UserParser
<<utlility>>
UserParser

+parseUsers(String filename, ArrayList<product> products) : ArrayList<Product>

Receipt
Receipt

-receiptString: String

+toString(): String

PaymentMethod
<<interface>>
PaymentMethod

+processPayment(double amount, Address fullAddress) : Receipt

ProductParser
<<utlility>>
ProductParser

+parseProducts(String filename) : ArrayList<Product>

PayPal
PayPal

-email: String

+getEmail(): String
+processPayment(double amount, Address fullAddress) : Receipt

implements

CreditCard
CreditCard

-cardNum: int
-securityNum: int

+getSecurityNum() : int
+getCardNum(): int
+processPayment(double amount, Address fullAddress) : Receipt

implements

User (abstract)
User (abstract)

- userId : int
- username: String
- name: String
- address: Address
- role: Role

+ getUserId(): int
+ getUsername(): String
+getName(): String
+getAddress(): Address
+getRole(): Role
+toString(): String
+viewAllProducts(ArrayList<Product> products) : ArrayList<String>

Role
<<enumeration>>
Role

ADMIN
CUSTOMER

Address
Address

-houseNum: int
- postcode: String
- city: String

+toString() : String

ProductString
<<utlility>>
ProductString

+getProductForCustomer(Product product) : String

Customer
Customer

-productsInBasket: ArrayList<Product>

+viewAllProducts(ArrayList<Product> products) : ArrayList<String>
+getProductForCustomer(Product product) : String
+addToBasket(Product product) : void
+clearBasket() : void
+getTotalCost() : double
+viewBasket() : ArrayList<Product>
+completePayment(PaymentMethod paymentMethod, Address address) : Receipt
+validEmail(String email) : bool
+validCreditCardDetails(int cardNum, int securityNum) : boolean
+updateStock() : void
-isAvailable(Product product, int quantityBought) : boolean
+searchByBarcode(ArrayList<Product> products, int barcode) : Product
+filterMiceByButtons(ArrayList<Product> products, int buttons) : ArrayList<Product>
+toString() : String

Admin
Admin

-products: ArrayList<Product>

+viewAllProducts(ArrayList<product> products) : ArrayList<String>
+addKeyboardStock(int barcode, String brand, String color, ConnectivityType connectivity, int quantityInStock, double originalCost, double retailPrice, KeyboardLayout layout, KeyboardType keyboardtype) : void

+addMouseStock(int barcode, String brand, String colour, ConnecitivityType connectivity, int quantity, double originalCost, double retailPrice, MouseType type, int numOfButtons) : void

+toString() : String

ProductCategory
<<enumeration>>
ProductCategory

KEYBOARD
MOUSE

Product (abstract)
Product (abstract)

- barcode: int
- brand: String
-colour: String
- connectivity: ConnectivityType
- quantityInStock: int
- originalCost : double
- retailPrice: double
-category: ProductCategory

+getBarcode(): int
+getBrand(): String
+getColor(): String
+getConnectivity(): ConnectivityType
+getQuantityInStock(): int
+setQuantityInStock(int quantityInStock) : void
+getOriginalCost(): double
+getRetailPrice(): double
+getCategory(): ProductCategory
+toString() : String

ConnectivityType
<<enumeration>>
ConnectivityType

WIRED
WIRELESS

Mouse
Mouse

-type: MouseType
-numOfButtons: int

+getType(): MouseType
+getNumOfButtons() : int
+toString(): String
+toStringForCustomer(): String

Keyboard
Keyboard

-layout: KeyboardLayout
-type: KeyboardType

+getLayout(): KeyboardLayout
+getType(): KeyboardType
+toString(): String
+toStringForCustomer: String

MouseType
<<enumeration>>
MouseType

STANDARD
GAMING

KeyboardLayout
<<enumeration>>
KeyboardLayout

UK
US

KeyboardType
<<enumeration>>
KeyboardType

STANDARD
FLEXIBLE
GAMING

Key

association | Generalisation | Aggregation | Composition | Implements