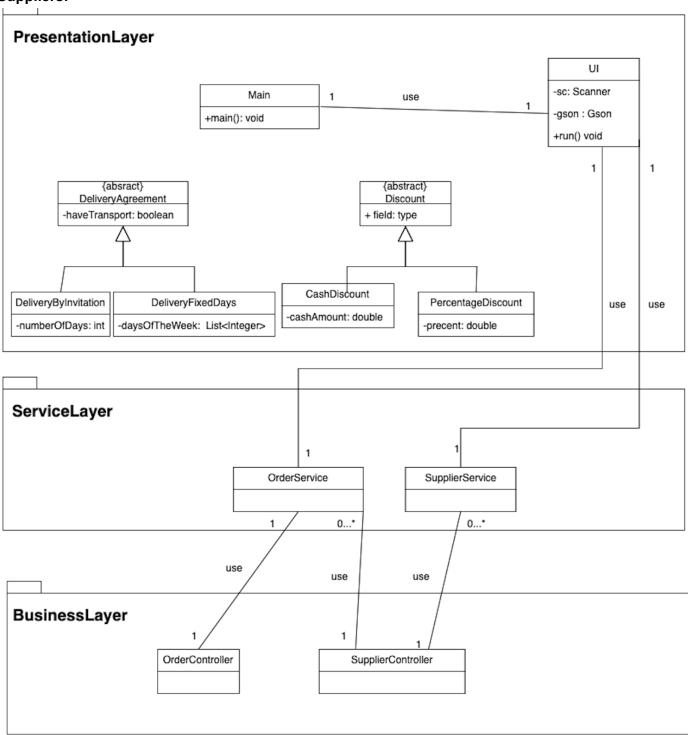
# Suppliers:



## BusinessLayer OrderController +order(Map<Integer, Integer> order): Map<String, Pair<Map<Integer, Integer>, Double>> SupplierController +addSupplier(//all the fields pf supplier (theres a lot))): void +removeSupplier(String bnNumber): void +getSupplier(String bnNumber): void //manages suppliers and their data BankAccount Agreement Supplier -bank: string -paymentMethod: string -name: string --branch: string -products: Map<Integer, product> -bnNumber: string -accountNumber:string +addProduct(Product p): void -paymentMethod: string +removeProduct(int productId):void -fields: List<string> +getProducts(): List<Product> -contactInfo: Map<string, Pair> +getBillOfQuantities(): BillOfQuantities //setters and getters Order Product -products: Map<int, int> {absract} -name: string DeliveryAgreement +addProduct(int productId, int amount): void -haveTransport: boolean -id: int //setters and getters -catalogNumber: string -price: double -numberOfUnits: int //setters and getters DeliveryByInvitation DeliveryFixedDays -numberOfDays: int -daysOfTheWeek: List<Integer> //setters and getters //setters and getters BillOfQuantities productsDiscount: Map<int, Map<int, Discount> -discountOnAmountOfProducts: Function<Double, Double> -discountOnTotalOrder: Function<Double, Double> -amountToReachForDiscount: int -priceToReachForDiscount: double PercentageDiscount -amountBeforeTotal: boolean +applyDiscount(double price): double +addProductDiscount(int productId, int amount, Discount discount): void +removeProductDiscount(int productId, int amount): void

CashDiscount

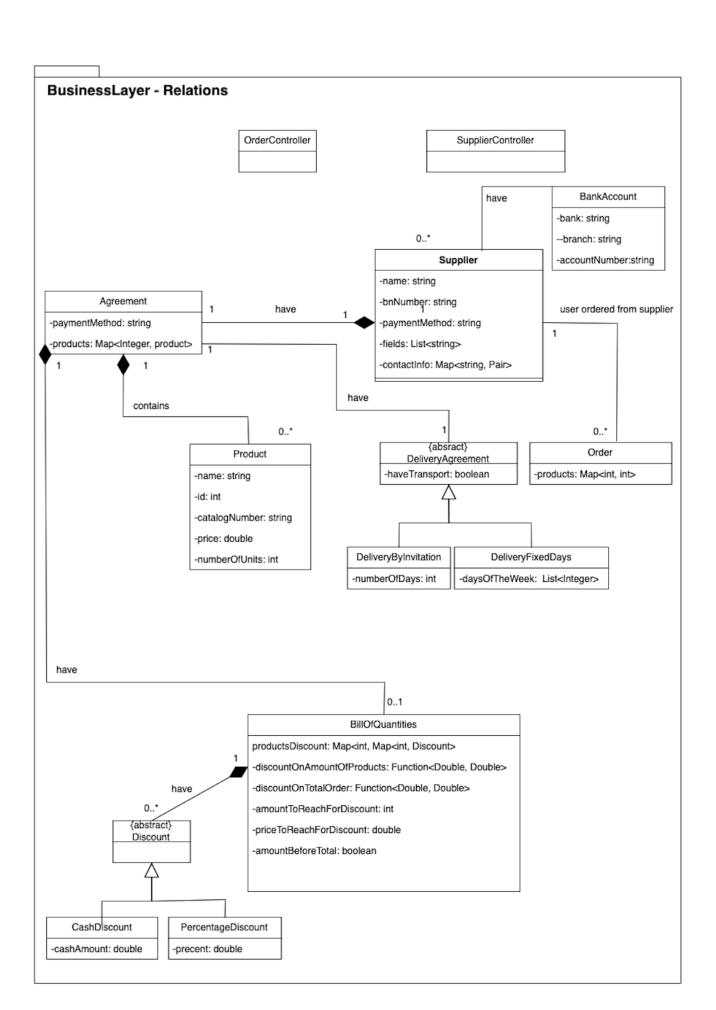
-cashAmount: double

PercentageDiscount

precent: double

+getProductPriceAfterDiscount(int productId, int amount, double price): double

+getPriceAfterDiscounts(int amount, double priceBefore): double



#### **Data Access Layer**

{abstract}
AbstractDataMapper
# tableName: String
# columns: String[]

#### BillOfQuantitiesDataMapper

+createBillOfQuantities(String BnNumber): BillOfQuantities

#### AgreementDataMapper

+createAgreement(String BnNumber): Agreement

#### OrderHistoryDataMapper

- +insert(String bnNumber, int orderld, String catalog\_number, int quantity): void
- +delete(String bnNumber, int orderld): void
- +find(String bnNumber, int orderId): Order
- +findAll(String bnNumber): List<Order>

#### DiscountOnAmountDataMapper

- +insert(String bnNumber, double amountToReach, Discount discount): void
- +delete(String bnNumber): void
- +updateAmountToReach(String bnNumber, double amountToReach): void
- +updateDiscount(String bnNumber, Discount discount): void
- +find(String bnNumber): Pair<Integer, Discount>

#### DiscountOnTotalOrderDataMapper

- $+ insert (String\ bn Number,\ double\ price To Reach,\ Discount\ discount) \colon\ void$
- +delete(String bnNumber): void
- +updatePriceToReach(String bnNumber, double priceToReach): void
- +updateDiscount(String bnNumber, Discount discount): void
- +find(String bnNumber): Pair<Integer, Discount>

#### DeliveryAgreementDataMapper

- + insert(String bnNumber, boolean haveTransport, boolean byInvitaion, String days): void
- + updateBnNumber(String bnNumber, String newBnNumber): void
- + updateHaveTransport(String bnNumber, boolean haveTransport): void
- + updateByInvitation(String bnNumber, boolean byInvitation): void
- + updateDays(String bnNumber, String days): void
- + find(String bnNumber): DeliveryAgreement
- + delete(String bnNumber): void

#### ContactsInfoDataMapper

- +insert(String Name, String email, String phoneNumber): void
- +delete(String name, String email, String phoneNumber): void
- +updateEmail(String name, String email, String phoneNumber, String newEmail): void
- +updateName(String name, String email, String phoneNumber, String newName): void
- +updatePhoneNumber(String name, String email, String phoneNumber, String newPhoneNumber): void
- +find(String name, String email, String phoneNumber): Pair<String, Pair<String, String>>

#### **Data Access Layer**

{abstract}
AbstractDataMapper
# tableName: String
# columns: String[]

AgreementDataMapper

+createAgreement(String BnNumber): Agreement

#### SuppliersProductsDataMapper

- +insert(String bnNumber, String catalogNumber, String SuppliersCatalogNumber, int quantity, double price): void
- +deleteAllProducts(String bnNumber): void
- +delete(String bnNumber, String catalogNumber): void
- +find(String bnNumber, String catalogNumber): Product
- +findAllProducts(String bnNumber): List<Product>
- +updateBnNumber(String bnNumber,String newBnNumber): void
- +updateSuppliersCatalogNumber(String bnNumber, String catalogNumber): void
- +updateQuantity(String bnNumber, String catalogNumber, int quantity): void
- +updatePrice(String bnNumber, String catalogNumber, double price): void

## SuppliersDataMapper

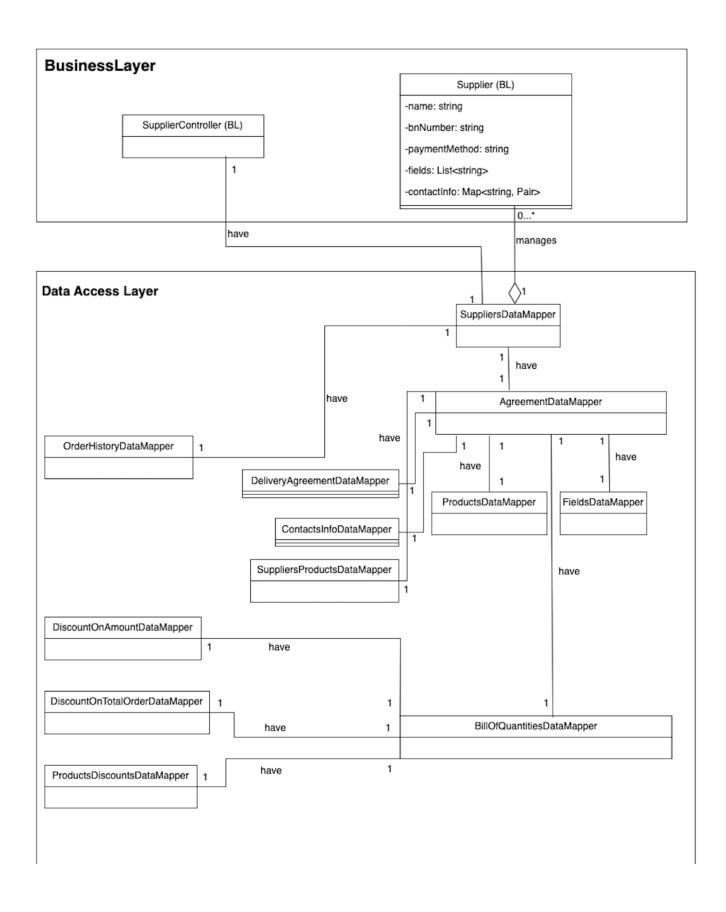
- +insert(String bnNumber, String name, String bank, String branch, String accountNumber, String payementMethod): void
- +delete(String bnNumber): void
- +find(String bnNumber): Supplier
- +updateBankAccount(String bnNumber,String bank, String branch, String accountNumber): void
- +updateName(String bnNumber, String name): void
- +updatePaymentMethod(String bnNumber, String PaymentMethod): void

### ProductsDiscountsDataMapper

- +insert(String bnNumber, String catalogNumber, int amount, double percentage, double cash): void
- +delete(String bnNumber, String catalogNumber): void
- +delete(String bnNumber, String catalogNumber, int amount): void
- +find(String bnNumber, String catalogNumber, int amount): Discount
- +findAllAmounts(String bnNumber, String catalogNumber): Map<Integer, Discount>
- +findAllProductsDiscounts(String bnNumber): Map<I<Integer, Map<Integer, Discount>>
- +updateDiscount(String bnNumber, String catalogNumber, int amount, double percentage, double cash): void

# ProductTypesDataMapper +insert(String catalogNumber, String name, String manufacture): void +delete(String catalogNumber): void +find(String catalogNumber): boolean +updateMaufacture(String catalogNumber, String manufacture): boolean +updateField(String bnNumber, String field): void +find(String bnNumber, String field, String new Field): void +find(String bnNumber, String field): String +findAll(String bnNumber): List<String>

# **DAL - Relations**

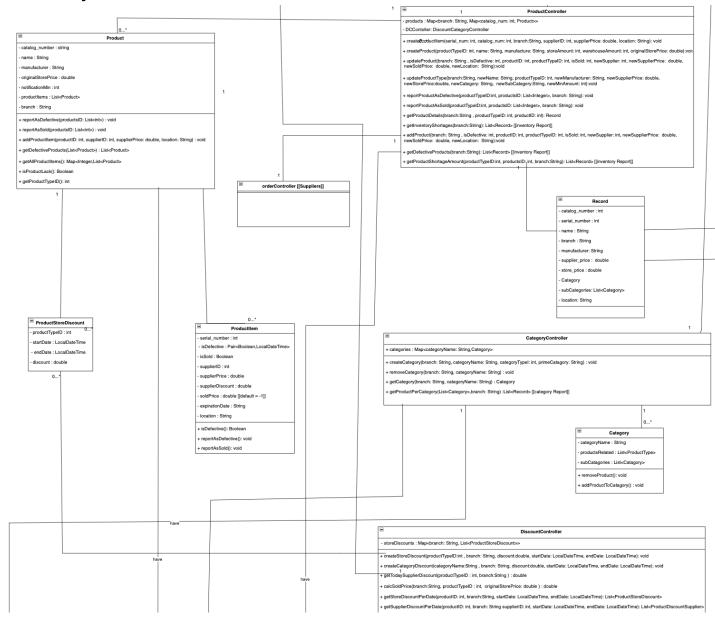


# Inventory:

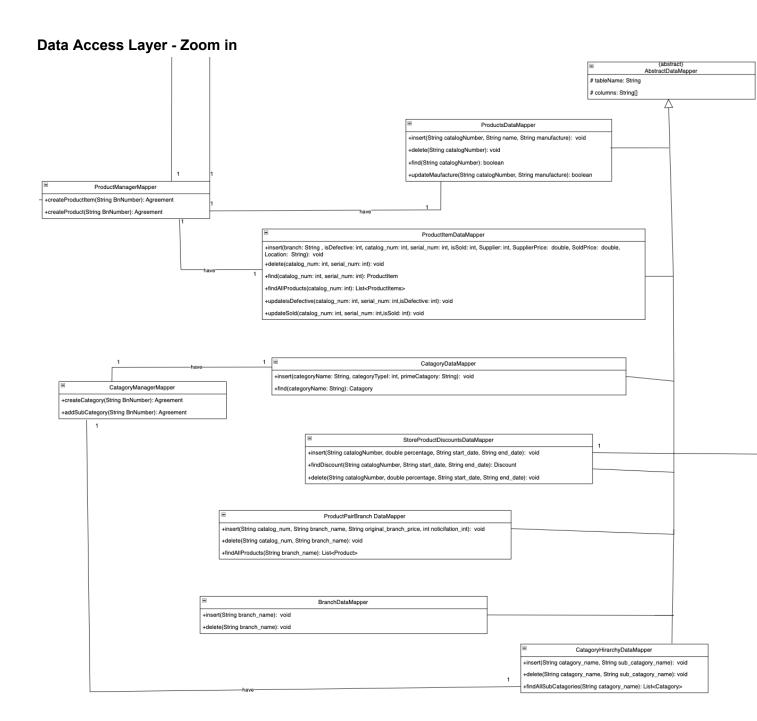
# Service Layer - Zoom in

<b>■</b> InventoryService	
+ getInventoryShortages(json: String) : Response	☐ CatagoriesService
+ removeProduct(json: String) : Response	
+ getStockProductsDetailed(json: String) : Response	+ removeMainCatagory(json: String) : Response
+ getStockProduct(json: String) : Response	+ removeSubCatagory(json: String) : Response
+ getDefectiveProducts(json: String) : Response	+ createSubCatagory(json: String) : Response
+ addProductType(json: String) : Response	+ createMainCatagory(json: String) : Response
+ updateProduct(json: String) : Response	
+ getStockProductsByCategory(json: String) : Response	
+ addProduct(json: String) : Response	0*
+ updateProductType(json: String) : Response	"
+ addStoreDiscount(json: String) : Response	
+ addProductDiscount(json: String) : Response	
+ addProductItemToBranch(json: String) : Response	
0*	

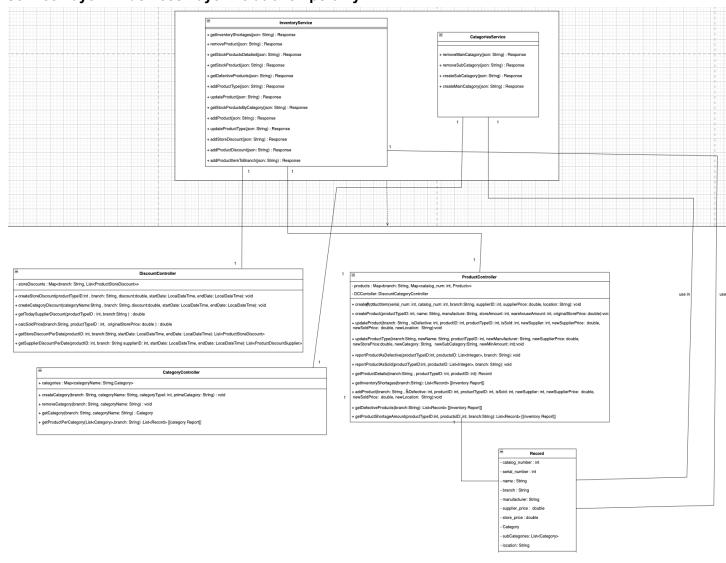
## **Business Layer - Zoom in**



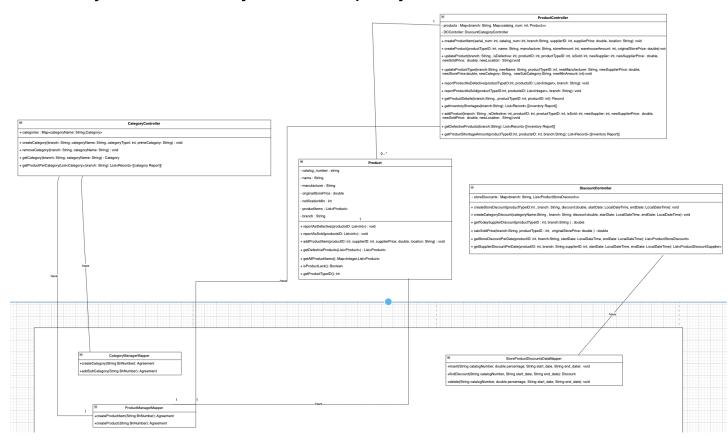
OrderController is the controller from the suppliers



# Service Layer + Business Layer Relationships only



# **Business Layer + Data Access Layer - Relationships only**



## Zoom out:

