Group 15 UML Diagram Final
Avijot Girn
Nour Ajami
Jesse Gerbrandt
Mohamed Amara

<<enumeration>>
FoodType

PRO
GRA
FV
OTHER
CAL

Client

-ClientType:String
-ClientID:int
-recommendedProtein:int
-recommendedGrain:int
-recommendedOther:int
-recommendedFV:int
-recommendedCalories:int

+Client(clientType:String, int grain, FV:int, protein:int, other:int, calories:int)
+getClientID(): int
+getClientType():String
+getProtein(): int
+getGrain(): int
+getOther(): int
+getFV(): int
+getCalories(): int
+printClientDetails():String

FoodAttributes

-ID:int
-NAME:String
-GRAIN_CONTENT:int
-FV_CONTENT:int
-PROTEIN_CONTENT:int
-OTHER_CONTENT:int
-FOOD_CALORIES:int
-baseNumberInInventory:int
numberInInventory:int

+FoodAttributes(ID:int, name:String, grain_content:int, fv_content:int, protein_content:int, other_content:int, food_calories:int)
+setBase(): void
+ResetBase(): void
+IncrementAmount(): void
+DecrementAmount(): void
+getAmount(): int
+getGrainContent():int
+getProteinContent():int
+getFVContent():int
+getOtherContent():int
+getCalorieContent():int
+getFoodID():int
+GetType(foodType: FoodType):int
+GetSumExcept(type: FoodType):int
+equals(foodToCompare: Object): boolean
+ToString():String

InverseProComparator

+ compare(a: FoodAttributes, b: FoodAttributes:int
+SumInverseCalories(a: FoodAttributes): int

InverseOtherComparator

+ compare(a: FoodAttributes, b: FoodAttributes:int
+SumInverseCalories(a: FoodAttributes): int

InverseGrainComparator

+ compare(a: FoodAttributes, b: FoodAttributes:int
+SumInverseCalories(a: FoodAttributes): int

InverseFVComparator

+ compare(a: FoodAttributes, b: FoodAttributes:int
+SumInverseCalories(a: FoodAttributes): int

InverseCalComparator

+ compare(a: FoodAttributes, b: FoodAttributes:int
+SumInverseCalories(a: FoodAttributes): int

SumsOfNutrients

-int sumPro = 0
-int sumGrain = 0
-int sumFV = 0
-int sumOther = 0
int sumCal = 0

+UpdateSums(food: FoodAttributes): void
+GetPro(): int
+GetGrain(): int
+GetOther(): int
+GetFV(): int
+GetCal(): int
+ToString(): String
+GetType(type: FoodType): int
+ClearData(): void

ClientList

-MAX_CAPACITY: int(10)
-numOfAdMales: Integer(0)
-numOfAdFemales: Integer(0)
-numOfChOv8: Integer(0)
-numOfChUn8: Integer(0)
-listOfClients: ArrayList<Client>
-clientNeeds: ArrayList<Client>

+createClientToAdd(ClientID:int, numOfClientsToAdd:int)
+addToList(clientToAdd:Client)
+addToClientNeeds(clientToAdd:Client)
+get(index:int):Client
+getNeeds(index:int):Client
+getTotalRecommendedGrainForWeek():int
+getTotalRecommendedFVForWeek():int
+getTotalRecommendedProteinForWeek():int
+getTotalRecommendedOtherForWeek():int
+getTotalRecommendedCaloriesForWeek():int
+size():int
+numOf(id:int):String
+ClearClients()

SQLReader

-dbConnet: Connection
-results : ResultSet
foodInv: FoodInventory
possibleClients : ClientList
testForm : OrderForm
finder : NutritionalFinder
+DBURL: String
+USERNAME : String
+PASSWORD : String

+SQLReader(url: String, user: String, pw: String)
+initializeConnection(): Connection
+delete(Name: String): void
+selectAllFoods(tableName: String): String
+selectClients(): void
+close(): void

ClassNotFoundException

IllegalArgumentException

Hamper

-hamperClients:ClientList
-hamperFood:ArrayList<FoodAttributes>

+Hamper(clients: ClientList)
+fillHamper(food: FoodAttributes)
+getClientTypes():String
+getHamperFood():String

OrderForm

-listOfHampers : ArrayList<Hamper>
-numberOfhampers : int
-increment : int

+addHamper(index: int) : void
+getHamper(index : int) : Hamper
+ public writeToOutput() : void

FoodInventory

-generalInventory():ArrayList<FoodAttributes>
-proteinSortedInventory():ArrayList<FoodAttributes>
-grainSortedInventory():ArrayList<FoodAttributes>
-FVSortedInventory():ArrayList<FoodAttributes>
-otherSortedInventory():ArrayList<FoodAttributes>
-calorieSortedInventory():ArrayList<FoodAttributes>
totalCal:int(0)

+addingToTheList(foodAttributes: FoodAttributes)
+TotalCal():
+SetBase():
+ResetBase():
+RemoveFromList(food: FoodAttributes):
+sortByNutrition():
+resetInventories():
+ToString():String
+GetGeneralInventory():ArrayList<FoodAttributes>
+GetGrainSortedInventory():ArrayList<FoodAttributes>
+GetProteinInventory():ArrayList<FoodAttributes>
+GetFVSortedInventory():ArrayList<FoodAttributes>
+GetOtherSortedInventory():ArrayList<FoodAttributes>
+GetCalorieSortedInventory():ArrayList<FoodAttributes>

NutritionalFinder

-int sumPro = 0
-int sumGrain = 0
-int sumFV = 0
-int sumOther = 0
-int sumCal = 0
-foodInv:FoodInventory
+son:SumsOfNutrients
+toAdd : ArrayList<FoodAttributes>

+ClientNutFinder(client: ClinetList): void
-DeleteFromDb(foodToAdd: FoodAttributes): void
+GetFoodInToAdd(toAdd: ArrayList<FoodAttributes>): void
-DistrubuteFood(inv: ArrayList<FoodAttributes>, OverLimitItems: ArrayList<FoodAttributes>, toAdd: ArrayList<FoodAttributes>): void
-DoesGoOverLimitExcept(food: FoodAttributes, pro: int, grain: int, fv: int, other: int, cal: int, type: FoodType) boolean
-findClosest(arr: ArrayList<FoodAttributes>, target: int, foodType: FoodType): FoodAttributes

---

GUIHamper

-int mID = 1
-int fID = 2
-int o8ID = 3
-int u8ID = 4
-JLabel instructions
-JLabel amLabel
-JLabel afLabel
-JLabel o8Label
-JLabel u8Label
-JLabel numFamLabel

-JTextField amInput
-JTextField afInput
-JTextField o8Input
-JTextField u8Input

+setupGUI()
+actionPerformed(event: ActiuonEvent)
+mouseClicked(event: MouseEvent)
+validateInput(): boolean
+createFamily(numAdultM: int, numAdultF: int, numOver8: int, numUnder8: int)

IOException

<<throws>>