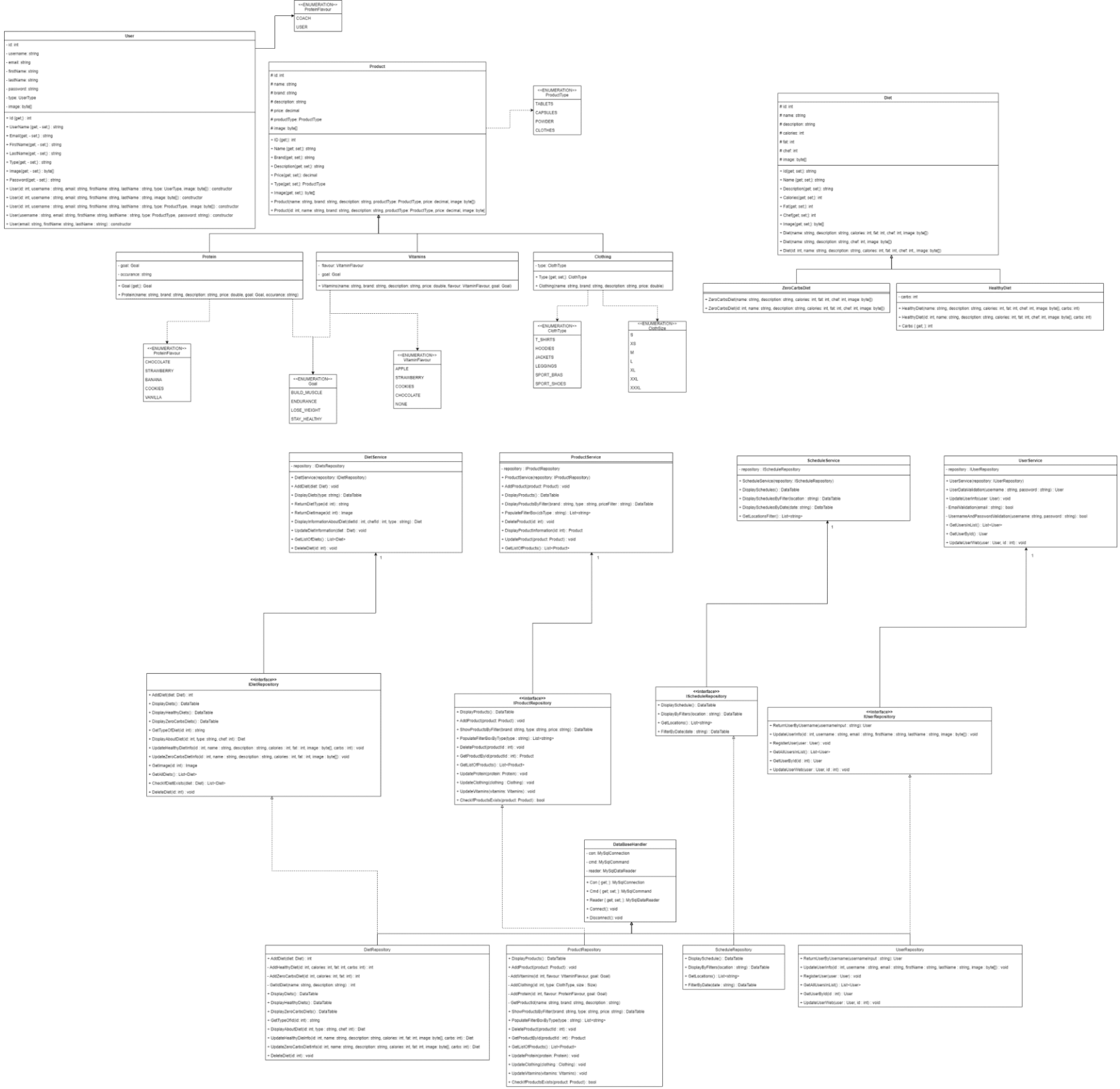
User
- id: int
- username: string
- email: string
- firstName: string
- lastName: string
- password: string
- type: UserType
- image: byte[]
+ Id (get,) : int
+ UserName (get, - set,) : string
+ Email(get, - set,) : string
+ FirstName(get, - set) : string
+ LastName(get, - set) : string
+ Type(get, - set) : string
+ Image(get, - set) : byte[]
+ Password(get, - set) : string
+ User(id: int, username : string, email: string, firstName: string, lastName : string, type: UserType, image: byte[]) : constructor
+ User(id: int, username : string, email: string, firstName: string, lastName : string, image: byte[]) : constructor
+ User(id: int, username : string, email: string, firstName: string, lastName : string, type: ProductType, image: byte[]) : constructor
+ User(username : string, email: string, firstName: string, lastName : string, type: ProductType, password: string) : constructor
+ User(email: string, firstName: string, lastName : string) : constructor
<<ENUMERATION>>
Protein Flavour
COACH
USER
Product
id: int
name: string
brand: string
description: string
price: decimal
productType: ProductType
image: byte[]
+ ID (get,): int
+ Name (get; set;): string
+ Brand(get; set;): string
+ Description(get; set;): string
+ Price(get; set;): decimal
+ Type(get; set;): ProductType
+ Image(get; set;): byte[]
+ Product(name: string, brand: string, description: string, productType: ProductType, price: decimal, image: byte[])
+ Product(id: int, name: string, brand: string, description: string, productType: ProductType, price: decimal, image: byte[]
<<ENUMERATION>>
ProductType
TABLETS
CAPSULES
POWDER
CLOTHES
Diet
id: int
name: string
description: string
calories: int
fat: int
chef: int
image: byte[]
+ Id(get; set;): string
+ Name (get; set;): string
+ Description(get; set;): string
+ Calories(get; set;): int
+ Fat(get; set;): int
+ Chef(get; set;): int
+ Image(get; set;): byte[]
+ Diet(name: string, description: string, calories: int, fat: int, chef: int, image: byte[])
+ Diet(name: string, description: string, chef: int, image: byte[])
+ Diet(id: int, name: string, description: string, calories: int, fat: int, chef: int, image: byte[])
Protein
- goal: Goal
- occurance: string
+ Goal (get,): Goal
+ Protein(name: string, brand: string, description: string, price: double, goal: Goal, occurance: string)
Vitamins
- flavour: VitaminFlavour
- goal: Goal
+ Vitamins(name: string, brand: string, description: string, price: double, flavour: VitaminFlavour, goal: Goal)
Clothing
- type: ClothType
+ Type (get; set;): ClothType
+ Clothing(name: string, brand: string, description: string, price: double)
ZeroCarbsDiet
+ ZeroCarbsDiet(name: string, description: string, calories: int, fat: int, chef: int, image: byte[])
+ ZeroCarbsDiet(id: int, name: string, description: string, calories: int, fat: int, chef: int, image: byte[])
HealthyDiet
- carbs: int
+ HealthyDiet(name: string, description: string, calories: int, fat: int, chef: int, image: byte[], carbs: int)
+ HealthyDiet(id: int, name: string, description: string, calories: int, fat: int, chef: int, image: byte[], carbs: int)
+ Carbs (get,): int
<<ENUMERATION>>
ProteinFlavour
CHOCOLATE
STRAWBERRY
BANANA
COOKIES
VANILLA
<<ENUMERATION>>
Goal
BUILD_MUSCLE
ENDURANCE
LOSE_WEIGHT
STAY_HEALTHY
<<ENUMERATION>>
VitaminFlavour
APPLE
STRAWBERRY
COOKIES
CHOCOLATE
NONE
<<ENUMERATION>>
ClothType
T_SHIRTS
HOODIES
JACKETS
LEGGINGS
SPORT_BRAS
SPORT_SHOES
<<ENUMERATION>>
ClothSize
S
XS
M
L
XL
XXL
XXXL
DietService
- repository : IDietsRepository
+ DietService(repository: IDietRepository)
+ AddDiet(diet: Diet) : void
+ DisplayDiets(type: string) : DataTable
+ ReturnDietType(id: int) : string
+ ReturnDietImage(id: int) : Image
+ DisplayInformationAboutDiet(dietId : int, chefId : int, type : string) : Diet
+ UpdateDietInformation(diet : Diet) : void
+ GetListOfDiets() : List<Diet>
+ DeleteDiet(id: int) : void
1
ProductService
- repository : IProductRepository
+ ProductService(repository: IProductRepository)
+ AddProduct(product: Product): void
+ DisplayProducts() : DataTable
+ DisplayProductsByFilter(brand : string, type : string, priceFilter : string) : DataTable
+ PopulateFilterBoxByType(type : string) : List<string>
+ DeleteProduct(id: int) : void
+ DisplayProductInformation(id: int) : Product
+ UpdateProduct(product: Product) : void
+ GetListOfProducts() : List<Product>
1
ScheduleService
- repository : IScheduleRepository
+ ScheduleService(repository: IScheduleRepository)
+ DisplaySchedules() : DataTable
+ DisplaySchedulesByFilter(location : string) : DataTable
+ DisplaySchedulesByDate(date: string) : DataTable
+ GetLocationsFilter() : List<string>
1
UserService
- repository : IUserRepository
+ UserService(repository: IUserRepository)
+ UserDataValidation(username : string, password : string) : User
+ UpdateUserInfo(user: User) : void
- EmailValidation(email : string) : bool
- UsernameAndPasswordValidation(username: string, password : string) : bool
+ GetUsersInList() : List<User>
+ GetUserById() : User
+ UpdateUserWeb(user: User, id : int) : void
1
<<interface>>
IDietRepository
+ AddDiet(diet: Diet) : int
+ DisplayDiets() : DataTable
+ DisplayHealthyDiets() : DataTable
+ DisplayZeroCarbsDiets() : DataTable
+ GetTypeOfDiet(id: int) : string
+ DisplayAboutDiet(id: int, type: string, chef: int) : Diet
+ UpdateHealthyDietInfo(id : int, name : string, description : string, calories : int, fat : int, image : byte[], carbs : int) : void
+ UpdateZeroCarbsDietInfo(id : int, name : string, description : string, calories : int, fat : int, image : byte[]) : void
+ GetImage(id: int) : Image
+ GetAllDiets() : List<Diet>
+ CheckIfDietExists(diet : Diet) : List<Diet>
+ DeleteDiet(id: int) : void
<<interface>>
IProductRepository
+ DisplayProducts() : DataTable
+ AddProduct(product: Product) : void
+ ShowProductsByFilter(brand: string, type: string, price: string) : DataTable
+ PopulateFilterBoxByType(type : string) : List<string>
+ DeleteProduct(productId : int) : void
+ GetProductById(productId : int) : Product
+ GetListOfProducts() : List<Product>
+ UpdateProtein(protein: Protein) : void
+ UpdateClothing(clothing : Clothing) : void
+ UpdateVitamins(vitamins: Vitamins) : void
+ CheckIfProductsExists(product: Product) : bool
<<interface>>
IScheduleRepository
+ DisplaySchedule() : DataTable
+ DisplayByFilters(location : string) : DataTable
+ GetLocations() : List<string>
+ FilterByDate(date : string) : DataTable
<<interface>>
IUserRepository
+ ReturnUserByUsername(usernameInput : string): User
+ UpdateUserInfo(id : int, username : string, email : string, firstName : string, lastName : string, image : byte[]) : void
+ RegisterUser(user : User) : void
+ GetAllUsersInList() : List<User>
+ GetUserById(id: int) : User
+ UpdateUserWeb(user : User, id : int) : void
DataBaseHandler
- con: MySqlConnection
- cmd: MySqlCommand
- reader: MySqlDataReader
+ Con (get,) : MySqlConnection
+ Cmd (get; set,) : MySqlCommand
+ Reader (get; set;): MySqlDataReader
+ Connect(): void
+ Disconnect(): void
DietRepository
+ AddDiet(diet: Diet) : int
+ AddHealthyDiet(id: int, calories: int, fat: int, carbs: int) : int
+ AddZeroCarbsDiet(id: int, calories: int, fat: int) : int
- GetIdDiet(name: string, description: string) : int
+ DisplayDiets() : DataTable
+ DisplayHealthyDiets() : DataTable
+ DisplayZeroCarbsDiets() : DataTable
+ GetTypeOfDiet(id: int) : string
+ DisplayAboutDiet(id: int, type : string, chef: int) : Diet
+ UpdateHealthyDietInfo(id: int, name: string, description: string, calories: int, fat: int, image: byte[], carbs: int) : Diet
+ UpdateZeroCarbsDietInfo(id: int, name: string, description: string, calories: int, fat: int, image: byte[], carbs: int) : Diet
+ DeleteDiet(id: int) : void
ProductRepository
+ DisplayProducts() : DataTable
+ AddProduct(product: Product) : void
+ AddVitamins(id: int, flavour: VitaminFlavour, goal: Goal)
+ AddClothing(id: int, type: ClothType, size : Size)
+ AddProtein(id: int, flavour: ProteinFlavour, goal: Goal)
- GetProductId(name: string, brand: string, description : string)
+ ShowProductsByFilter(brand: string, type: string, price: string) : DataTable
+ PopulateFilterBoxByType(type : string) : List<string>
+ DeleteProduct(productId : int) : void
+ GetProductById(productId : int) : Product
+ GetListOfProducts() : List<Product>
+ UpdateProtein(protein: Protein) : void
+ UpdateClothing(clothing : Clothing) : void
+ UpdateVitamins(vitamins: Vitamins) : void
+ CheckIfProductsExists(product: Product) : bool
ScheduleRepository
+ DisplaySchedule() : DataTable
+ DisplayByFilters(location : string) : DataTable
+ GetLocations() : List<string>
+ FilterByDate(date : string) : DataTable
UserRepository
+ ReturnUserByUsername(usernameInput : string): User
+ UpdateUserInfo(id : int, username : string, email : string, firstName : string, lastName : string, image : byte[]) : void
+ RegisterUser(user : User) : void
+ GetAllUsersInList() : List<User>
+ GetUserById(id: int) : User
+ UpdateUserWeb(user : User, id : int) : void