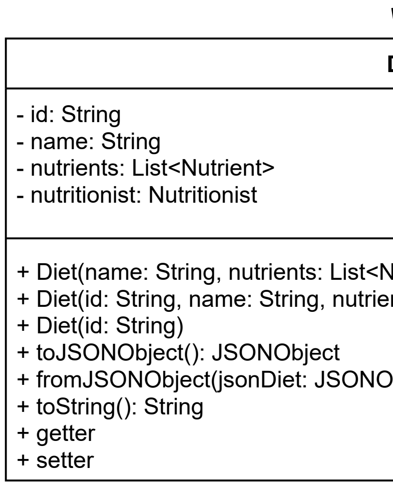

user
diet
diet
nutritionist
attuale
completata
sospesa
nutritionist
diet
attuale
user
diet
user
diet
user
diet
user


user
user
user
user
current
stopped
completed
completed
current
current
stopped
user
current
diet
diet
user
completed
diet
diet
stopped
current
stopped
nutritionist
nutritionist
user
stopped
completed
diet
current
user
completed
completed
stopped
diet
completed
user
stopped
user


user


user
user
user
user
user
completed
current
stopped
completed
current
current
stopped
diet
diet
user
current
diet
user
current
diet
stopped
completed
nutritionist
user
nutritionist
stopped
completed
diet
current
user
completed
stopped
diet
completed
user
user

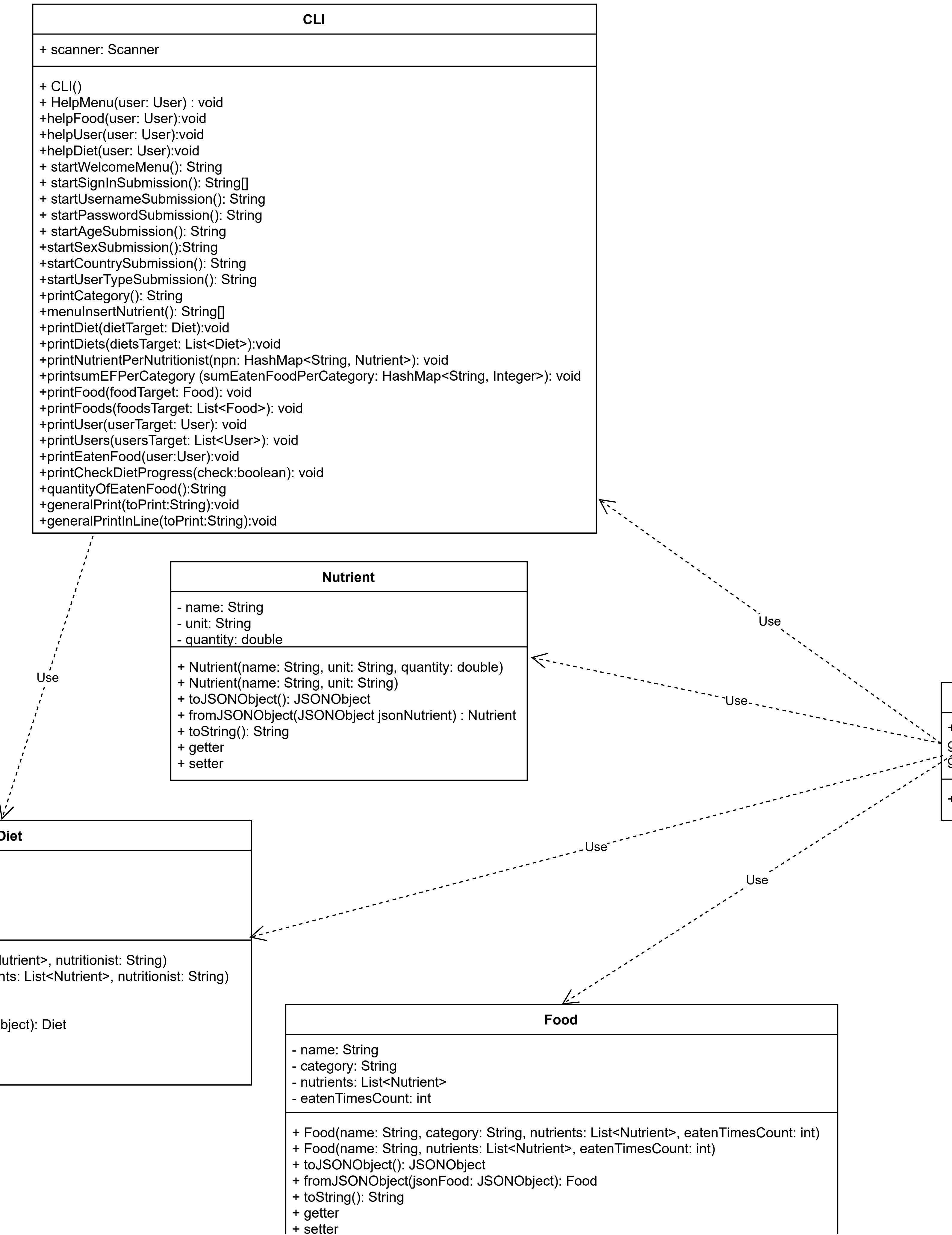

User
{username: "RobinArmstrong"}
FOLLOWS
{status: failed}
{id: "61d71499cc42d5580397a41d",
succeededCount: 4,
followersCount: 2,
failedCount: 1}
{username: "AlessandroAndrei"}
user
FOLLOWS
{status: current}
diet
PROVIDES
nutritionist
{username: "IlyasAbbadi"}
FOLLOWS
{status: succeeded}
user
{username: "YaelArad"}



D

- id: String
- name: String
- nutrients: List<Nutrient>
- nutritionist: Nutritionist

+ Diet(name: String, nutrients: List<N
+ Diet(id: String, name: String, nutrier
+ Diet(id: String)
+ toJSONObject(): JSONObject
+ fromJSONObject(jsonDiet: JSONO
+ toString(): String
+ getter
+ setter



CLI

+ scanner: Scanner

+ CLI()
+ HelpMenu(user: User) : void
+helpFood(user: User):void
+helpUser(user: User):void
+helpDiet(user: User):void
+ startWelcomeMenu(): String
+ startSignInSubmission(): String[]
+ startUsernameSubmission(): String
+ startPasswordSubmission(): String
+ startAgeSubmission(): String
+startSexSubmission():String
+startCountrySubmission(): String
+startUserTypeSubmission(): String
+printCategory(): String
+menuInsertNutrient(): String[]
+printDiet(dietTarget: Diet):void
+printDiets(dietsTarget: List<Diet>):void
+printNutrientPerNutritionist(npn: HashMap<String, Nutrient>): void
+printsumEFPerCategory (sumEatenFoodPerCategory: HashMap<String, Integer>): void
+printFood(foodTarget: Food): void
+printFoods(foodsTarget: List<Food>): void
+printUser(userTarget: User): void
+printUsers(usersTarget: List<User>): void
+printEatenFood(user:User):void
+printCheckDietProgress(check:boolean): void
+quantityOfEatenFood():String
+generalPrint(toPrint:String):void
+generalPrintInLine(toPrint:String):void

Nutrient

- name: String
- unit: String
- quantity: double

+ Nutrient(name: String, unit: String, quantity: double)
+ Nutrient(name: String, unit: String)
+ toJSONObject(): JSONObject
+ fromJSONObject(JSONObject jsonNutrient) : Nutrient
+ toString(): String
+ getter
+ setter

Diet

lutrient>, nutritionist: String)
nts: List<Nutrient>, nutritionist: String)

bject): Diet

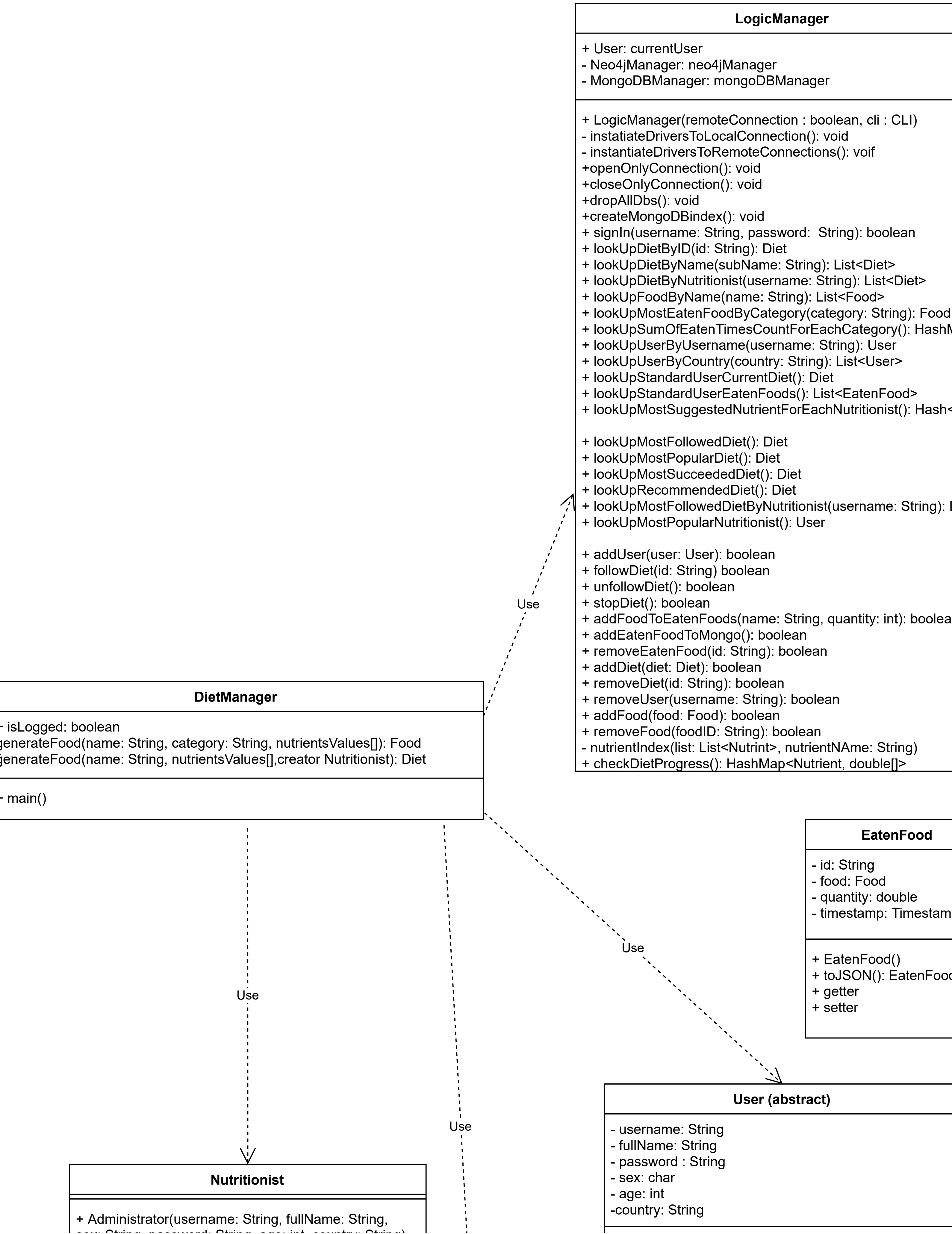
+
g
g

+

Food

- name: String
- category: String
- nutrients: List<Nutrient>
- eatenTimesCount: int

+ Food(name: String, category: String, nutrients: List<Nutrient>, eatenTimesCount: int)
+ Food(name: String, nutrients: List<Nutrient>, eatenTimesCount: int)
+ toJSONObject(): JSONObject
+ fromJSONObject(jsonFood: JSONObject): Food
+ toString(): String
+ getter
+ setter



LogicManager

+ User: currentUser
- Neo4jManager: neo4jManager
- MongoDBManager: mongoDBManager

+ LogicManager(remoteConnection : boolean, cli : CLI)
- instatiateDriversToLocalConnection(): void
- instantiateDriversToRemoteConnections(): voif
+openOnlyConnection(): void
+closeOnlyConnection(): void
+dropAllDbs(): void
+createMongoDBindex(): void
+ signIn(username: String, password: String): boolean
+ lookUpDietByID(id: String): Diet
+ lookUpDietByName(subName: String): List<Diet>
+ lookUpDietByNutritionist(username: String): List<Diet>
+ lookUpFoodByName(name: String): List<Food>
+ lookUpMostEatenFoodByCategory(category: String): Food
+ lookUpSumOfEatenTimesCountForEachCategory(): HashM
+ lookUpUserByUsername(username: String): User
+ lookUpUserByCountry(country: String): List<User>
+ lookUpStandardUserCurrentDiet(): Diet
+ lookUpStandardUserEatenFoods(): List<EatenFood>
+ lookUpMostSuggestedNutrientForEachNutritionist(): Hash<

+ lookUpMostFollowedDiet(): Diet
+ lookUpMostPopularDiet(): Diet
+ lookUpMostSucceededDiet(): Diet
+ lookUpRecommendedDiet(): Diet
+ lookUpMostFollowedDietByNutritionist(username: String): I
+ lookUpMostPopularNutritionist(): User

+ addUser(user: User): boolean
+ followDiet(id: String) boolean
+ unfollowDiet(): boolean
+ stopDiet(): boolean
+ addFoodToEatenFoods(name: String, quantity: int): boolea
+ addEatenFoodToMongo(): boolean
+ removeEatenFood(id: String): boolean
+ addDiet(diet: Diet): boolean
+ removeDiet(id: String): boolean
+ removeUser(username: String): boolean
+ addFood(food: Food): boolean
+ removeFood(foodID: String): boolean
- nutrientIndex(list: List<Nutrint>, nutrientNAme: String)
+ checkDietProgress(): HashMap<Nutrient, double[]>

DietManager

- isLogged: boolean
generateFood(name: String, category: String, nutrientsValues[]): Food
generateFood(name: String, nutrientsValues[],creator Nutritionist): Diet

- main()

EatenFood

- id: String
- food: Food
- quantity: double
- timestamp: Timestam

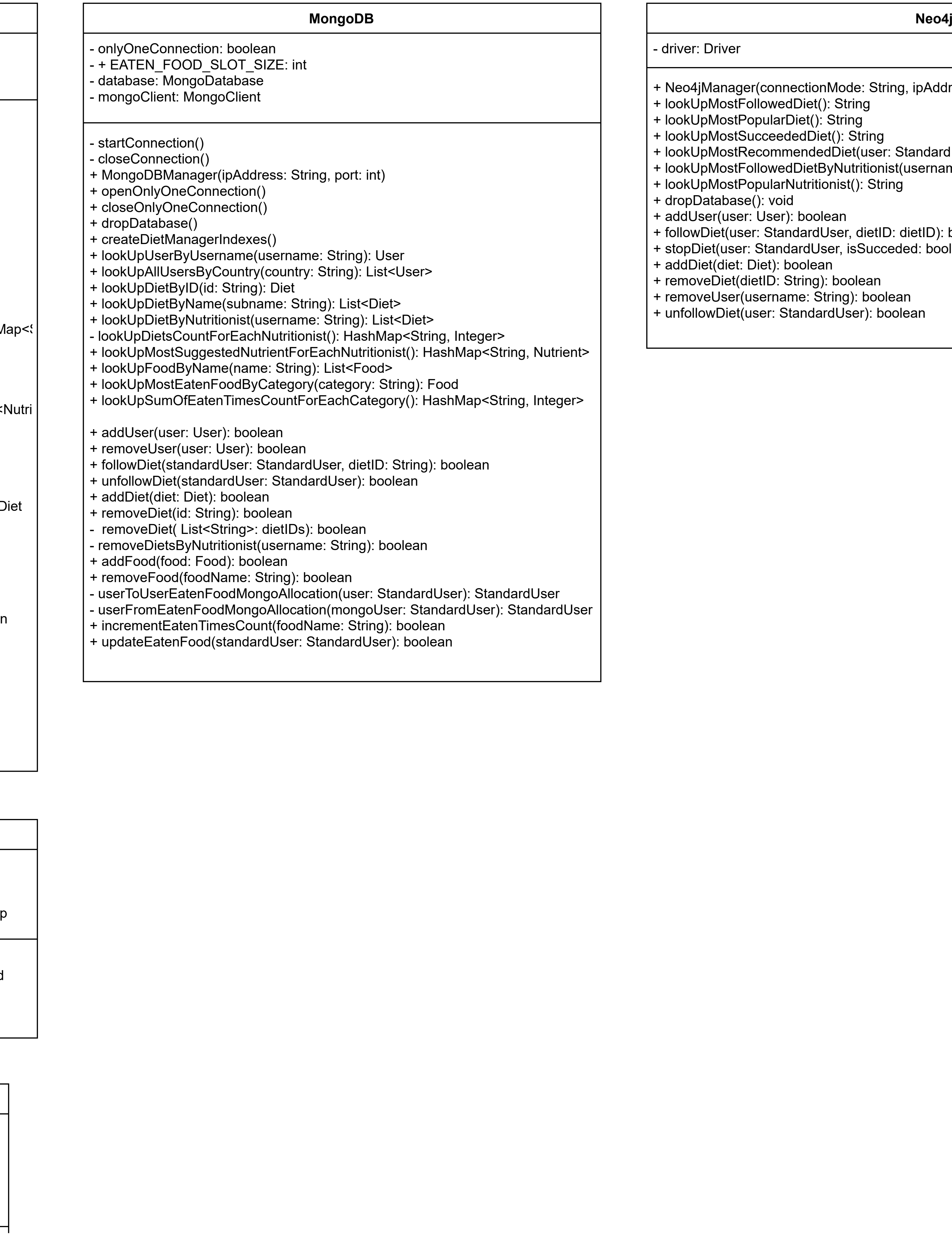
+ EatenFood()
+ toJSON(): EatenFood
+ getter
+ setter

User (abstract)

- username: String
- fullName: String
- password : String
- sex: char
- age: int
-country: String

Nutritionist

+ Administrator(username: String, fullName: String,
sex: String, password: String, age: int, country: String)



Map<S

<Nutri

Diet

n

p

d

MongoDB

- onlyOneConnection: boolean
- + EATEN_FOOD_SLOT_SIZE: int
- database: MongoDatabase
- mongoClient: MongoClient

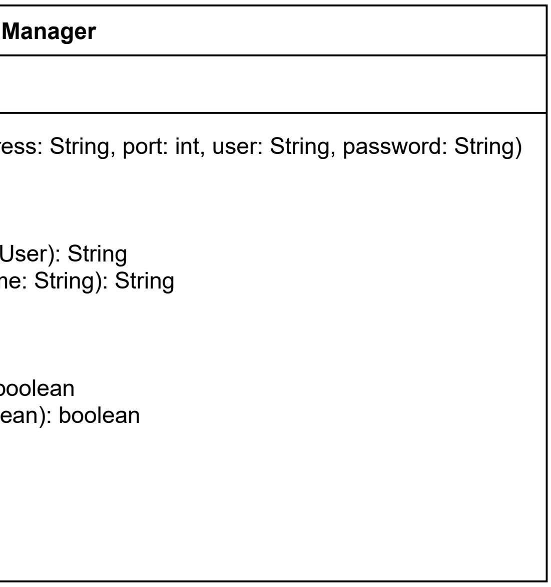
- startConnection()
- closeConnection()
+ MongoDBManager(ipAddress: String, port: int)
+ openOnlyOneConnection()
+ closeOnlyOneConnection()
+ dropDatabase()
+ createDietManagerIndexes()
+ lookUpUserByUsername(username: String): User
+ lookUpAllUsersByCountry(country: String): List<User>
+ lookUpDietByID(id: String): Diet
+ lookUpDietByName(subname: String): List<Diet>
+ lookUpDietByNutritionist(username: String): List<Diet>
- lookUpDietsCountForEachNutritionist(): HashMap<String, Integer>
+ lookUpMostSuggestedNutrientForEachNutritionist(): HashMap<String, Nutrient>
+ lookUpFoodByName(name: String): List<Food>
+ lookUpMostEatenFoodByCategory(category: String): Food
+ lookUpSumOfEatenTimesCountForEachCategory(): HashMap<String, Integer>

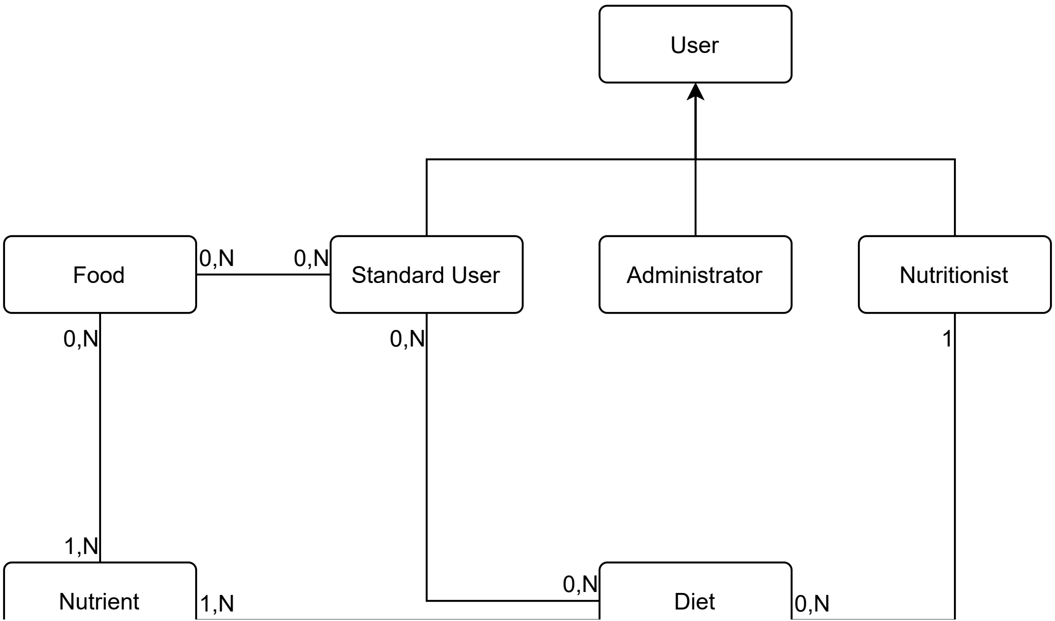
+ addUser(user: User): boolean
+ removeUser(user: User): boolean
+ followDiet(standardUser: StandardUser, dietID: String): boolean
+ unfollowDiet(standardUser: StandardUser): boolean
+ addDiet(diet: Diet): boolean
+ removeDiet(id: String): boolean
- removeDiet(List<String>: dietIDs): boolean
- removeDietsByNutritionist(username: String): boolean
+ addFood(food: Food): boolean
+ removeFood(foodName: String): boolean
- userToUserEatenFoodMongoAllocation(user: StandardUser): StandardUser
- userFromEatenFoodMongoAllocation(mongoUser: StandardUser): StandardUser
+ incrementEatenTimesCount(foodName: String): boolean
+ updateEatenFood(standardUser: StandardUser): boolean

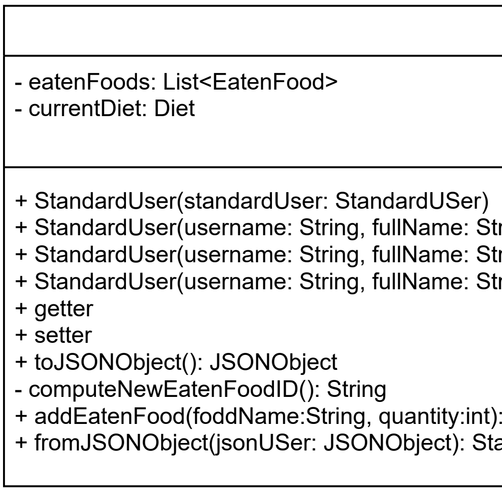
Neo4j

- driver: Driver

+ Neo4jManager(connectionMode: String, ipAddr
+ lookUpMostFollowedDiet(): String
+ lookUpMostPopularDiet(): String
+ lookUpMostSucceededDiet(): String
+ lookUpMostRecommendedDiet(user: Standard
+ lookUpMostFollowedDietByNutritionist(usernam
+ lookUpMostPopularNutritionist(): String
+ dropDatabase(): void
+ addUser(user: User): boolean
+ followDiet(user: StandardUser, dietID: dietID): b
+ stopDiet(user: StandardUser, isSucceded: bool
+ addDiet(diet: Diet): boolean
+ removeDiet(dietID: String): boolean
+ removeUser(username: String): boolean
+ unfollowDiet(user: StandardUser): boolean

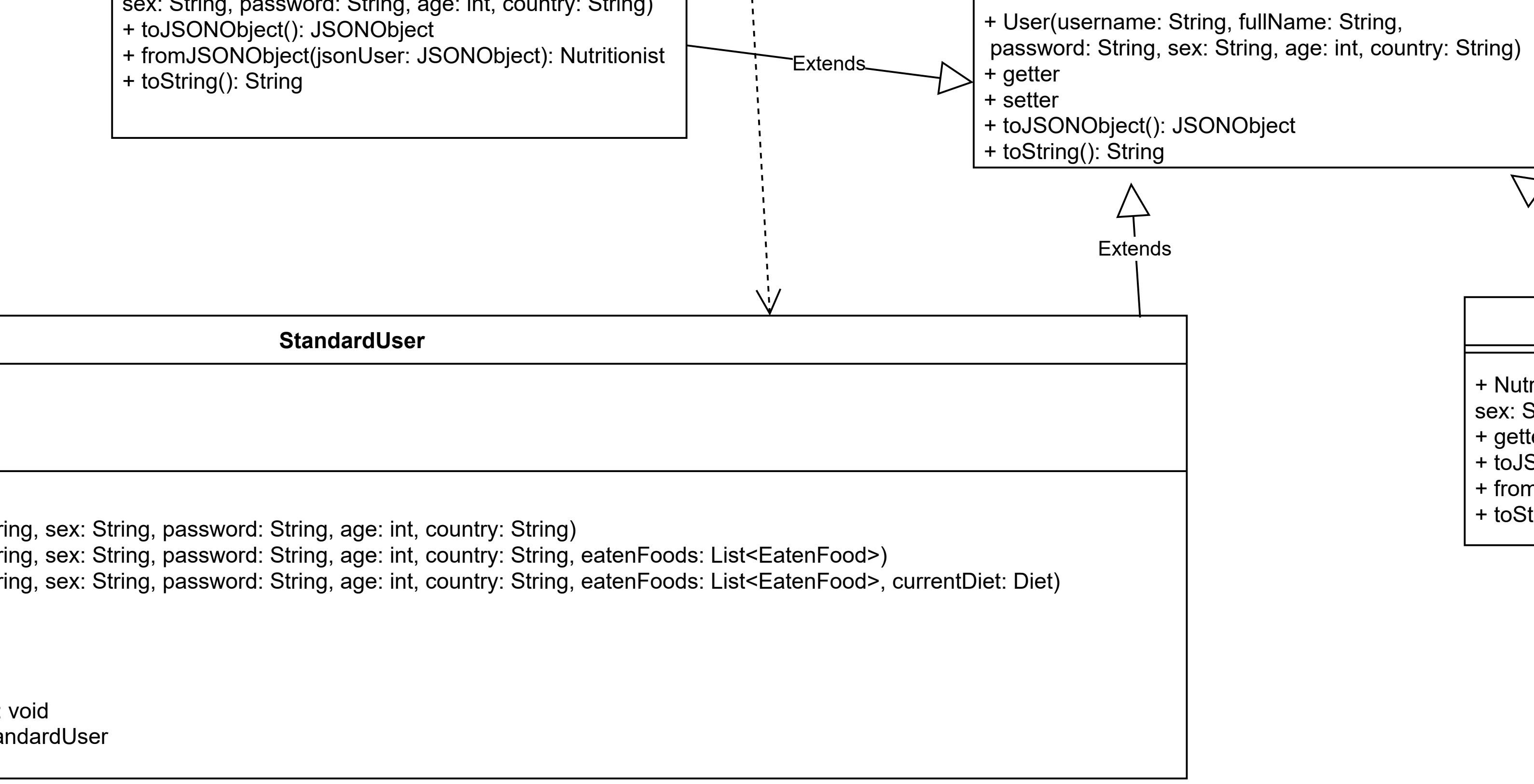

Manager
ress: String, port: int, user: String, password: String)
User): String
ne: String): String
boolean
ean): boolean

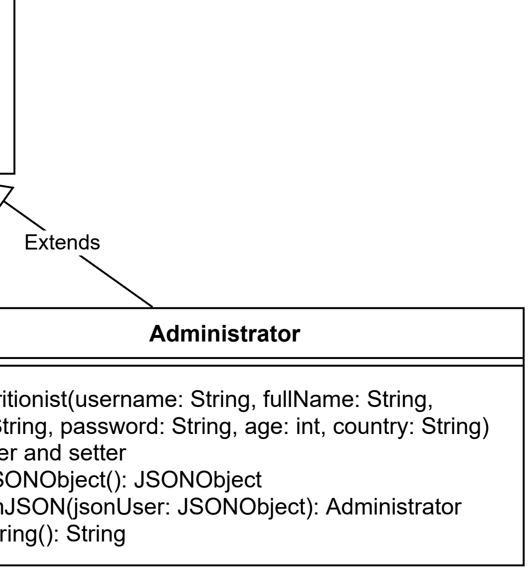

User
Food
0,N
0,N
Standard User
Administrator
Nutritionist
0,N
0,N
1
1,N
Nutrient
1,N
0,N
Diet
0,N



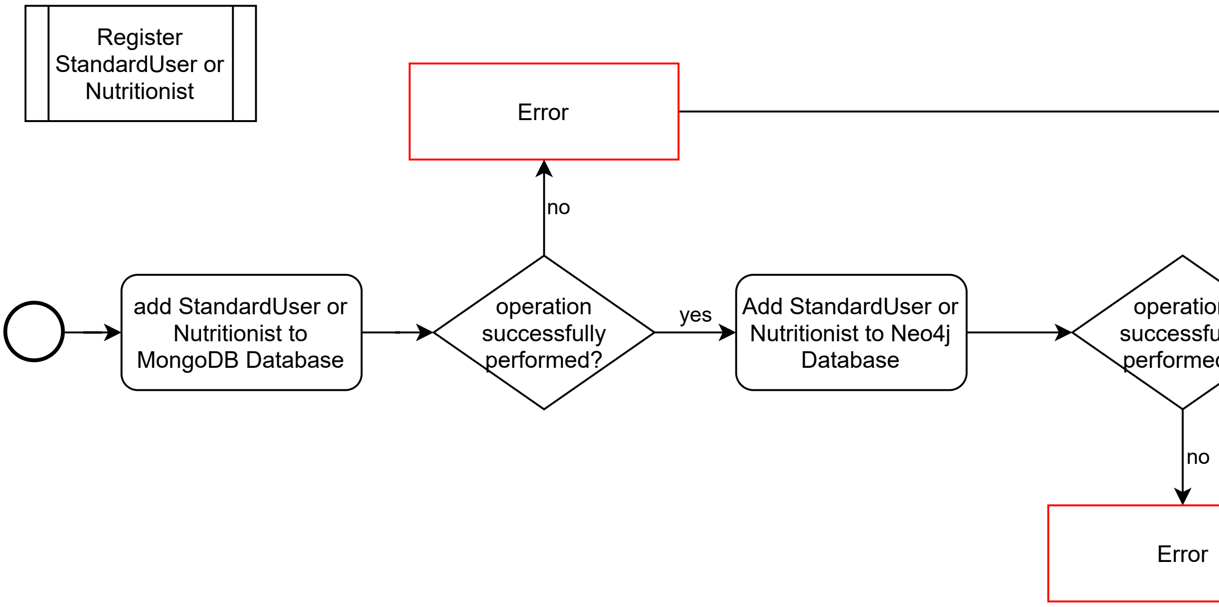
- eatenFoods: List<EatenFood>
- currentDiet: Diet

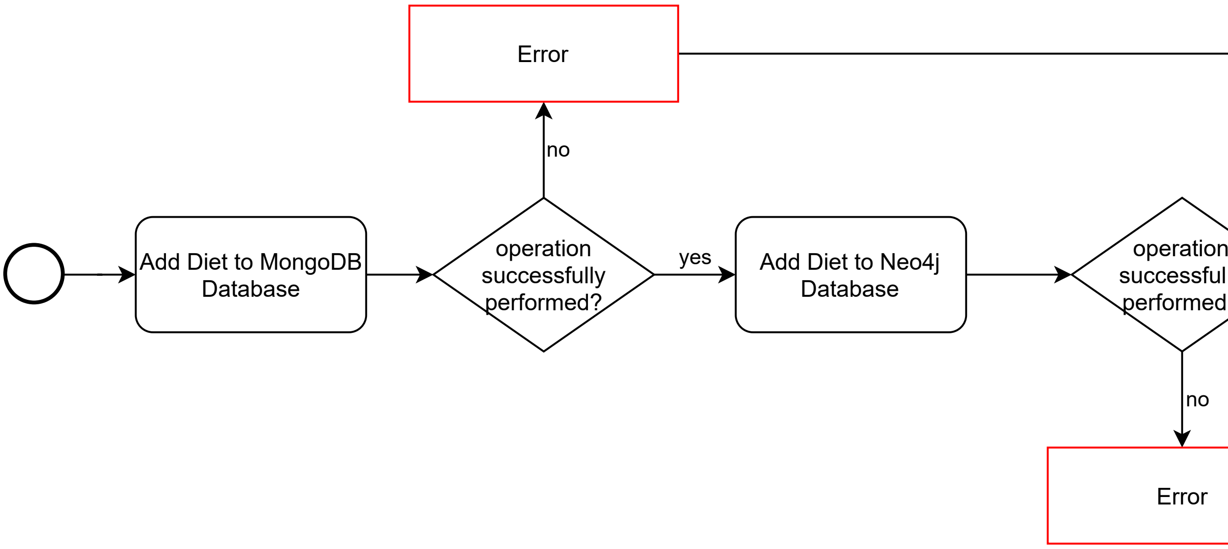
+ StandardUser(standardUser: StandardUSer)
+ StandardUser(username: String, fullName: Str
+ StandardUser(username: String, fullName: Str
+ StandardUser(username: String, fullName: Str
+ getter
+ setter
+ toJSONObject(): JSONObject
- computeNewEatenFoodID(): String
+ addEatenFood(foddName:String, quantity:int):
+ fromJSONObject(jsonUSer: JSONObject): Sta

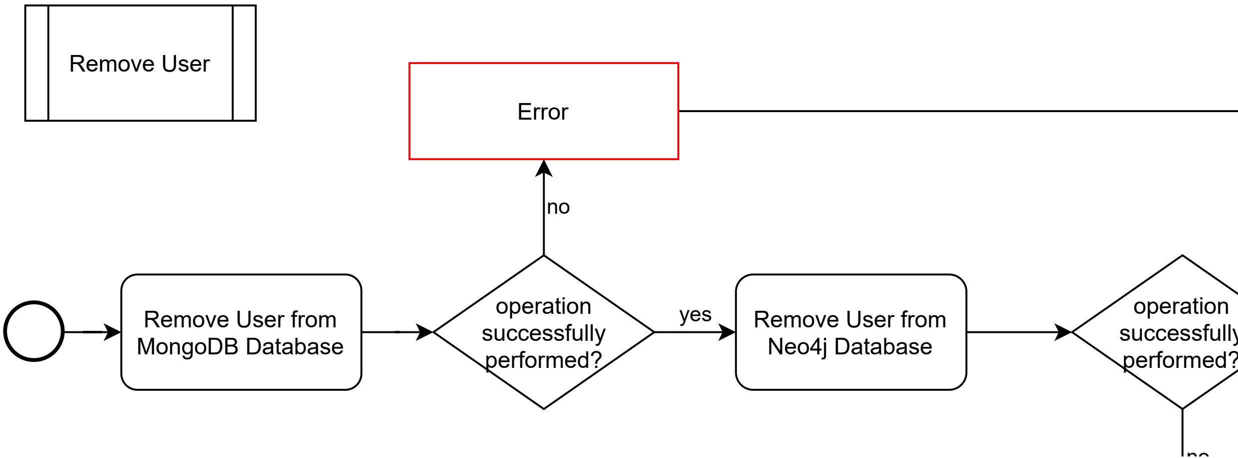

sex: String, password: String, age: int, country: String)
+ toJSONObject(): JSONObject
+ fromJSONObject(jsonUser: JSONObject): Nutritionist
+ toString(): String
Extends
+ User(username: String, fullName: String,
password: String, sex: String, age: int, country: String)
+ getter
+ setter
+ toJSONObject(): JSONObject
+ toString(): String
Extends
StandardUser
ring, sex: String, password: String, age: int, country: String)
ring, sex: String, password: String, age: int, country: String, eatenFoods: List<EatenFood>)
ring, sex: String, password: String, age: int, country: String, eatenFoods: List<EatenFood>, currentDiet: Diet)
: void
andardUser
+ Nutr
sex: S
+ gett
+ toJS
+ from
+ toSt

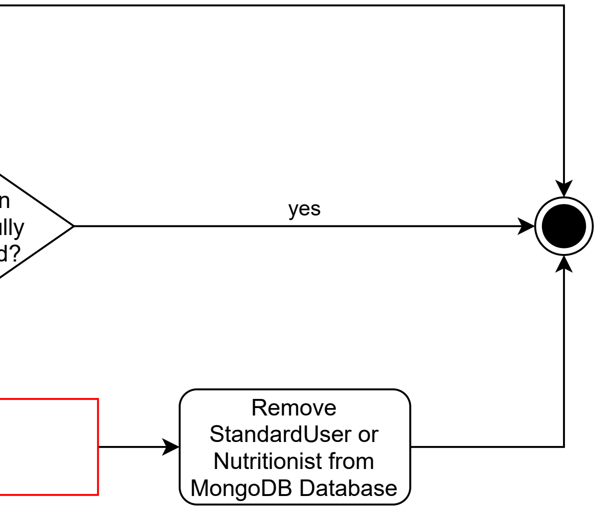

Extends
Administrator
ritionist(username: String, fullName: String,
String, password: String, age: int, country: String)
er and setter
SONObject(): JSONObject
nJSON(jsonUser: JSONObject): Administrator
ring(): String

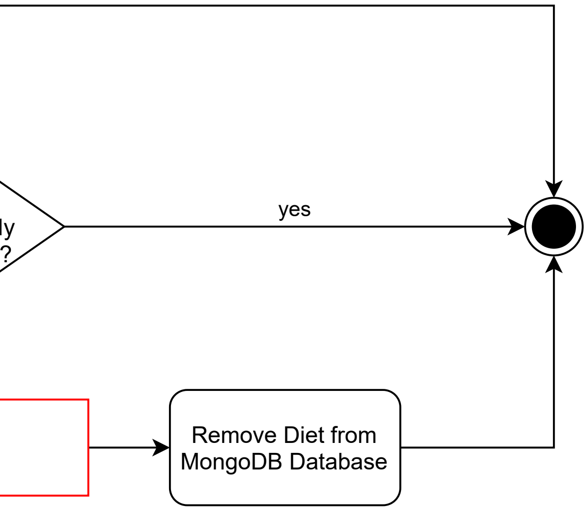
0,N

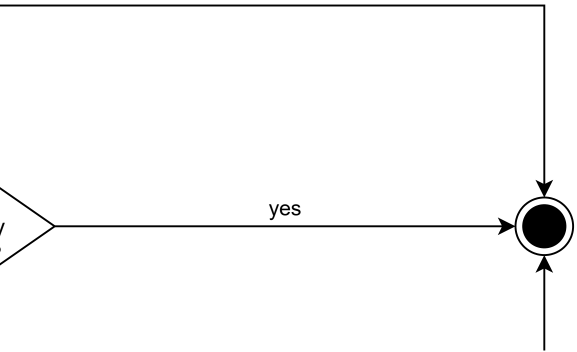

Register StandardUser or Nutritionist
Error
no
add StandardUser or Nutritionist to MongoDB Database
operation successfully performed?
yes
Add StandardUser or Nutritionist to Neo4j Database
operation successfu performed
no
Error

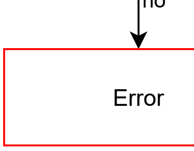

Error
no
Add Diet to MongoDB Database
operation successfully performed?
yes
Add Diet to Neo4j Database
operation successful performed
no
Error

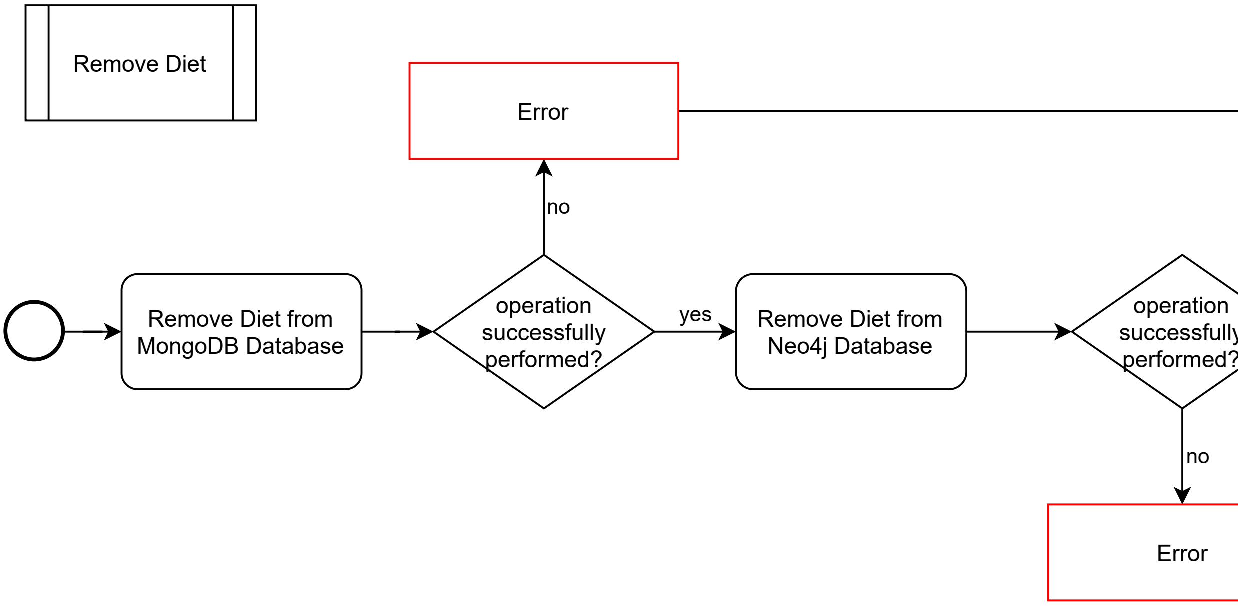

Remove User
Error
no
Remove User from MongoDB Database
operation successfully performed?
yes
Remove User from Neo4j Database
operation successfully performed?
no

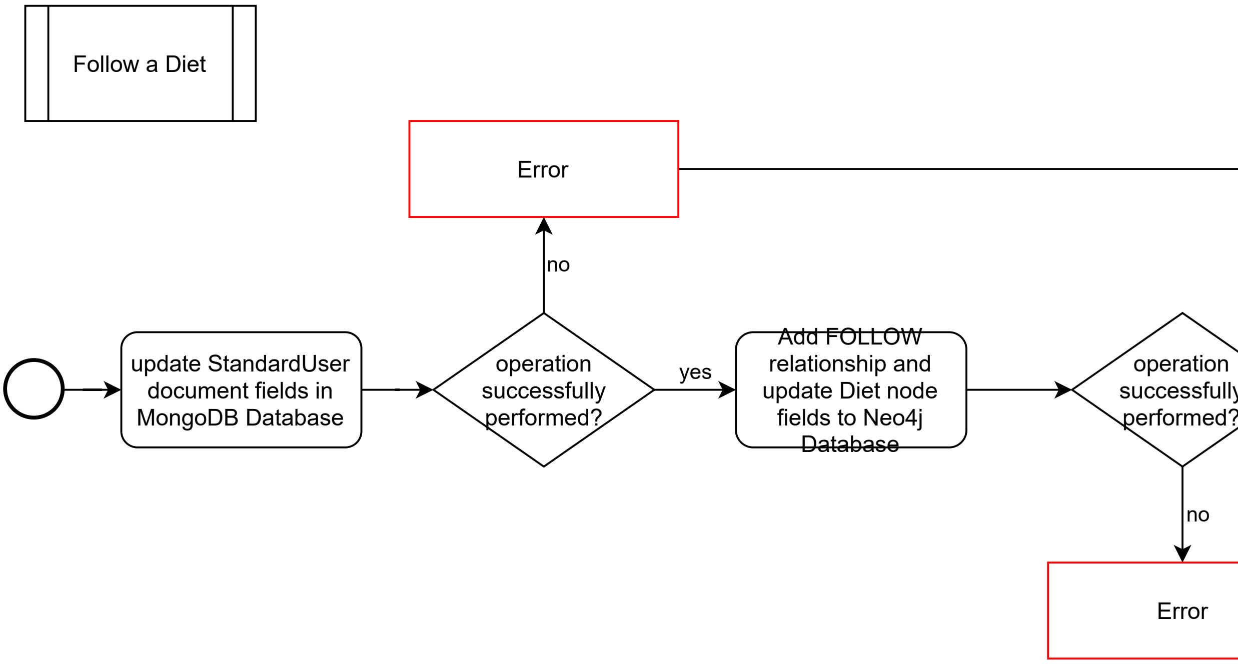

n
lly
d?
yes
Remove StandardUser or Nutritionist from MongoDB Database

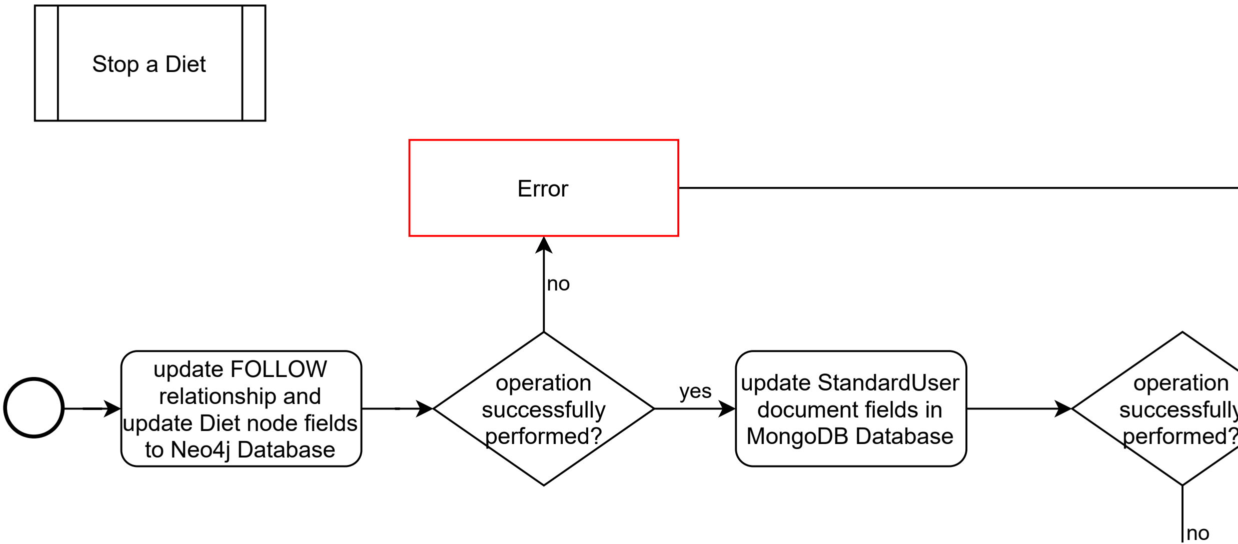

ly
?
yes
Remove Diet from MongoDB Database

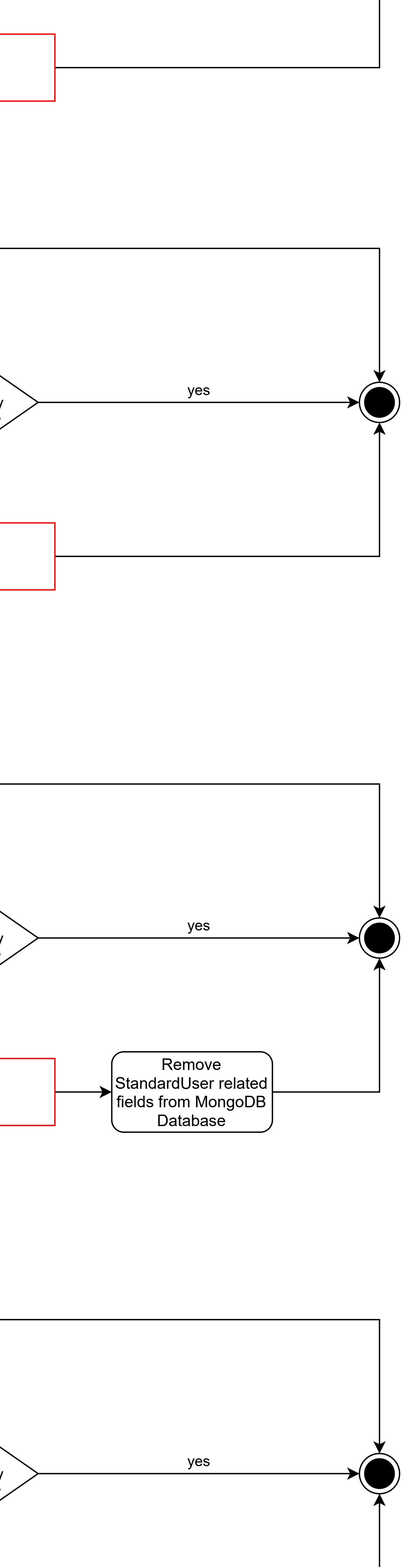

yes

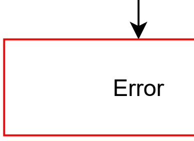

no
Error

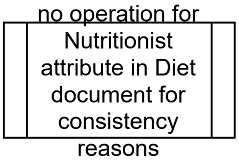

Remove Diet
Error
no
Remove Diet from MongoDB Database
operation successfully performed?
yes
Remove Diet from Neo4j Database
operation successfully performed?
no
Error


Follow a Diet
Error
no
update StandardUser document fields in MongoDB Database
operation successfully performed?
yes
Add FOLLOW relationship and update Diet node fields to Neo4j Database
operation successfully performed?
no
Error


Stop a Diet
Error
no
update FOLLOW relationship and update Diet node fields to Neo4j Database
operation successfully performed?
yes
update StandardUser document fields in MongoDB Database
operation successfully performed?
no

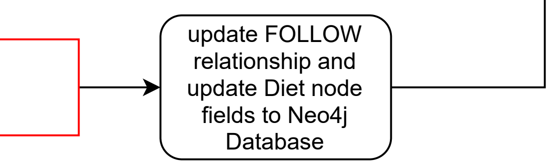

yes
yes
Remove StandardUser related fields from MongoDB Database
yes



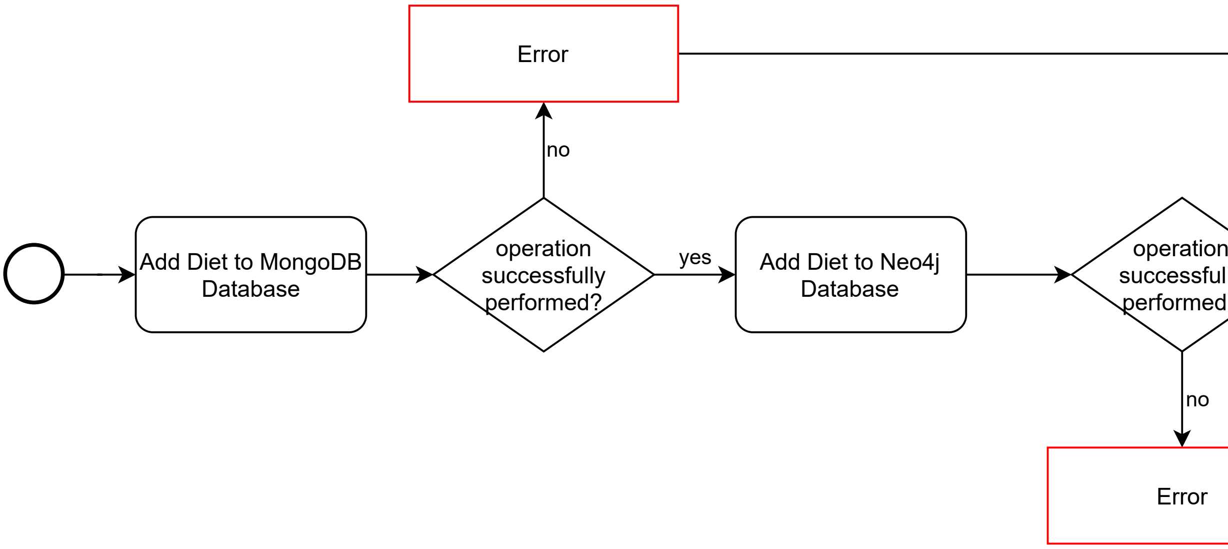

no operation for Nutritionist attribute in Diet document for consistency reasons

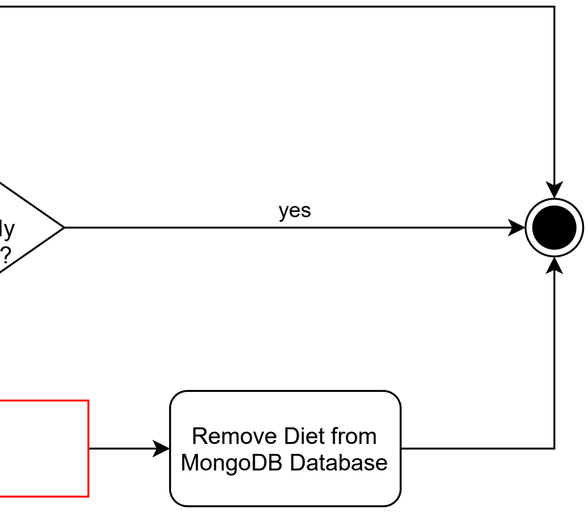
Stop a diet differences:

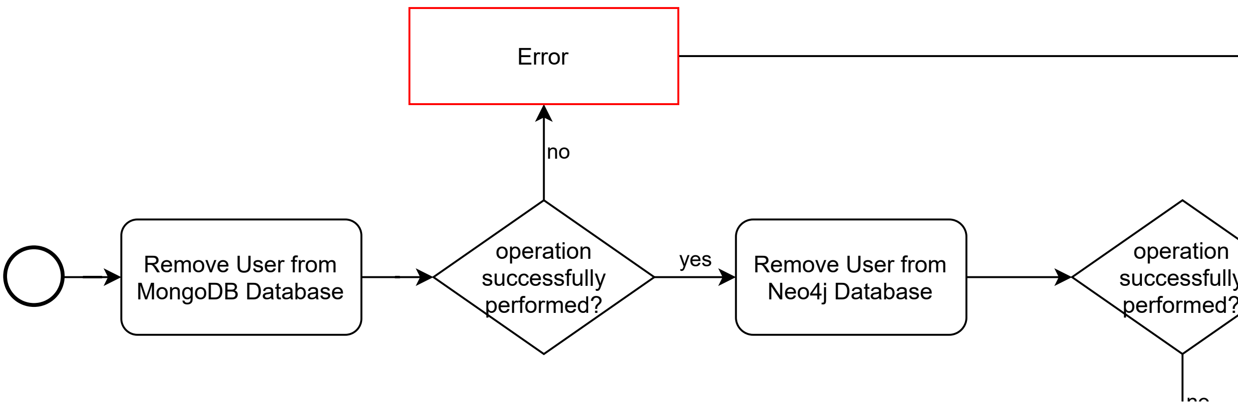
Instead of first update MongoDB and then Neo4j as for all the other consistency query processes, we perform a first update to Neo4j and then to MongoDB because, otherwise would require a first read of the StandardUser document to store the "currentDiet" and "currentDietStartDate" information and rollback them in case of error updating Neo4j

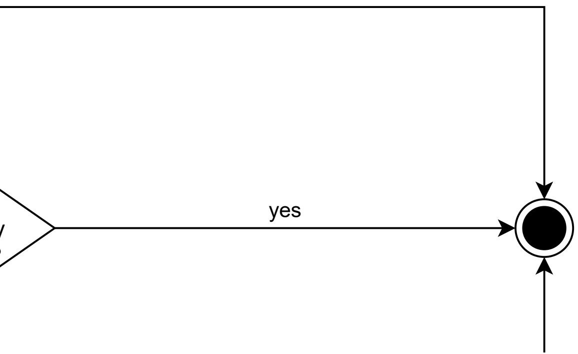

update FOLLOW relationship and update Diet node fields to Neo4j Database

e, it

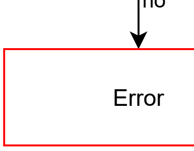

Error
no
Add Diet to MongoDB Database
operation successfully performed?
yes
Add Diet to Neo4j Database
operation successful performed
no
Error

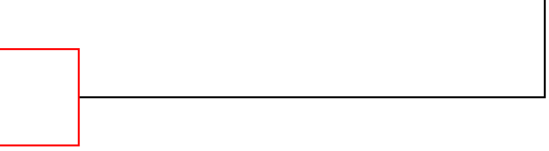

ly
?
yes
Remove Diet from MongoDB Database

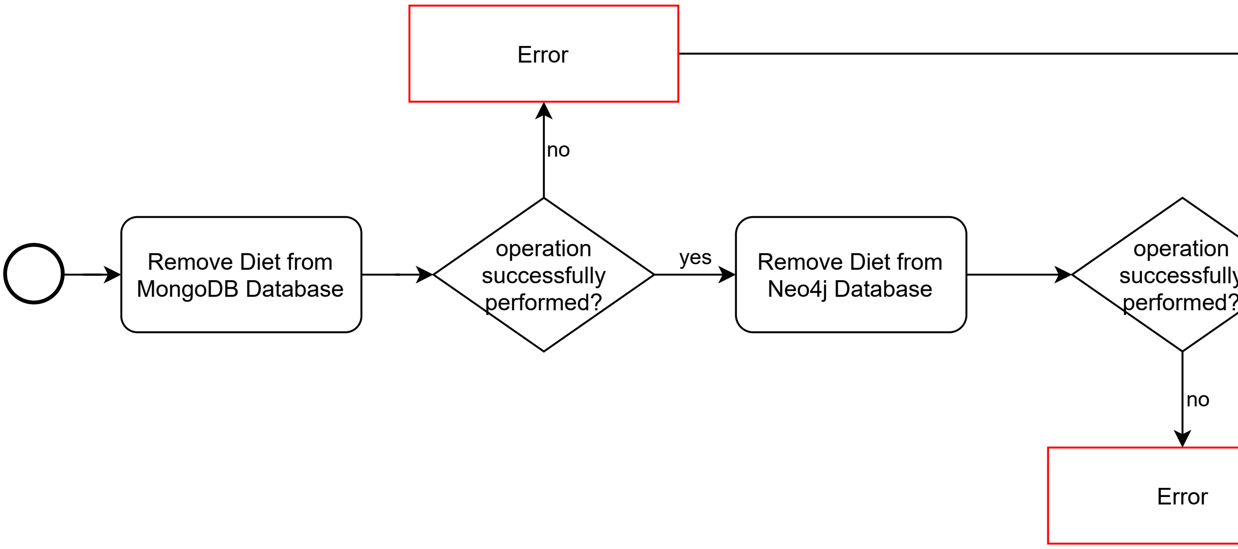

Error
no
Remove User from MongoDB Database
operation successfully performed?
yes
Remove User from Neo4j Database
operation successfully performed?
no

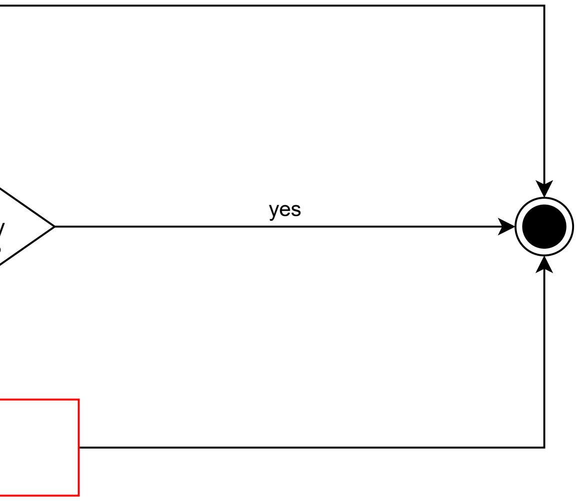


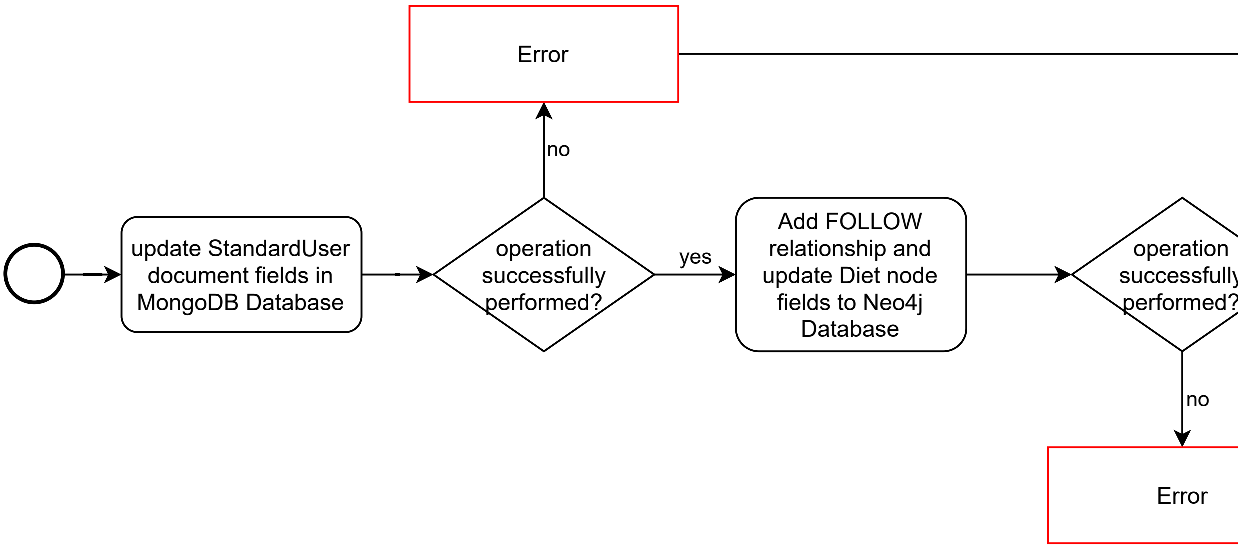
yes

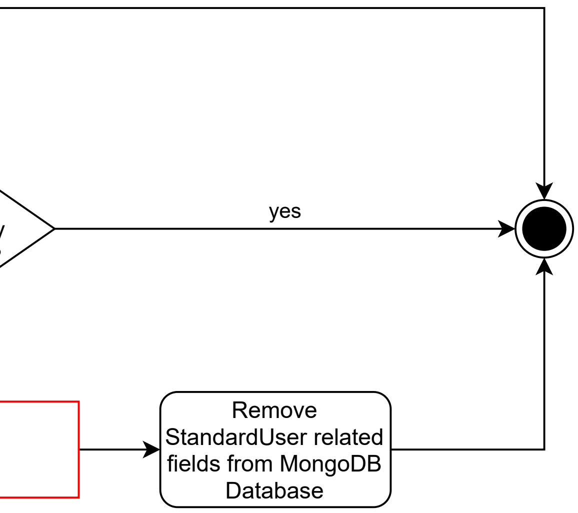

no
Error

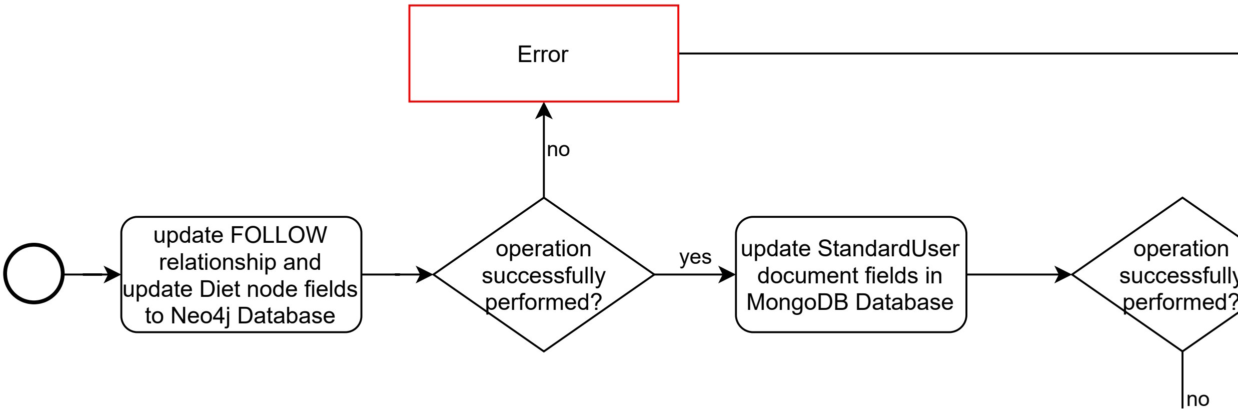


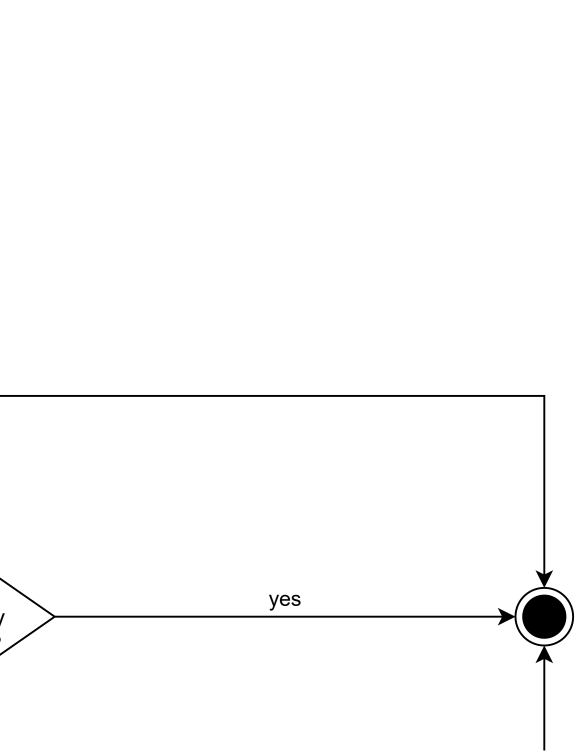

Error
no
Remove Diet from MongoDB Database
operation successfully performed?
yes
Remove Diet from Neo4j Database
operation successfully performed?
no
Error


yes

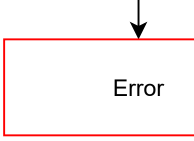

Error
no
update StandardUser document fields in MongoDB Database
operation successfully performed?
yes
Add FOLLOW relationship and update Diet node fields to Neo4j Database
operation successfully performed?
no
Error

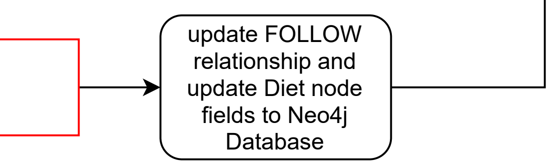

yes
Remove StandardUser related fields from MongoDB Database

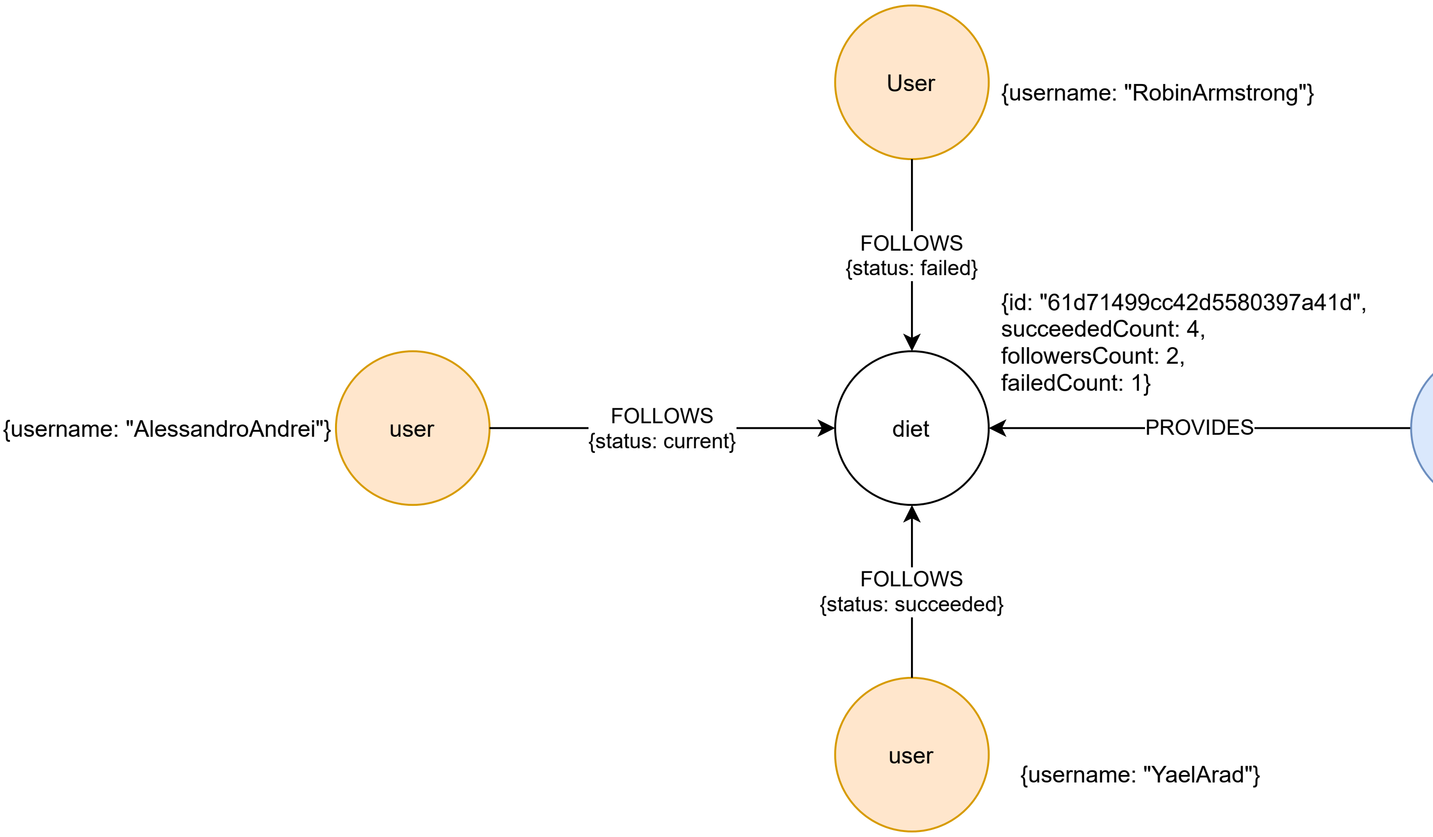

Error
no
update FOLLOW relationship and update Diet node fields to Neo4j Database
operation successfully performed?
yes
update StandardUser document fields in MongoDB Database
operation successfully performed?
no



yes


Error


update FOLLOW relationship and update Diet node fields to Neo4j Database


User
{username: "RobinArmstrong"}
FOLLOWS
{status: failed}
{id: "61d71499cc42d5580397a41d",
succeededCount: 4,
followersCount: 2,
failedCount: 1}
{username: "AlessandroAndrei"}
user
FOLLOWS
{status: current}
diet
PROVIDES
FOLLOWS
{status: succeeded}
user
{username: "YaelArad"}

nutritionist