

Storage **Presentation Layer** PresentationController CLI serviceController : ServiceController presentationController : PresentationController addProduct(catalogNumber, name, category, subCategory, size, buyPrice,salePrice,discount, isDamageReportAlerted: boolean supplierDiscount, manufacturer, aisle, minimalQuantity) : String isProductReportAlerted: boolean addToProduct(expirationDate, catalogNumber, inStore, inStorage) : String produceProductReport(categories) : String main(): void produceDamageReport() : String removeProduct(int catalogNumber) : String getCategoriesFromUser(scanner) : String alertOnMinimalQuantity() : String getCateegories() : String expiredCount() : String getSubCategories(): String getProduct(catalogNumber) : String getSubSubCategories() : String updateDiscountForProduct(catalogNumber, discount) : String resetFacotry() : void updateDiscountForCategory(categories, discount) : String productFactory() : void updateDamageForProduct(catalogNumber, inStore[], inStorage[], expirationDate[]) : String moveProductToStore(catalogNumber, expirationDate, quantity) : String subtractFromStore(catalogNumber, products) : String moveToExpired(catalogNumber, expirationDate, inStore, inStorage) : String parseAddProductMessage(str) : String parseRemoveProductMessage(str) : String parseUpdateDiscountForProductMessage(str) : String parseUpdateDamageForProductMessage(str) : String parseGetProductMessage(str) : String parseProduceProductReportMessage(str) : String parseProduceDamageReportMessage() : String parseUpdateDiscountForCategoryMessage(str) : String **Service Layer** ServiceController addProduct(catalogNumber, name, category, subCategory, size, buyPrice,salePrice,discount, supplierDiscount, manufacturer, aisle, minimalQuantity): void addToProduct(expirationDate, catalogNumber, inStore, inStorage) : void produceProductReport(categories) : String produceDamageReport() : String removeProduct(int catalogNumber) : void subtractFromStore(catalogNumber, products) : void moveProductToStore(catalogNumber, expirationDate, quantity) : void updateDamageForProduct(catalogNumber, inStore[], inStorage[], expirationDate[]) : void updateDiscountForCategory(categories, discount) : void updateDiscountForProduct(catalogNumber, discount) : void getProduct(catalogNumber) : String expiredCount() : String alertOnMinimalQuantity(): String moveToExpired(catalogNumber, expirationDate, inStore, inStorage) : void Domain Layer Product DomainManager catalogNumber : int Repository name : String categoriesList category : String subCategoriesList subCategory : String sizesList size : String addProduct(product) : void expirationDates : map<date, map<Integer, Integer>> getProduct(catalogNumber) : Product expiredProducts : map<date, Integer> removeProduct(product) : void buyPrice : double getProductsByCategories(categories) : List<Product> salePrice : double moveProductToStore(catalogNumber, expirationDate, quantity) : void discount : double subtractFromStore(catalogNumber, items) : void supplierDiscount : double expiredCount() : Map<int, int> storageQuantity : int alertOnMinimalQuantity() : String storeQuantity : int addToProduct(expirationDate, catalogNumber, inStore, inStorage) : void damageQuantity : int moveToExpired(catalogNumber, expirationDate, inStore, inStorage) : void manufacturer : String setDiscount(catalogNumber, discount) : void minimalQuantity : int aisle : String addByExpirationDate(amountForStore, amountForStorage, expirationDate) : void removeOne(isStorage, expirationDate) : void moveToDamage(inStore[], inStorage[], expirationDates) : void moveToExpired(inStore, inStorage, expirationDate) : void toString(): String productReportToString(): String damagedReportToString() : String moveProductToStore(expirationDate, quantity) : void expiredCount(): int Data Access Layer Repository ProductDAO productDAO : ProductDAO connectToDB(): getProduct(catalogNumber) : Product getProduct(catalogNumber) : Product addProduct(product) : void addProduct(product) : void updateProduct(product) : void updateProduct(product) : void deleteProduct(product) : void deleteProduct(product) : void getProductsByCategories(categories[]) : Product[] getProductsByCategories(categories[]) : Product[] expiredCount() : Map<int, int> expiredCount() : Map<int, int> alertOnMinimalQuantity() : String alertOnMinimalQuantity(): String