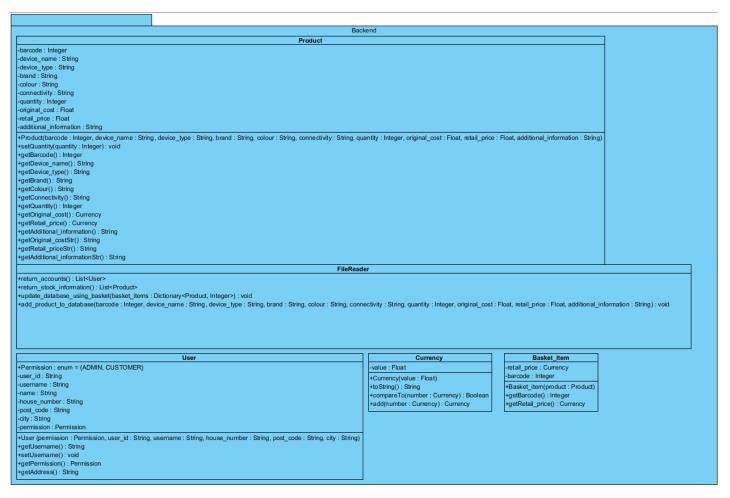Backend package: Deals with file handling and database modifications.

Backend

Product

-barcode : Integer
-device_name : String
-device_type : String
-brand : String
-colour : String
-connectivity : String
-quantity : Integer
-original_cost : Float
-retail_price : Float
-additional_information : String

+Product(barcode : Integer, device_name : String, device_type : String, brand : String, colour : String, connectivity : String, quantity : Integer, original_cost : Float, retail_price : Float, additional_information : String)
+setQuantity(quantity : Integer) : void
+getBarcode() : Integer
+getDevice_name() : String
+getDevice_type() : String
+getBrand() : String
+getColour() : String
+getConnectivity() : String
+getQuantity() : Integer
+getOriginal_cost() : Currency
+getRetail_price() : Currency
+getAdditional_information() : String
+getOriginal_costStr() : String
+getRetail_priceStr() : String
+getAdditional_informationStr() : String

FileReader

+return_accounts() : List<User>
+return_stock_information() : List<Product>
+update_database_using_basket(basket_items : Dictionary<Product, Integer>) : void
+add_product_to_database(barcode : Integer, device_name : String, device_type : String, brand : String, colour : String, connectivity : String, quantity : Integer, original_cost : Float, retail_price : Float, additional_information : String) : void

User

+Permission : enum = {ADMIN, CUSTOMER}
-user_id : String
-username : String
-name : String
-house_number : String
-post_code : String
-city : String
-permission : Permission

+User (permission : Permission, user_id : String, username : String, house_number : String, post_code : String, city : String)
+getUsername() : String
+setUsername() : void
+getPermission() : Permission
+getAddress() : String

Currency

-value : Float

+Currency(value : Float)
+toString() : String
+compareTo(number : Currency) : Boolean
+add(number : Currency) : Currency

Basket_item

-retail_price : Currency
-barcode : Integer

+Basket_item(product : Product)
+getBarcode() : Integer
+getRetail_price() : Currency

Product

-barcode : Integer
-device_name : String
-device_type : String
-brand : String
-colour : String
-connectivity : String
-quantity : Integer
-original_cost : Float
-retail_price : Float
-additional_information : String

+Product(barcode : Integer, device_name : String, device_type : String, brand : String, colour : String, connectivity : String, quantity : Integer, original_cost : Float, retail_price : Float, additional_information : String)
+setQuantity(quantity : Integer) : void
+getBarcode() : Integer
+getDevice_name() : String
+getDevice_type() : String
+getBrand() : String
+getColour() : String
+getConnectivity() : String
+getQuantity() : Integer
+getOriginal_cost() : Currency
+getRetail_price() : Currency
+getAdditional_information() : String
+getOriginal_costStr() : String
+getRetail_priceStr() : String
+getAdditional_informationStr() : String

Filereader handles reading and writing to database and making copies of its data.

FileReader

+return_accounts() : List<User>
+return_stock_information() : List<Product>
+update_database_using_basket(basket_items : Dictionary<Product, Integer>) : void
+add_product_to_database(barcode : Integer, device_name : String, device_type : String, brand : String, colour : String, connectivity : String, quantity : Integer, original_cost : Float, retail_price : Float, additional_information : String) : void

User
+Permission : enum = {ADMIN, CUSTOMER} -user_id : String -username : String -name : String -house_number : String -post_code : String -city : String -permission : Permission
+User (permission : Permission, user_id : String, username : String, house_number : String, post_code : String, city : String) +getUsername() : String +setUsername() : void +getPermission() : Permission +getAddress() : String

Currency is a custom class that lets me manipulate floats as

Currency
-value : Float
+Currency(value : Float) +toString() : String +compareTo(number : Currency) : Boolean +add(number : Currency) : Currency

Basket_item
-retail_price : Currency -barcode : Integer
+Basket_item(product : Product) +getBarcode() : Integer +getRetail_price() : Currency

GUI package: Deals with frontend.

I tried to only include the most relevant parts of the javafx.

GUI

LoginController

-scene : Scene

+getScene() : Scene
+setup(scene : Scene) : void
+openStore(event : ActionEvent) : void

SceneController

+stage : Stage
+static scenes : enum = {LOGIN, STORE}
+active_user : User

+start(arg0 : Stage) : void
+switch_scene(scene : scenes) : void

StoreController

-store_scene : Scene
-stock : ObservableList<Product>
-basket_items : ObservableList<Product>
-item_quantities : Dictionary<Product, Integer>

+getScene() : scene
+add_Listeners() : void
+add_product_input_validated() : Boolean
+clear_basket() : void
+populate_store_tableview(numkeys : Integer)
+set_item_quantities() : void
+setup_basket_tableview() : void
+update_store_according_permissions() : void
+update_total_cost() : void
+isNumeric() : Boolean
+validate_email() : Boolean
+validate_card_details() : Boolean
+checkout(payment_method : String) : void
+setup(new_scene : Scene)