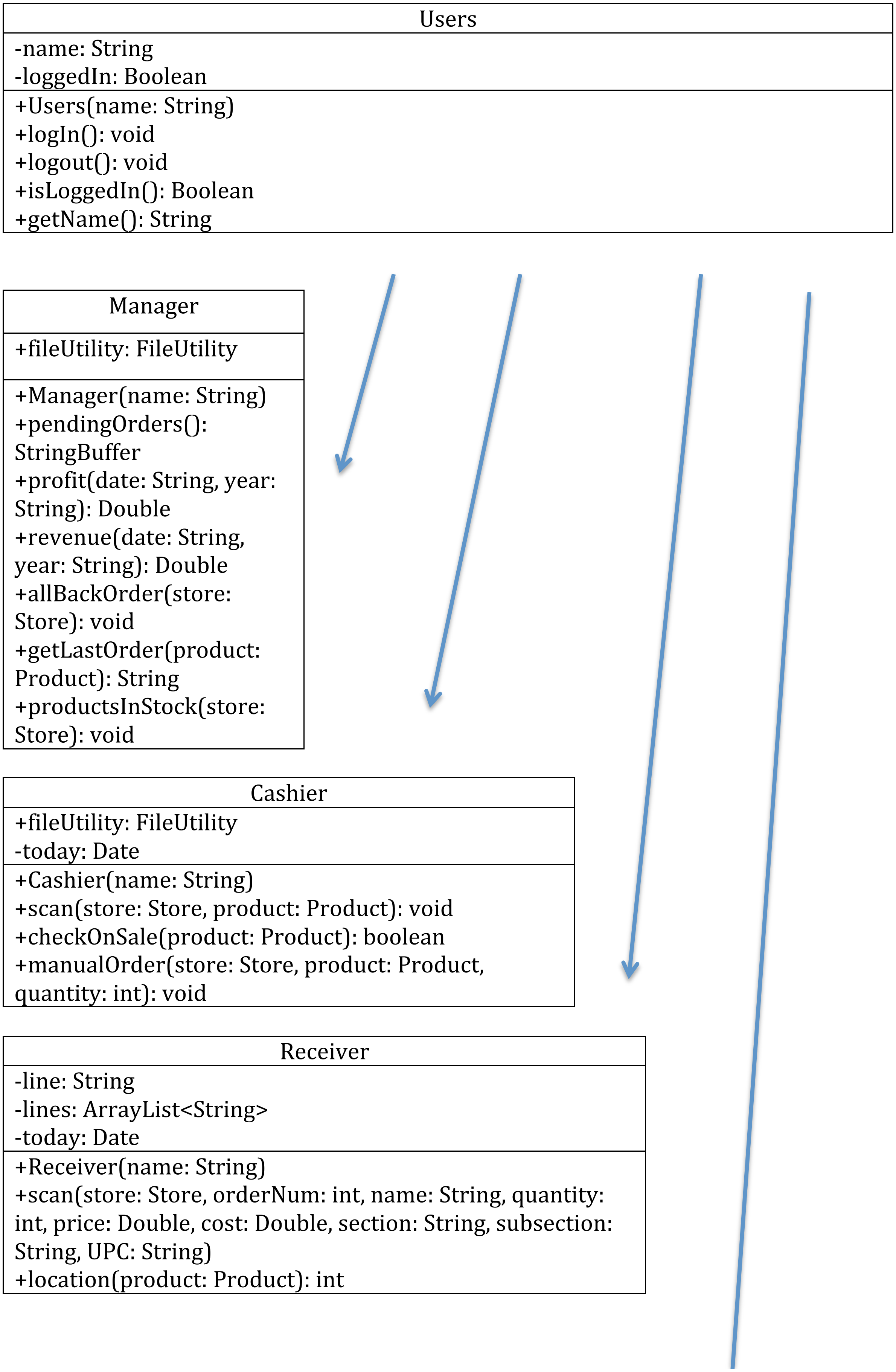
UML Diagrams

Location
-section: String -subsection: String -aisle: int
+Location(section: String, subsection: String, aisle: int) +getSection(): String +getAisle(): int +setAisle(num: int): void

Price
-price: Double -salePrice: Double -cost: Double -currPrice: Double -sale: Boolean -startDate: Date -endDate: Date
+Price(price: Double, cost: Double) +priceSetSale(price: Double, startDate: Date, endDate: Date): void +putOnSale(): void +takeOffSale(): void +onSale(): Boolean +getPrice(): Double +getCurrPrice(): Double +getCost(): Double +getStartDate(): Date +getEndDate(): Date

Section
-name: String -store: Store -section: HashMap<String, HashMap<String, Product>>
+Section(store: Store, name: String) +addProduct(product: Product): void +addSubsection(subsection: String): void +inSection(product: Product): boolean +getSection():HashMap<String, HashMap<String, Product>> +getSubsection(subsection: String): HashMap<String, Product>

UML Diagrams



Users
-name: String -loggedIn: Boolean
+Users(name: String) +logIn(): void +logout(): void +isLoggedIn(): Boolean +getName(): String

Manager
+fileUtility: FileUtility
+Manager(name: String) +pendingOrders(): StringBuffer +profit(date: String, year: String): Double +revenue(date: String, year: String): Double +allBackOrder(store: Store): void +getLastOrder(product: Product): String +productsInStock(store: Store): void

Cashier
+fileUtility: FileUtility -today: Date
+Cashier(name: String) +scan(store: Store, product: Product): void +checkOnSale(product: Product): boolean +manualOrder(store: Store, product: Product, quantity: int): void

Receiver
-line: String -lines: ArrayList<String> -today: Date

+Receiver(name: String)
+scan(store: Store, orderNum: int, name: String, quantity: int, price: Double, cost: Double, section: String, subsection: String, UPC: String)
+location(product: Product): int

UML Diagrams

+checkCost(product: Product): Double +checkCurrPrice(product: Product): Double +priceHistory(product: Product): StringBuffer

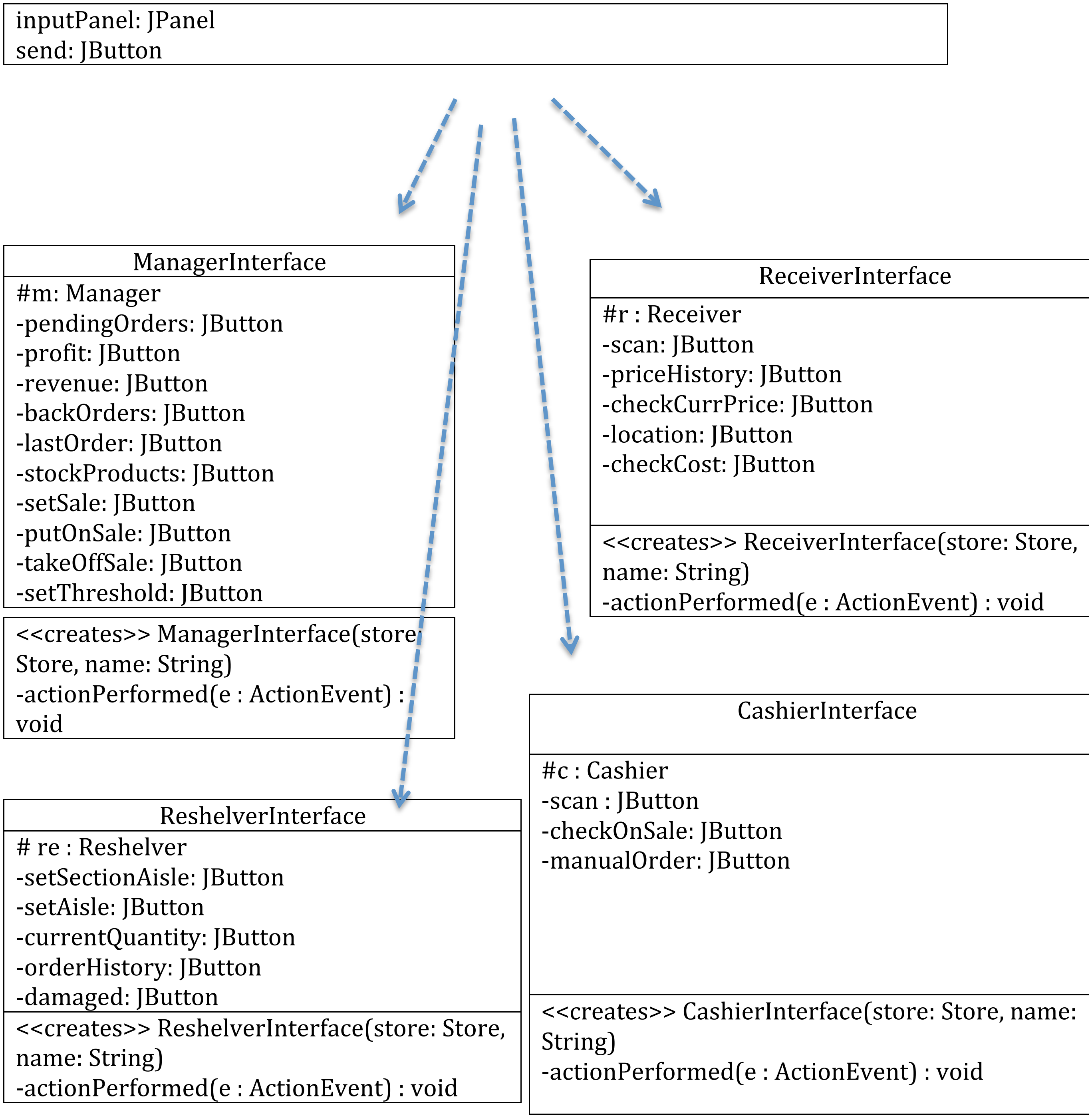
Reshelver
-today: Date +fileUtility: FileUtility
+Reshelver(name: String) +orderHistory(product: Product): StringBuffer +currentQuality(product: Product): int +setAisle(product: Product, aisle: int): void +setSectionAisle(store: Store, section: String, aisle: int): void +scanDamageItem(store: Store, product: Product): void

Store
-name: String -storeMap: HashMap<String, Section> -allSections: ArrayList<String> -allProducts: <String, Product> -orderNum: int
+Store(name: String) +addSection(name: String): void +getAllProducts():HashMap<String, Product> +getOrderNum(): int +incrementOrder(): void +getName(): String +setName(name: String): void

groceryInterface
main(arg : String{}) : void

GenericInterface
bottom: JPanel top: Jpanel send:Jbutton output: JtextArea input: JtextField data: String lastButton: String split: JSplitPane scroll: JScrollPanel

UML Diagrams



inputPanel: JPanel
send: JButton

ManagerInterface

#m: Manager
-pendingOrders: JButton
-profit: JButton
-revenue: JButton
-backOrders: JButton
-lastOrder: JButton
-stockProducts: JButton
-setSale: JButton
-putOnSale: JButton
-takeOffSale: JButton
-setThreshold: JButton

<<creates>> ManagerInterface(store: Store, name: String)
-actionPerformed(e : ActionEvent) : void

ReceiverInterface

#r : Receiver
-scan: JButton
-priceHistory: JButton
-checkCurrPrice: JButton
-location: JButton
-checkCost: JButton

<<creates>> ReceiverInterface(store: Store, name: String)
-actionPerformed(e : ActionEvent) : void

CashierInterface

#c : Cashier
-scan : JButton
-checkOnSale: JButton
-manualOrder: JButton

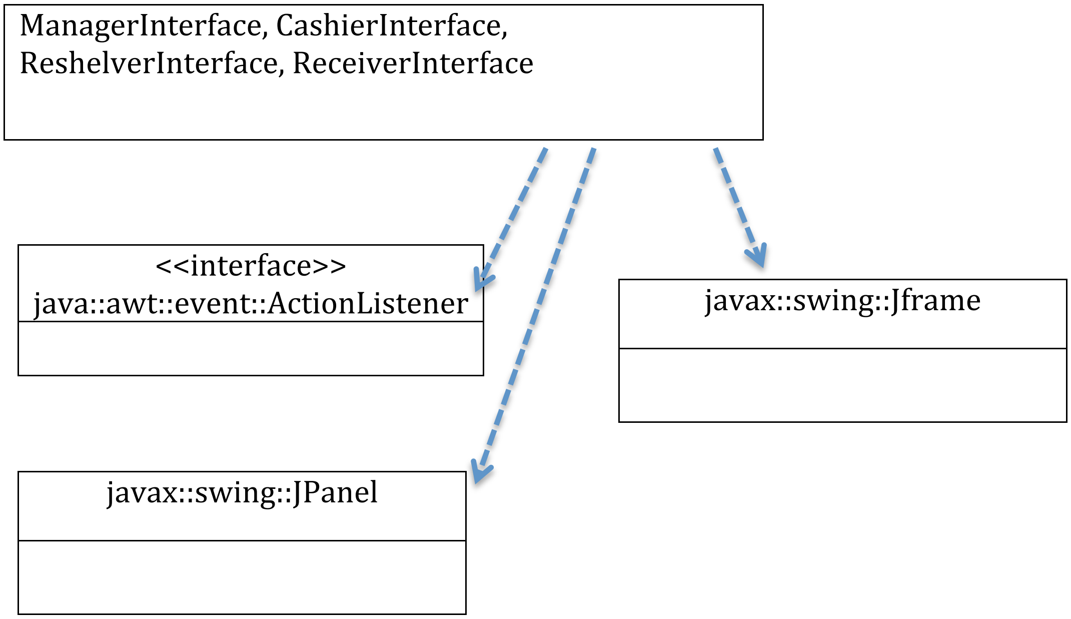
<<creates>> CashierInterface(store: Store, name: String)
-actionPerformed(e : ActionEvent) : void

ReshelverInterface

# re : Reshelver
-setSectionAisle: JButton
-setAisle: JButton
-currentQuantity: JButton
-orderHistory: JButton
-damaged: JButton

<<creates>> ReshelverInterface(store: Store, name: String)
-actionPerformed(e : ActionEvent) : void

UML Diagrams


ManagerInterface, CashierInterface, ReshelverInterface, ReceiverInterface
<<interface>>
java::awt::event::ActionListener
javax::swing::Jframe
javax::swing::JPanel

UML Diagrams

Product

-name: String
-quantity: int
-price: Price
-location: Location
-threshold: int
-UPC: String
-onBackOrder: Boolean
-orderNum: int
-today: Date

+Product(name: String, quantity: int, price: Price, location: Location, UPC: String)
+getName(): String
+getQuantity(): int
+decrement(): void
+add(quantity: int): void
+getUPC(): String
+setSale(price: Double, startDate: Date, endDate: Date): void
+putOnSale(): void
+takeOffSale(): void
+getPrice(): Price
+getLocation(): Location
+getThreshold(): int
+setThreshold(threshold: int): int
+underThreshold(): boolean
+onBackOrder(): boolean
+setOnBackOrder(): void
+getOrderNum(): int
+setOrderNum(num: int): void

FileUtility

+FileWriter(filename: String, data: String): void
+ProductFileScanner(filename: String, object: String): StringBuffer
+SumScanner(position: int, date: String, year: String): Double