

allitems(Singleton)

ProductList: ArrayList<Product>

- totalProfit: double

totalRevenue: double

nextUPC: int

StoreRecord: allItems

allItem():void

- getStoreRecord(): allItems

- getProductList(): ArrayList<Product>

+ getAllSale(): ArrayList<Product>

- getNextUPC(): int

+ getProductByAisle(int a): ArrayList<Produc

+ replace(Object r): void

readResolve(): Object

+ getNum(): int

+ changeAisle(int a, int b): void

+ getProduct(String upc): Product

+ addProfit(double p): void

+ getProfit(): double

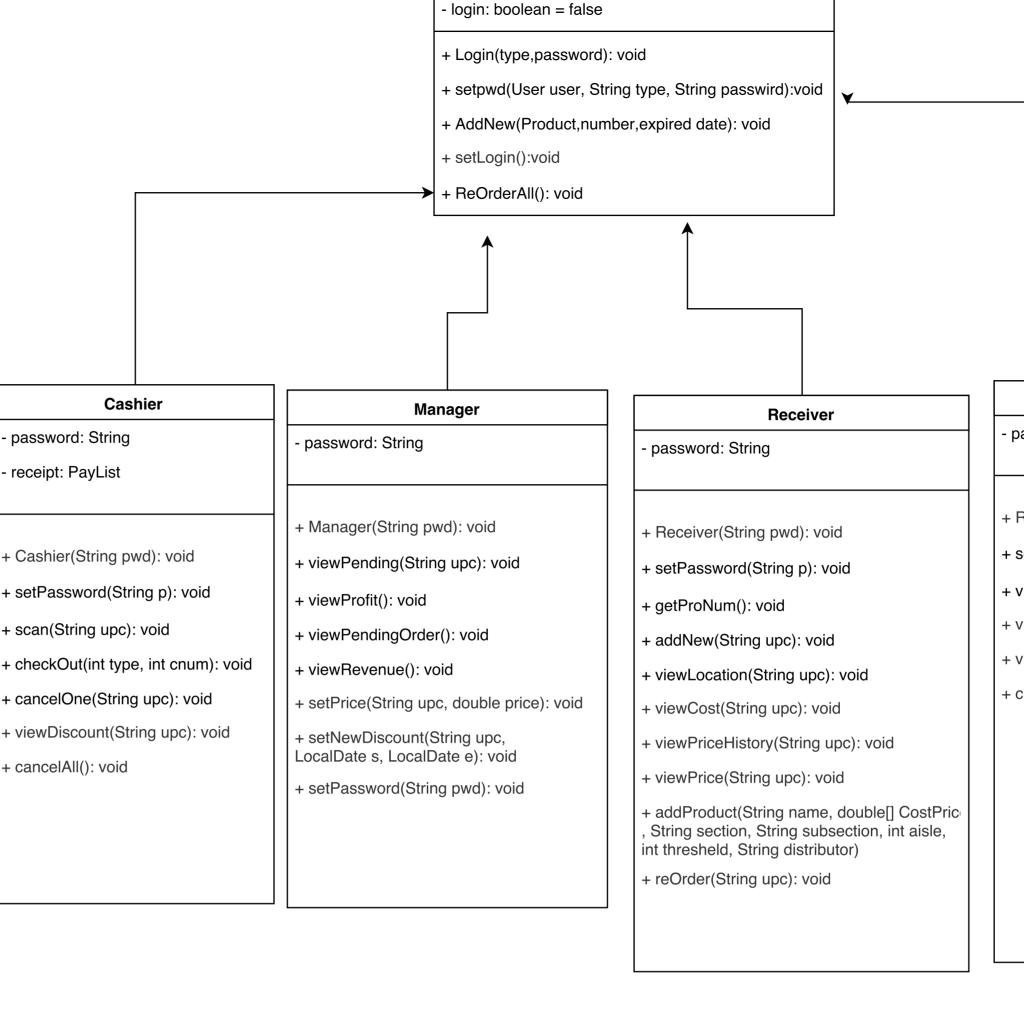
- addProduct(Product p): void

- getPendingList(): String

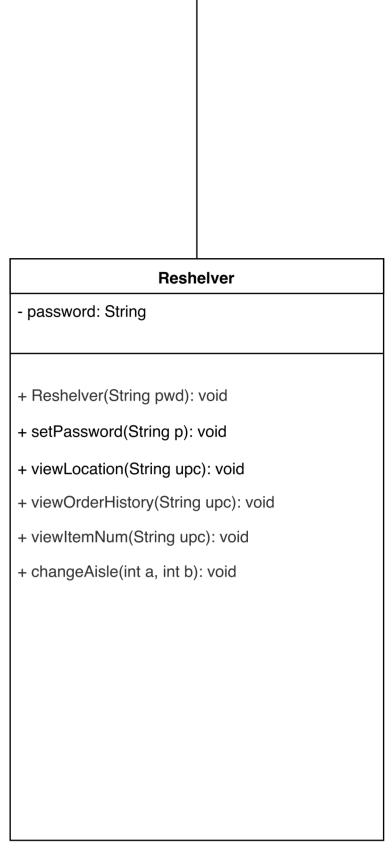
+ addRevenue(double r): void

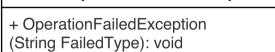
- addProduct(Product): void

- today: LocalDate endDate: LocalDate - priceHistory: HashMap<LocalDate, double> discountType: int - name: String discount: double - section: String - buyX: int - Subsection: String - nextY: int - aisle: int - threshold: int + Discount(double off, LocalDate start, LocalDate end): void - price: double pendingNumber: int + getEndDate(): LocalDate - discounts: Discount + getStartDate(): LocalDate distributor: String + getPrize(double prize, int num): double - orderHistory: ArrayList<LocalDate> + toString(): String - cost: double upc: String - itemNum: int NeedReOrder: boolean + Product(String name, double[] CostPrice, String psection, String psubsection, int aisle, int r, String pdistributor) + addNew(): void + removeItem(int num): void + addDiscount(double d, LocalDate start, LocalDate end): void + checkout(int): double + getName(): String + getSection(): String + getAisle(): int + setAisle(int aisle): void + getThreshold(): int + getPrice(): double + setPrice(double price): void + getPendingNumber(): int



User





OperationFailedException

Exception

+ OperationFailedException (User user): void

HandleEvent

- user: User
- login: boolean

- + getDiscounts(): Discount
- + getOrderHistory(): ArrayList<LocalDate>
- + getCost(): double
- + getUpc(): String
- + getItemNum(): String
- + requestReorder(): void
- + isNeedReOrder(): boolean
- + checkout(int num): double
- + toString(): String

- ourInstance: HandleEvent
- +HandleEvent(): void
- + getEvent(): HandleEvent
- + isLogin(): boolean
- + passMessage(String message): void
- + passCommand(String[] command): void

PayList - payList: Map<Product, Integer>

- nextID: int = 1
- productsList: Array
- totalRevenue: double
- toalPrice: double
- PaymentType: int = -1
- CardNum: int = -1
- + PayList(): void
- + addItem(String upc): void
- + cancel(String upc): void
- + purchase(int type, int cnum): void

shopSimulation

+ Main(String[] args): void