SENG300 P3-G1

Derek Atabayev : 30177060
Enioluwafe Balogun : 30174298
Subeg Chahal : 30196531
Jun Heo : 30173430
Emily Kiddle : 30122331
Anthony Kostal-Vazquez : 30048301
Jessica Li : 30180801
Sua Lim : 30177039
Savitur Maharaj : 30152888
Nick McCamis : 30192610
Ethan McCorquodale : 30125353
Katelan Ng : 30144672
Arcleah Pascual : 30056034
Dvij Raval : 30024340
Chloe Robitaille : 30022887
Danissa Sandykbayeva : 30200531
Emily Stein : 30149842
Thi My Tuyen Tran : 30193980
Aoi Ueki : 30179305
Ethan Woo : 30172855
Kingsley Zhong : 30197260

SelfCheckoutStationLogic class:

- Connects all elements of software together
- Install selfCheckoutStation and Session

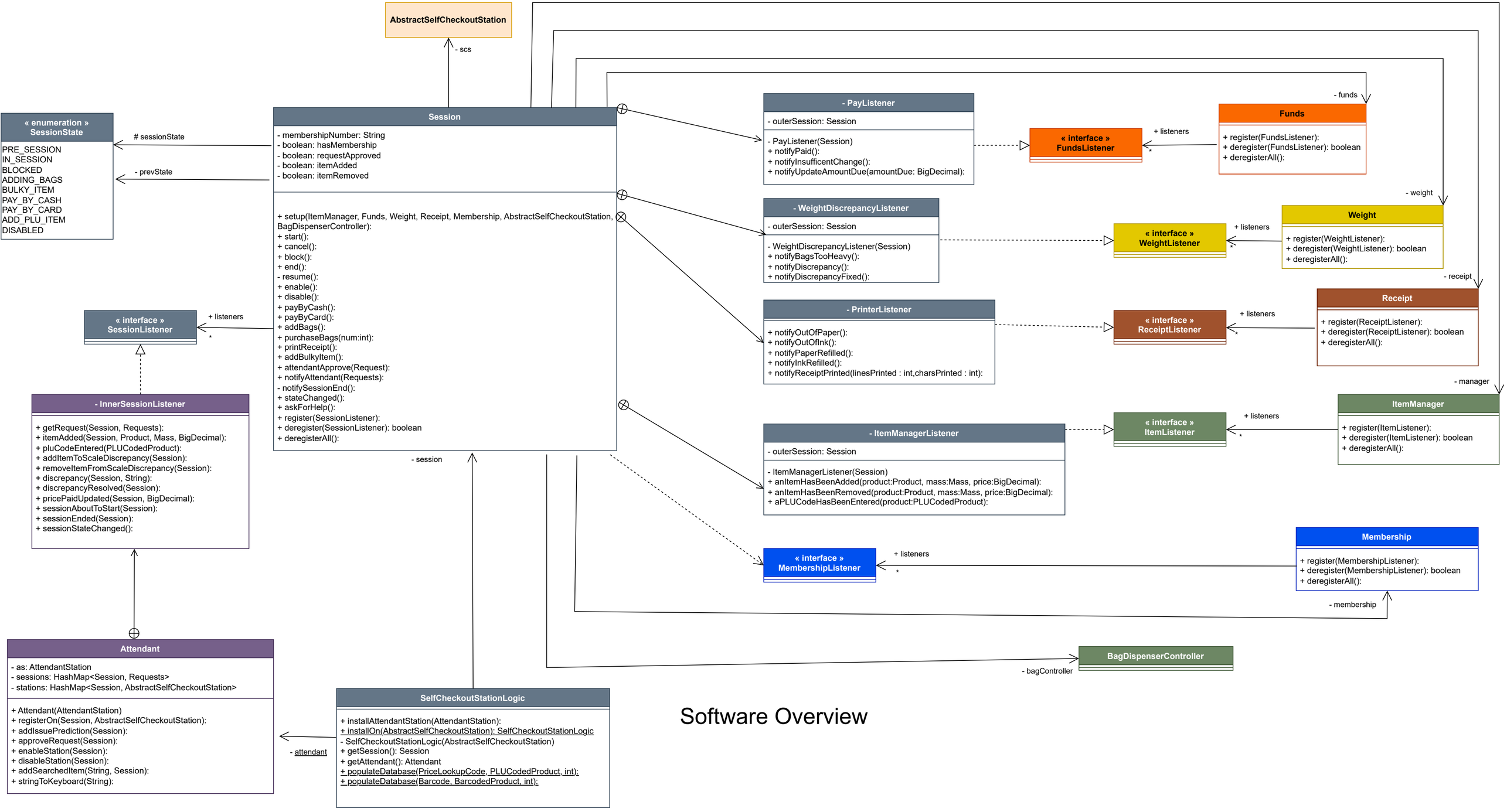
Session has the following:

- Manages the state of the session
- Has the actions that customers can perform for a self checkout system such as:
    - Start session
    - add/remove items of various types
    - Purchase bags and add own bags
    - Pay for items via cash/card (credit & debit)
    - Enter membership card
    - Get receipt
    - Ask/signal for attendant's help
- Weight package:
    - Updates the weight as items are added to the system
    - Checks for weight discrepancy
- Funds package:
    - Keeps track of the total price of items, the amount paid by the customer, and the remaining amount due by checking status of payment
    - Supports different modes of payment such as pay via cash and pay via card
- Receipt package:
    - Prints receipt once payment has been made
- Items package:
    - Defines a rule for adding items based on different modes of adding items
    - Manages aspects to adding items to an order (scan, visual catalogue, PLU, bulky item)
    - Adds reusable bags (and places in the dispenser)
- Membership package:
    - Defines the instances of members and provides functionality for entering membership numbers

General features for the Attendant (attendant package):

- Attendant session is connected to their own station; listen to all registered customer sessions
- Attendant can do the following to resolve customers' issues:
    - Maintain ink, paper, coin and banknote
    - Enable/disable session
    - Handle weight discrepancy by removing/adding items and/or bags
    - Add item via text search
    - Predict an error pre or post session

All these features are interacted through a graphical user interface for the self-checkout station software, hardware and for the attendant station, developed using the Java Swing library. That allows you to visually interact with all of the components a part of the software.


AbstractSelfCheckoutStation
- scs
« enumeration »
SessionState
PRE_SESSION
IN_SESSION
BLOCKED
ADDING_BAGS
BULKY_ITEM
PAY_BY_CASH
PAY_BY_CARD
ADD_PLU_ITEM
DISABLED
sessionState
- prevState
Session
- membershipNumber: String
- boolean: hasMembership
- boolean: requestApproved
- boolean: itemAdded
- boolean: itemRemoved
+ setup(ItemManager, Funds, Weight, Receipt, Membership, AbstractSelfCheckoutStation, BagDispenserController):
+ start():
+ cancel():
+ block():
+ end():
- resume():
+ enable():
+ disable():
+ payByCash():
+ payByCard():
+ addBags():
+ purchaseBags(num:int):
+ printReceipt():
+ addBulkyItem():
+ attendantApprove(Request):
+ notifyAttendant(Requests):
- notifySessionEnd():
+ stateChanged():
+ askForHelp():
+ register(SessionListener):
+ deregister(SessionListener): boolean
+ deregisterAll():
« interface »
SessionListener
+ listeners
*
- InnerSessionListener
+ getRequest(Session, Requests):
+ itemAdded(Session, Product, Mass, BigDecimal):
+ pluCodeEntered(PLUCodedProduct):
+ addItemToScaleDiscrepancy(Session):
+ removeItemFromScaleDiscrepancy(Session):
+ discrepancy(Session, String):
+ discrepancyResolved(Session):
+ pricePaidUpdated(Session, BigDecimal):
+ sessionAboutToStart(Session):
+ sessionEnded(Session):
+ sessionStateChanged():
- session
Attendant
- as: AttendantStation
- sessions: HashMap<Session, Requests>
- stations: HashMap<Session, AbstractSelfCheckoutStation>
+ Attendant(AttendantStation)
+ registerOn(Session, AbstractSelfCheckoutStation):
+ addIssuePrediction(Session):
+ approveRequest(Session):
+ enableStation(Session):
+ disableStation(Session):
+ addSearchedItem(String, Session):
+ stringToKeyboard(String):
- attendant
SelfCheckoutStationLogic
+ installAttendantStation(AttendantStation):
+ installOn(AbstractSelfCheckoutStation): SelfCheckoutStationLogic
- SelfCheckoutStationLogic(AbstractSelfCheckoutStation)
+ getSession(): Session
+ getAttendant(): Attendant
+ populateDatabase(PriceLookupCode, PLUCodedProduct, int):
+ populateDatabase(Barcode, BarcodedProduct, int):
- PayListener
- outerSession: Session
- PayListener(Session)
+ notifyPaid():
+ notifyInsufficentChange():
+ notifyUpdateAmountDue(amountDue: BigDecimal):
« interface »
FundsListener
+ listeners
*
Funds
+ register(FundsListener):
+ deregister(FundsListener): boolean
+ deregisterAll():
- funds
- WeightDiscrepancyListener
- outerSession: Session
- WeightDiscrepancyListener(Session)
+ notifyBagsTooHeavy():
+ notifyDiscrepancy():
+ notifyDiscrepancyFixed():
« interface »
WeightListener
+ listeners
*
Weight
+ register(WeightListener):
+ deregister(WeightListener): boolean
+ deregisterAll():
- weight
- PrinterListener
+ notifyOutOfPaper():
+ notifyOutOfInk():
+ notifyPaperRefilled():
+ notifyInkRefilled():
+ notifyReceiptPrinted(linesPrinted : int,charsPrinted : int):
« interface »
ReceiptListener
+ listeners
*
Receipt
+ register(ReceiptListener):
+ deregister(ReceiptListener): boolean
+ deregisterAll():
- receipt
- ItemManagerListener
- outerSession: Session
- ItemManagerListener(Session)
+ anItemHasBeenAdded(product:Product, mass:Mass, price:BigDecimal):
+ anItemHasBeenRemoved(product:Product, mass:Mass, price:BigDecimal):
+ aPLUCodeHasBeenEntered(product:PLUCodedProduct):
« interface »
ItemListener
+ listeners
*
ItemManager
+ register(ItemListener):
+ deregister(ItemListener): boolean
+ deregisterAll():
- manager
« interface »
MembershipListener
+ listeners
*
Membership
+ register(MembershipListener):
+ deregister(MembershipListener): boolean
+ deregisterAll():
- membership
BagDispenserController
- bagController
Software Overview

SelfCheckoutStationLogic

+ installAttendantStation(AttendantStation):
+ installOn(AbstractSelfCheckoutStation): SelfCheckoutStationLogic
- SelfCheckoutStationLogic(AbstractSelfCheckoutStation)
+ getSession(): Session
+ getAttendant(): Attendant
+ populateDatabase(PriceLookupCode, PLUCodedProduct, int);
+ populateDatabase(Barcode, BarcodedProduct, int);

- session

Session

+ enable():
+ disable():
+ attendantApprove(Request):
+ notifyAttendant(Requests):
- notifySessionEnd():
+ notifyCloseHardware():
+ notifyOpenHardware():
+ stateChanged():
+ askForHelp():
+ register(SessionListener):
+ deregister(SessionListener): boolean
+ deregisterAll():
+ register(HardwareListener):
+ deregister(HardwareListenerr): boolean
+ deregisterAll():

- session

« enumeration » Requests

NO_REQUEST
WEIGHT_DISCREPANCY
BAGS_TOO_HEAVY
BULKY_ITEM
CANT_MAKE_CHANGE
CANT_PRINT_RECEIPT
ADD_ITEM_SEARCH
HELP_REQUESTED

TextSearchController

- searchField: String
- searchFieldSB: StringBuilder
- searchResults: HashMap<String, Product>
- shift: boolean

+ TextSearchController(IKeyboard):
+ getSearchResults(): HashMap<String, Product>
+ getSearchField(): String

- InnerListener

+ aDeviceHasBeenDisabled(IDevice<? extends IDeviceListener>:
+ aDeviceHasBeenEnabled(IDevice<? extends IDeviceListener>:
+ aDeviceHasBeenTurnedOff(IDevice<?> extends IDeviceListener>:
+ aDeviceHasBeenTurnedOn(IDevice<?> extends IDeviceListener>:
+ aKeyHasBeenReleased(label:String):
+ aKeyHasBeenPressed(label:String):

TouchScreenListener

KeyboardListener

AttendantStation

- as

- ts

- attendant

Attendant

- sessions: HashMap<Sessions, Requests>
- predictors: HashMap<Session, IssuePreductor>
- stations: HashMap<Session, AbstractSelfCheckoutStation>

+ method(type): type

IssuePredictor

- estimatedInk: int
- estimatedPaper: int
- lowInk: boolean
- lowPaper: boolean
- lowCoins: boolean
- fullCoins: boolean
- lowBanknotes: boolean
- fullBanknotes: boolean

+ IssuePredictor(Session, AbstractSelfCheckoutStation):
- predictionCheck(Session):
+ checkLowInk(Session, IReceiptPrinter):
+ checkLowPaper(Session, IReceiptPrinter):
+ checkLowCoins(Session, HashMap<BigDecimal, ICoinDispenser>):
+ checkLowBanknotes(Session, HashMap<BigDecimal, IBanknoteDispenser>):
+ checkCoinsFull(Session, CoinStorageUnit):
+ checkBanknotesFull(Session, BanknoteStorgeUnit):
- notifyLowPaper(Session):
- notifyCoinsLow(Session):
- notifyBanknotesLow(Session):
- notifyCoinsFull(Session):
- notifyBanknotesFull(Session):
- notifyNoIssues(Session):
+ register(IssuePredictorListener):
+ deregister(IssuePredictorListener): boolean
+ deregisterAll():

- banknoteStorage

BanknoteStorageUnit

- coinStorage

CoinStorageUnit

- banknoteDispensers

Map<BigDecimal, IBanknoteDispenser>

- coinDispensers

Map<BigDecimal, ICoinDispenser>

- receiptPrinter

IReceiptPrinter

- InnerSessionListener

+ getRequest(Session, Requests):
+ itemAdded(Session, Product, Mass, BigDecimal):
+ pluCodeEntered(PLUCodedProduct):
+ addItemToScaleDiscrepancy(Session):
+ removeItemFromScaleDiscrepancy(Session):
+ discrepancy(Session, String):
+ discrepancyResolved(Session):
+ pricePaidUpdated(Session, BigDecimal):
+ sessionAboutToStart(Session):
+ sessionEnded(Session):
+ sessionStateChanged():

- InnerPredictionListener

+ notifyPredictedLowInk(Session):
+ notifyPredictedLowPaper(Session):
+ notifyPredictCoinsFull(Session):
+ notifyPredictBanknotesFull(Session):
+ notifyPredictLowCoins(Session):
+ notifyPredictLowBanknotes(Session):
+ notifyNoIssues(Sessions):

- listeners

« interface » IssuePredictorListener

- listeners

« interface » SessionListener

- listeners

- InnerSessionListener

+ getRequest(Session, Requests):
+ itemAdded(Session, Product, Mass, BigDecimal):
+ pluCodeEntered(PLUCodedProduct):
+ addItemToScaleDiscrepancy(Session):
+ removeItemFromScaleDiscrepancy(Session):
+ discrepancy(Session, String):
+ discrepancyResolved(Session):
+ pricePaidUpdated(Session, BigDecimal):
+ sessionAboutToStart(Session):
+ sessionEnded(Session):
+ sessionStateChanged():

- InnerReceiptListener

+ notifyOutOfPaper():
+ notifyOutOfInk():
+ notifyPaperRefilled():
+ notifyInkRefilled():
+ notifyReceiptPrinted(linesPrinted : int,charsPrinted : int):

- listeners

« interface » HardwareListener

- listeners

- InnerHardwareListener

+ aStationHasBeenOpened():
+ aStationHasBeenClosed():

Receipt

- listeners

« interface » ReceiptListener

MaintenanceManager

- isOpen:boolean
- amountOfInkRefilled: int
- amountOfPaperRefilled: int
- state: SessionState

+ openHardware(Session):
+ addCoins(BigDecimal, Coin):
+ removeCoins(): List<Coin>
- verifyBanknoteDenomination(BigDecimal): boolean
+ addBanknotes(BigDecimal: Banknote):
+ removeBanknotes(): List<Banknote>
+ closeHardware():
+ refillInk(amount:int):
+ refillPaper(amount:int):
+ register(MaintenanceManagerListener):
+ deregister(MaintenanceManagerListener): boolean
+ deregisterAll():

- listeners

« interface » MaintenanceManagerListener

- banknoteStorage

BanknoteStorageUnit

- coinStorage

CoinStorageUnit

- banknoteDispensers

Map<BigDecimal, IBanknoteDispenser>

- coinDispensers

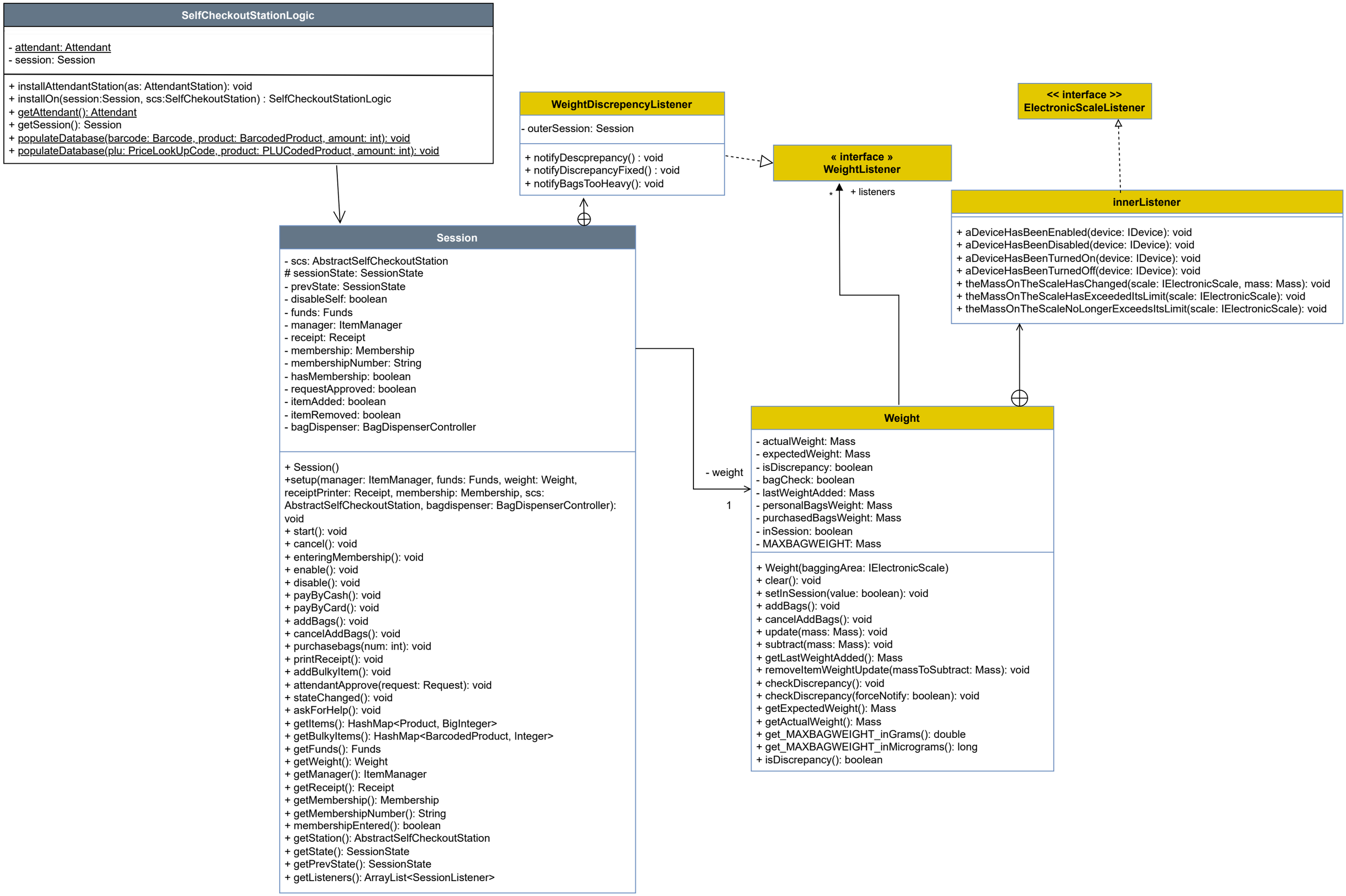
Map<BigDecimal, ICoinDispenser>

- receiptPrinter

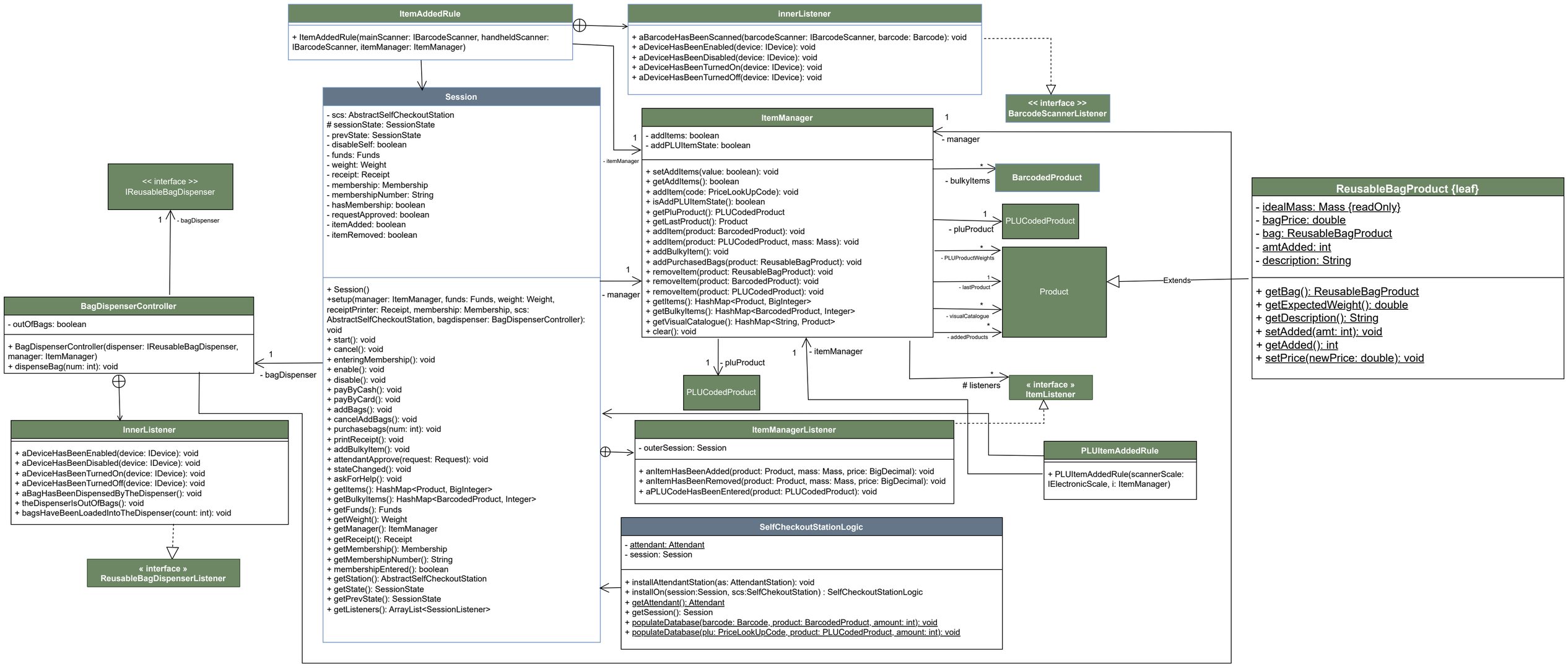
IReceiptPrinter

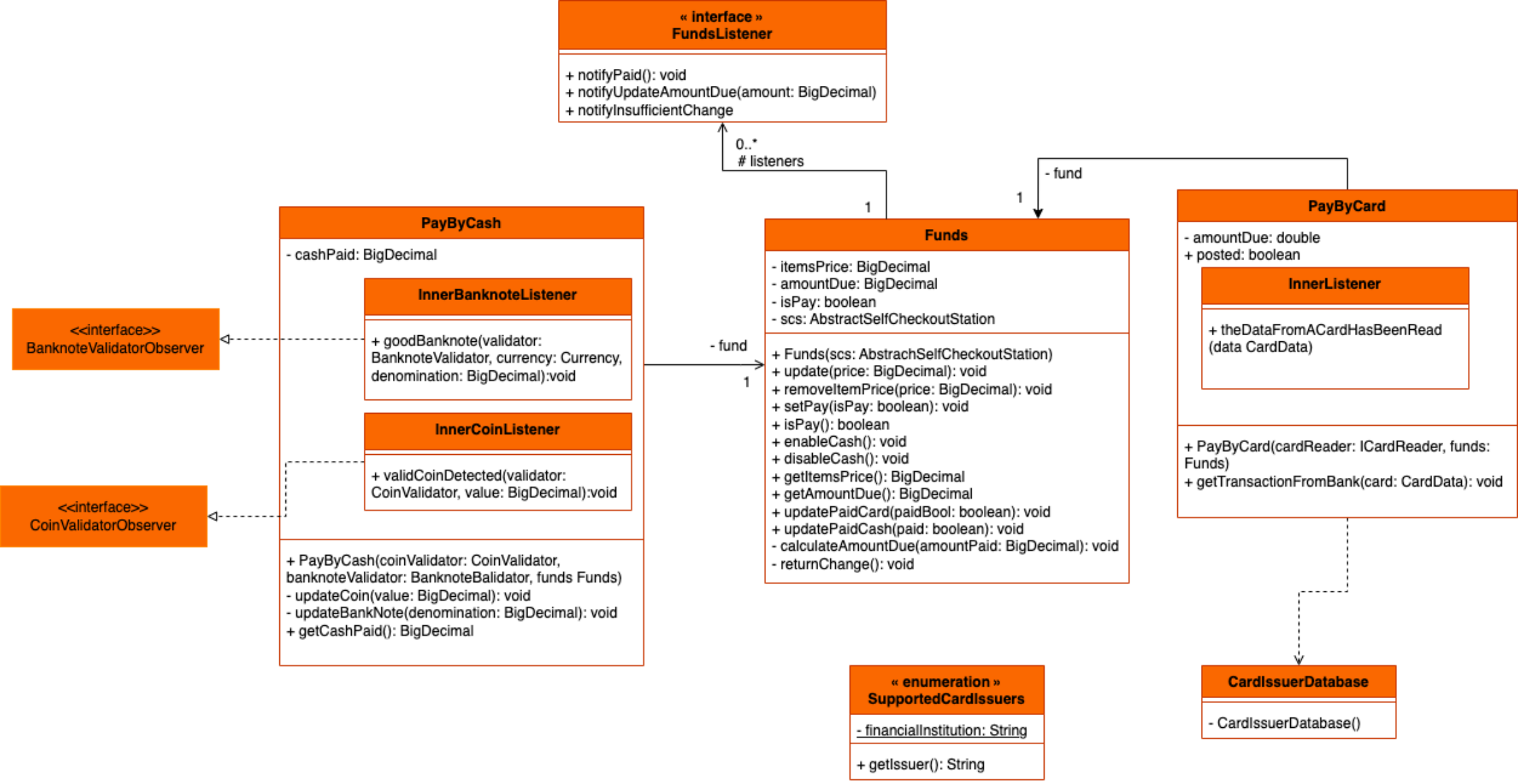
Attendant Package Overview

weight Package Diagram

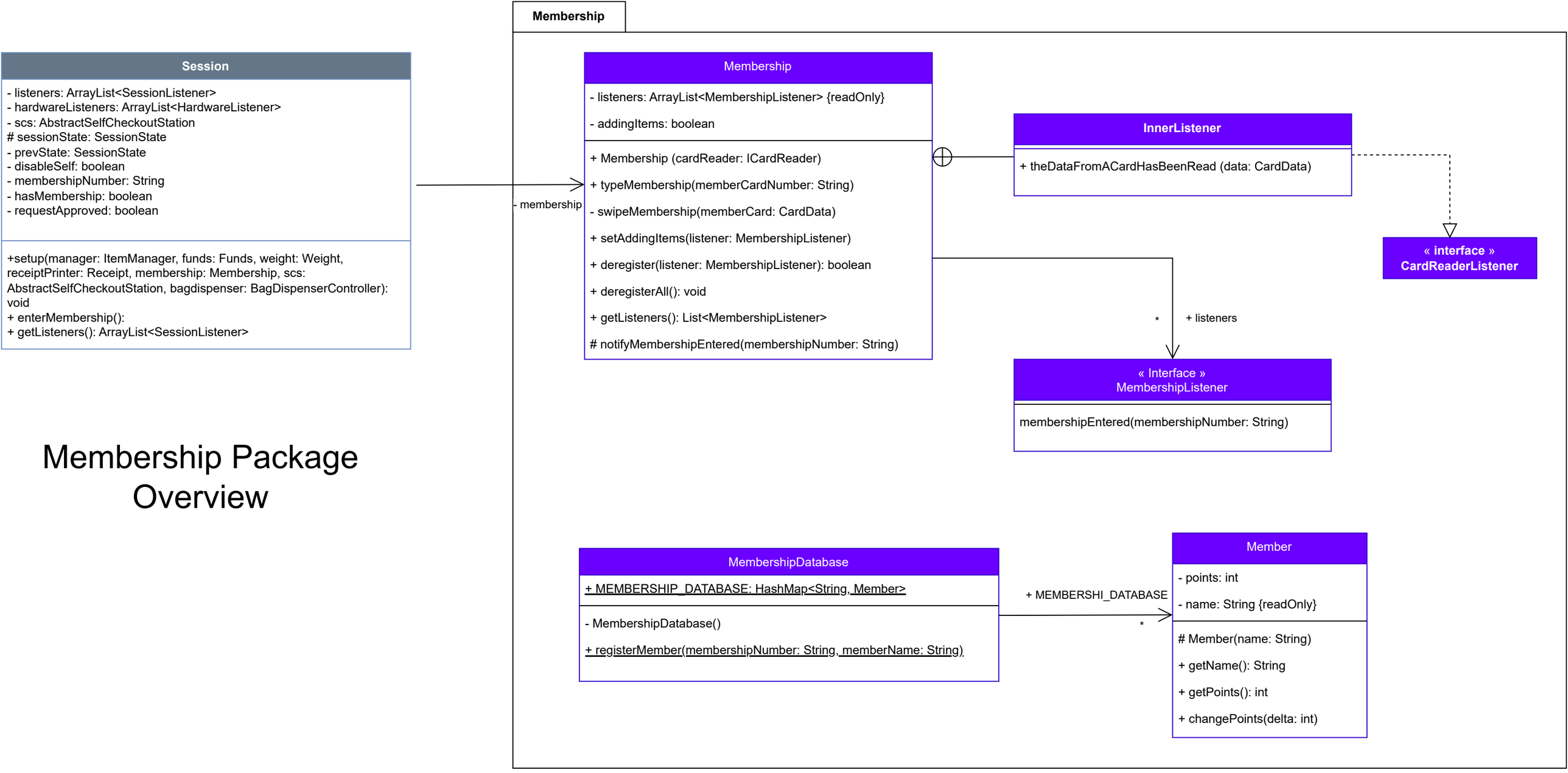


items Package Diagram




« interface »
FundsListener
+ notifyPaid(): void
+ notifyUpdateAmountDue(amount: BigDecimal)
+ notifyInsufficientChange
0..*
listeners
1
- fund
1
PayByCash
- cashPaid: BigDecimal
InnerBanknoteListener
+ goodBanknote(validator: BanknoteValidator, currency: Currency, denomination: BigDecimal):void
InnerCoinListener
+ validCoinDetected(validator: CoinValidator, value: BigDecimal):void
+ PayByCash(coinValidator: CoinValidator, banknoteValidator: BanknoteBalidator, funds Funds)
- updateCoin(value: BigDecimal): void
- updateBankNote(denomination: BigDecimal): void
+ getCashPaid(): BigDecimal
<<interface>>
BanknoteValidatorObserver
<<interface>>
CoinValidatorObserver
- fund
1
Funds
- itemsPrice: BigDecimal
- amountDue: BigDecimal
- isPay: boolean
- scs: AbstractSelfCheckoutStation
+ Funds(scs: AbstrachSelfCheckoutStation)
+ update(price: BigDecimal): void
+ removeItemPrice(price: BigDecimal): void
+ setPay(isPay: boolean): void
+ isPay(): boolean
+ enableCash(): void
+ disableCash(): void
+ getItemsPrice(): BigDecimal
+ getAmountDue(): BigDecimal
+ updatePaidCard(paidBool: boolean): void
+ updatePaidCash(paid: boolean): void
- calculateAmountDue(amountPaid: BigDecimal): void
- returnChange(): void
PayByCard
- amountDue: double
+ posted: boolean
InnerListener
+ theDataFromACardHasBeenRead (data CardData)
+ PayByCard(cardReader: ICardReader, funds: Funds)
+ getTransactionFromBank(card: CardData): void
« enumeration »
SupportedCardIssuers
- financialInstitution: String
+ getIssuer(): String
CardIssuerDatabase
- CardIssuerDatabase()

Membership Package Overview



GUI Package Overview

com.thelocalmarketplace.GUI

com.thelocalmarketplace.GUI.session

+ CartProduct
- product: Product
- session: Session
- quantity: int
- isPLU: boolean
+ CartProduct(product: Product, session: Session, mass: Mass)
+ addProduct(): boolean
+ removeProduct(): boolean
+ updateMass(inGrams: BigDecimal): void

+ ProductPanel
- serialVersionUID: long
- product: Product
- session: Session
- addButton: PlainButton
+ ProductPanel(product: Product, session: Session)
# paintComponent(g Graphics): void
+ getAddButon(): JButton

+JPanel

+ ActionListener
+ actionPerformed(e: ActionEvent): void

*panel

+ AddedProducts
- serialVersionUID
- productList: HashMap<Product, CartProduct>
- session: Session
+ AddedProducts(session: Session)
+ removeProduct(product: Product, mass: Mass): void
+ getProducts(): HashMap<Product, CartProduct>
+ contains(product: Product)
+ amount(): int

+ SearchCatalogue
+ SearchCatalogue(session: Session)
+ getHashMapForButtons(): HashMap<Product, Pr

+SessionListener
+ itemAdded(session: Session, product: Product, ofProduct: Mass, currentExpectedWeight: Mass, currentExpectedPrice: BigDecimal): void
+ itemRemoved(session: Session, product: Product, ofProduct: Mass, currentExpectedWeight: Mass, currentExpectedPrice: BigDecimal): void
+ addItemToScaleDiscrepancy(session: Session): void
+ removeItemFromScaleDiscrepancy(session:Session): void
+ discrepancy(session: Session, message: String): void
+ discrepancyResolved(session: Session): void
+ pricePaidUpdated(session: Session, amountDue, BigDecimal): void
+ getRequest(session: Session, request Requests): void
+ sessionAboutToStart(session: Session): void
+ sessionEnded(session: Session): void
+ pluCodeEntered(product: PLUCodedProduct): void
+ sessionStateChanged(): void

+ catalogue
+ cartItemsPanel
+ InnerListener
+ ButtonListener

+ SoftwareGUI
+ SoftwareGUI(session: Session)
+ start(): JPanel
+ end(): JPanel
+ mainPanel(): JPanel
+ displayStart(): void
+ displayMain(): void
+ displayEnd(): void
+ displayDisabled(): void
+ update(price: double, mass: double): void
+ hide(): void
+ unhide(): void

- addBagsScreen
+ paymentScreen
- pluNumPad

+ PaymentPopup
+ public PaymentPopup(session: Session)
+ popUp(): void
+ hide(): void
+ isVisibile(): boolean
+ getCashButton(): JButton
+ getCardButton(): JButton
+ getMembershipButton(): JButton
+ getMembershipPad(): MembershipPad

+ PLUNumPad
- session: Session
+ MembershipNumPad(session: Session)
+ getValidMembership(): boolean
+ isVisible(): boolean

- membershipPad

+ MembershipNumPad
- session: Session
- plu: String
+ MembershipNumPad(session: Session)
+ getValidMembership(): boolean
+ isVisible(): boolean

+ NumberOfBags
- session: Session
+ NumberOfBags(session: Session)
+ returnNum(): void
+ updateDisplay(): void
+ isVisible(): boolean
+ getNumber(): String
+ getDelete(): PlainButton
+ getDone(): PlainButtoin

+ AddBagsPopup
+ AddBagsPopup(session: Session)
+ getAddPersonalBagButton(): JButton
+ getAddStoreBagButton(): JButton
+ getCancelButton(): JButton
+ getNumOfBagsScreen(): NumberOfBags
+ isVisible(): boolean

- numOfBagsScreen

+ Simulation
- scs: AbstractSelfCheckoutStation
- as: AttendantStation
- attendant: Attendant
- session: Session
- receipt: Receipt
- predictor: IssuePredictor
- manager: MaintenanceManager
+ Simulation()
+ setupLogic(): void
+ setupData(): void
+ setupLoading(): void
+ unhide(): void

- softwareGUI
- hardwareGUI
- attendantGUI

+ Main
+ main(String[] args): void

com.thelocalmarketplace.GUI.customComponents

+ Colors
+ GradientPanel
+ CustomBarUI
+ PlainButton
+ RoundPanel

com.thelocalmarketplace.GUI.attendant

+ AttendantGUI
+ attendant: Attendant
+ manager: MaintenanceManager
+ predictor: IssuePredictor
+ sessions: HashMap <Session, Requests>
+ textSearch: TextSearchController
+ asScreen: ITouchScreen
- populateSessions(): void
+ hide(): void
+ unhide(): void

* panel

+ StationPanel
+ session: Session
- attendant: Attendant
- issues: HashMap <String, Boolean>
+ enabled: Boolean
- populateMessages(): void
+ issuesPresent(): Boolean
+ getIssues(): HashMap <String, Boolean>
- updateIssues(HashMap <String, Boolean>): void

+ ProductItem
+ ProductItem(Product, Session, Attendant)

+ AttendantCatalogue
- inventory: Map <String, Product>
+ AttendantCatalogue(Session, Attendant)
+ remove(): void

- InnerPredictionListener
- InnerResolutionListener

- power
- addBySearch
- approveRequest

com.thelocalmarketplace.software.attendant

- IssuePredictorListener
+ notifyPredictLowInk(Session)
+ notifyPredictLowPaper(Session)
+ notifyPredictCoinsFull(Session)
+ notifyPredictBanknotesFull(Session)
+ notifyPredictLowCoins(Session)
+ notifyPredictLowBanknotes(Session)
+ notifyNoIssues(Session)

- MaintenanceManagerListener
+ notifyInkAdded(Session)
+ notifyPaperAdded(Session)
+ notifyCoinAdded(Session)
+ notifyBanknoteAdded(Session)
+ notifyCoinRemoved(Session, CoinStorageUnit)
+ notifyBanknoteRemoved(Session, BanknoteStorageUnit)
+ notifyHardwareOpened(Session)
+ notifyHardwareClosed(Session)

com.thelocalmarketplace.GUI.hardware

+ HardwareGUI
- supervisor: AttendantStation
- scs: AbstractSelfCheckoutStation
+ HardwareGUI(scs: AbstractSelfCheckoutStation, supervisor: AttendantStation)
+ populateItems(): void
+ introPanel(): JPanel
+ itemPanel(title: String, list: JList<ItemObject< list): JPanel
+ addData(data: ItemObject, list: JList<ItemObject>): void
+ removeData(data: ItemObject, list: JList<ItemObject>): void
+ mainScan(): void
+ handScan(): void
+ getSupervisor(): AttendantStation
+ getStation(): AbstractSelfCheckoutStation
+ setVisibility(visibility: boolean): void

*panel

+ CashPanel
- serialVersionUID: long
- scs: AbstractSelfCheckoutStation
+ FiveBillBtn: JButton
+ button_one_coin: JButton
- Five_Cent_Coin: Coin
- Ten_Cent_Coin: Coin
- TwentyFive_Cent_Coin: Coin
- One_Dollar_Coin: Coin
- Two_Dollar_Coin: Coin
- Non_Coin: Coin
- Five_Dollar_Bill: Banknote
- Ten_Dollar_Bill: Banknote
- Twenty_Dolalr_Bill: Banknote
- Fifty_Dollar_Bill: Banknote
- Hundred_Dollar_Bill: Banknote
- Non_Bill: Banknote
+ CashPanel(scs: AbstractSelfCheckoutStation)
+ setupCashData(): void

+JPanel

*panel

+ CardPanel
- serialVersionUID: long
- scs: AbstractSelfCheckoutStation
- cardSelected: Card
- cardSelectedString: String
- debitCard: Card
- creditCard: Card
- invalidCard: Card
- membershipCard: Card
- infoLabel: JLabel
- infoPanel: JPanel
- pane12: JPanel
- inserted: boolean
- swipeButton: JButton
- insertButton: JButton
- tapButton: JButton
+ CardPanel(scs: AbstractSelfCheckoutStation)
# updateSelection(): void
+ setupCardData(): void

+ CardListener

+ ActionListener
+ actionPerformed(e: ActionEvent): void

+ ButtonPanel
- bags: ReusableBag[ ]
- numBags: int
- long serialVersionUID
+ ButtonPanel(gui: HardwareGUI)

- gui

- lastObject
lastItem

+ ListTransferHandler
- index: int
- addIndex: int
- addCount: int
+ canImport(info: TransferHandler.TransferSupport): boolean
# createTransferable(c: JComponent): Transferrable
+ getSourceActions(c: JComponent): int
+ importData(info: TransferHandler:TransferSupport): boolean
# exportDone(c: JComponent, data: Transferable, action: int): void

+ TransferHandler

+ ItemObject
- name: String
- item: Item
+ flavor: DataFlavor
+ ItemObject(item: BarcodedItem, name: String)
+ ItemObject(item: PLUCodedItem, name: String)
+ getBarcode(): Barcode
+ getPLUCode(): PriceLookUpCode
+ getMass(): Mass
+ getItem(): Item
+ toString(): String
+ createTransferable(obj: Object): Transferrable
+ isDataFlavorSupported(oflavor: DataFlavor)
+ getDataFlavor(): DataFlavor
+ getTransferData(oflavor: DataFlavor): Object