SENG 300 GA Iteration 3

Dany Saaid Bonilla - 30050992
Grace Heemeryck - 30068546
Matt Jarrams - 30061191
Nathan Lum - 30064695
Mathew Luong - 30068650
Zeyad Omran - 30096692
Kylie Sicat - 30062029
Mackenzie Dalton- 30063061
Joseph Lam - 30065199
Kirk Elumir - 30073334
Sarthak Sharan - 30092835

Explanation:

Structural Diagrams:

- The class SelfCheckoutSoftware contains methods/member variables with all the ListenerStubs, Databases and Object classes. This is denoted with the aggregation arrows.
- ListenerStubs: all implement their respective Listeners (from Hardware) this is denoted with the implementation dotted arrows.
- All the Listeners extend AbstractDeviceListener (from Hardware) this is denoted with the inheritance arrows.

GUI Structural Diagram

- All the GUI classes contain methods/member variables (i.e the control variable). This is denoted with the aggregation arrows.
- All the GUI Panels extend Jpanel (from Swing GUI classes) this is denoted with the inheritance arrows.

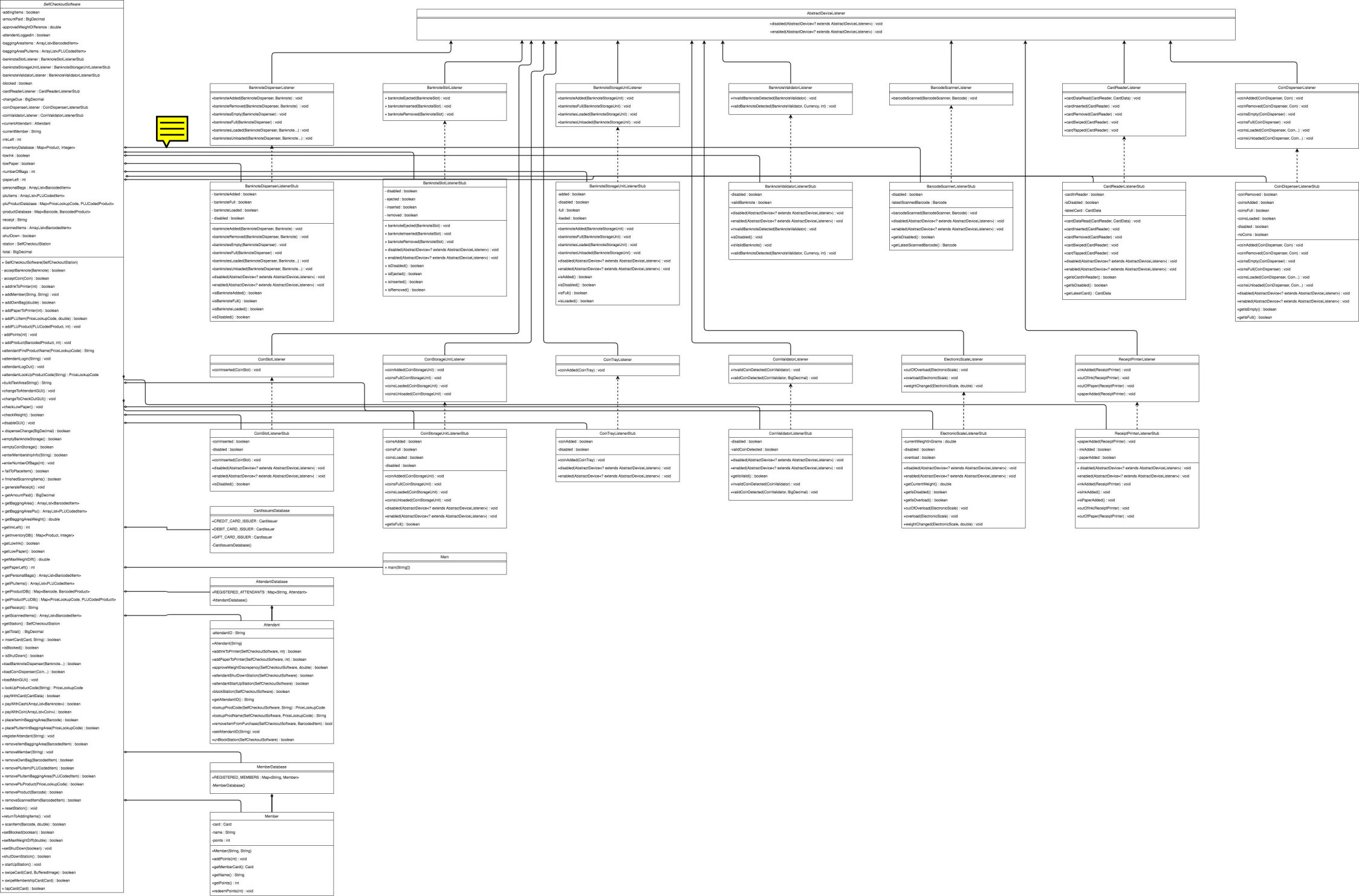
Sequence Diagrams

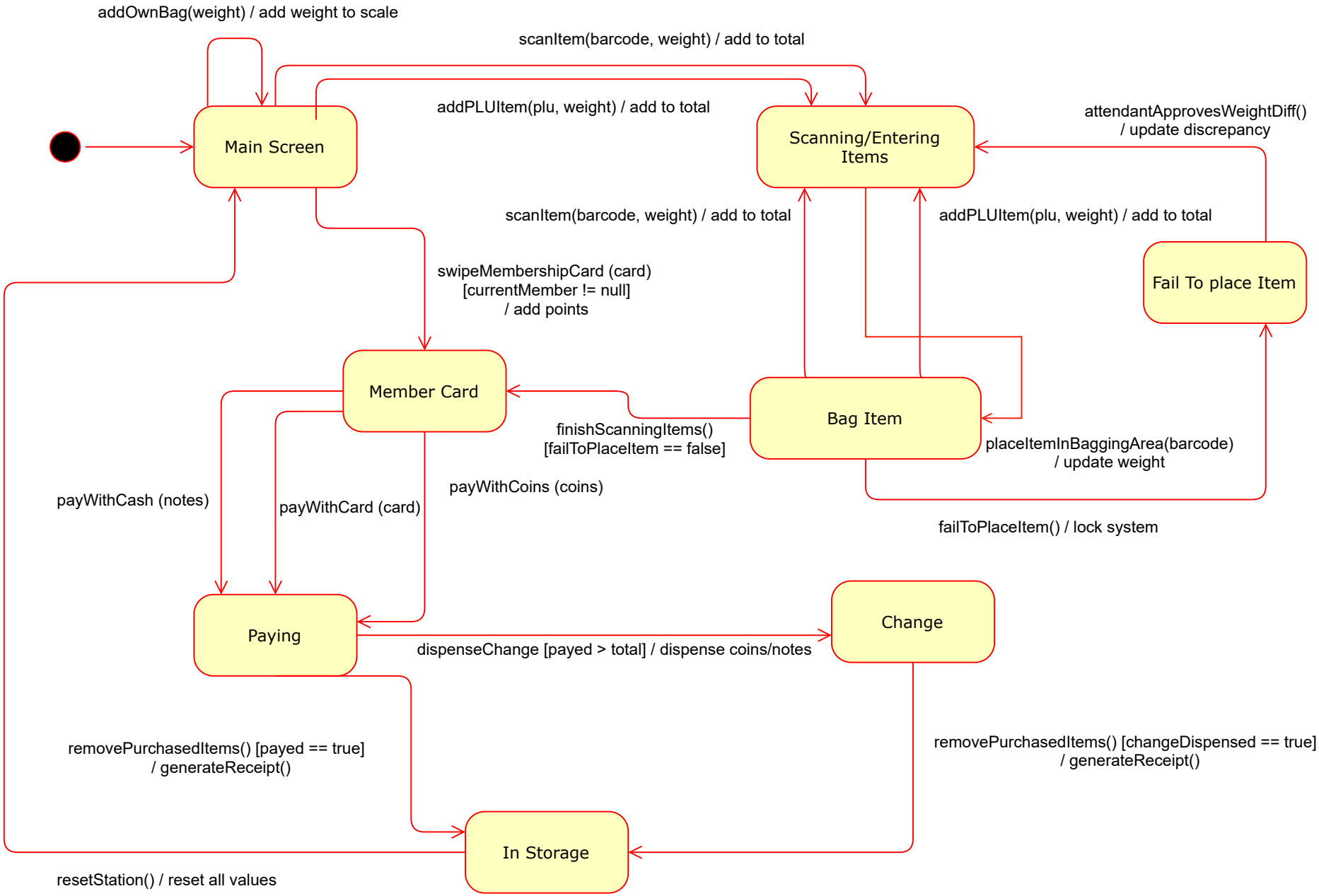
- In the scanItem and PlaceItemInBaggingArea diagrams, a BarcodedItem object is created/instantiated, this is denoted by the <<create>> arrow
- Roles are used to model generic objects, e.g. SelfCheckoutSoftware is a role in the diagrams, roles that have no prefix before the : are anonymous (s:SelfCheckoutSoftware means s is the name of an instance of SelfCheckoutSoftware)
- The role of type map<Barcode, BarcodedProduct> is the productdatabase, and map<Product, Integer> is the inventory database (these are fields in the software class)
- Alt frames are used frequently to model if/else constructs, where the if condition is contained within a guard, denoted by square brackets e.g. [condition]
- In PayWithCard, ScanItem, and PlaceItemInBaggingArea, return statements connected to the SelfCheckoutSoftware lifeline denote the method returning true/false

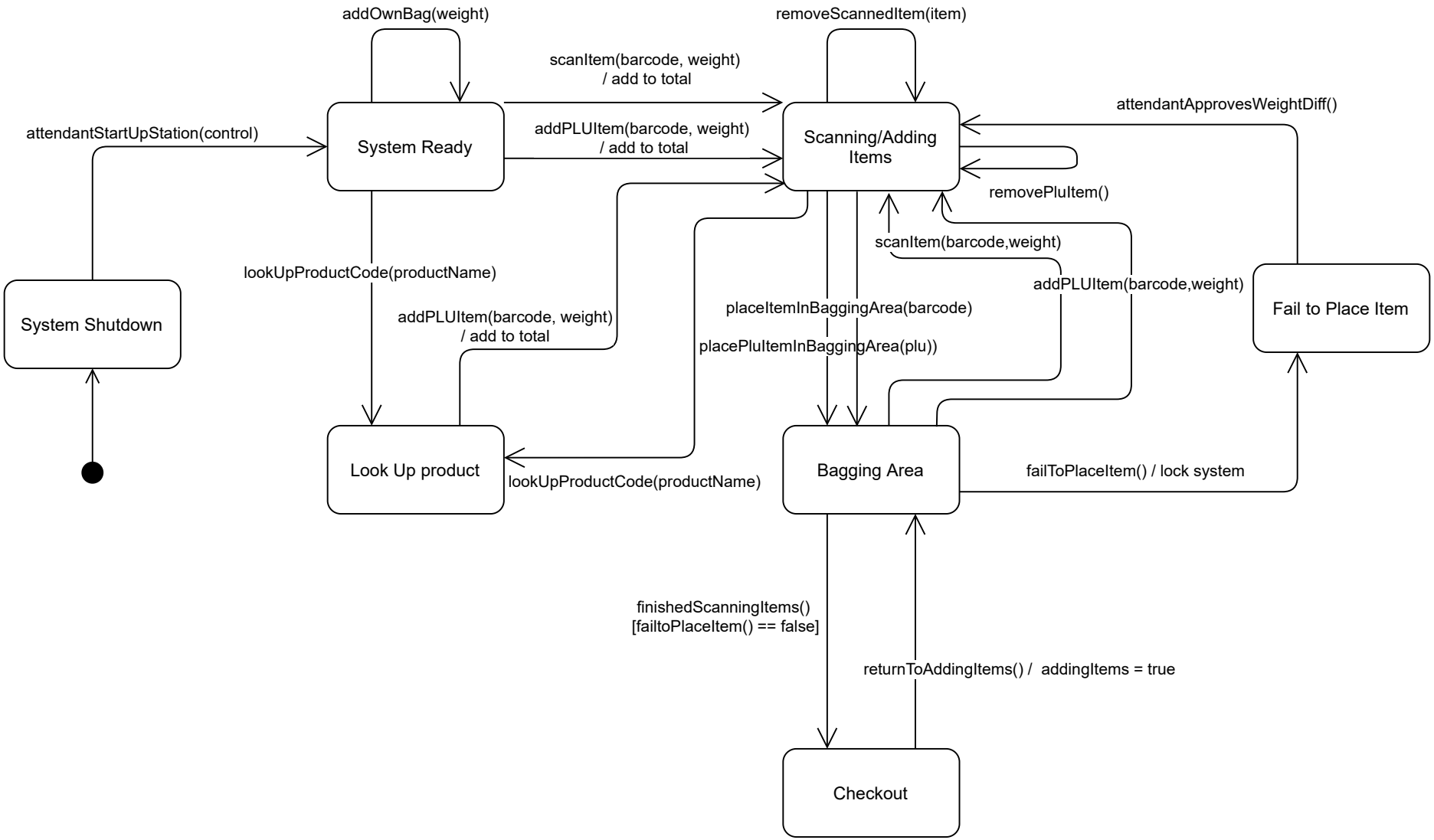
State Diagrams

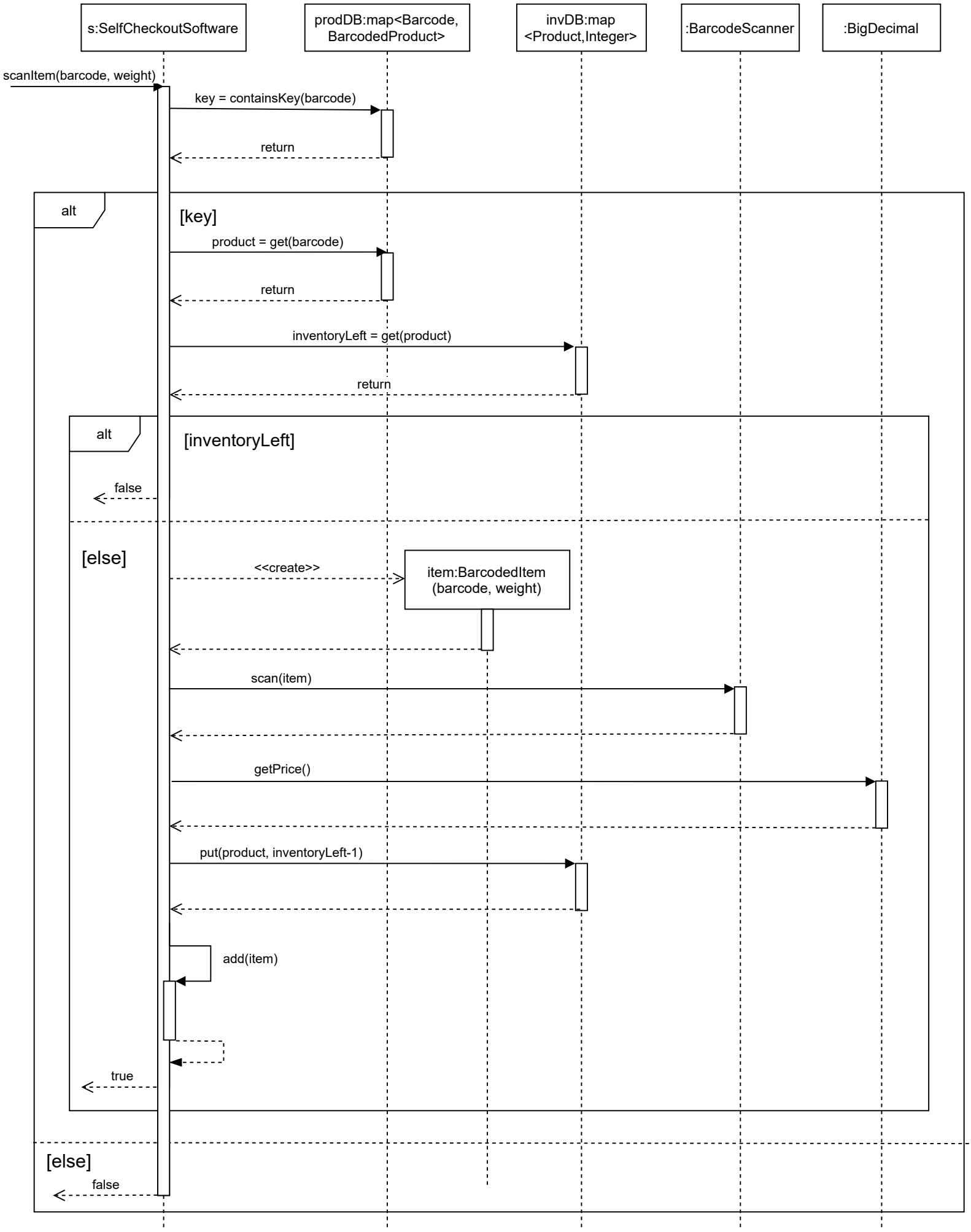
- The state diagrams try to show the most important parts of our system and the states associated with those processes. We have a high level one to show how the whole system reacts in different states and how it gets to them. But we also make sure to show some of the lower level processes since they have their own states they go through that impact the higher level of the system.
- This is shown through a checking items, bagging area and a paying state diagram (the major functionalities, we thought it was best to abstract other smaller state diagrams)

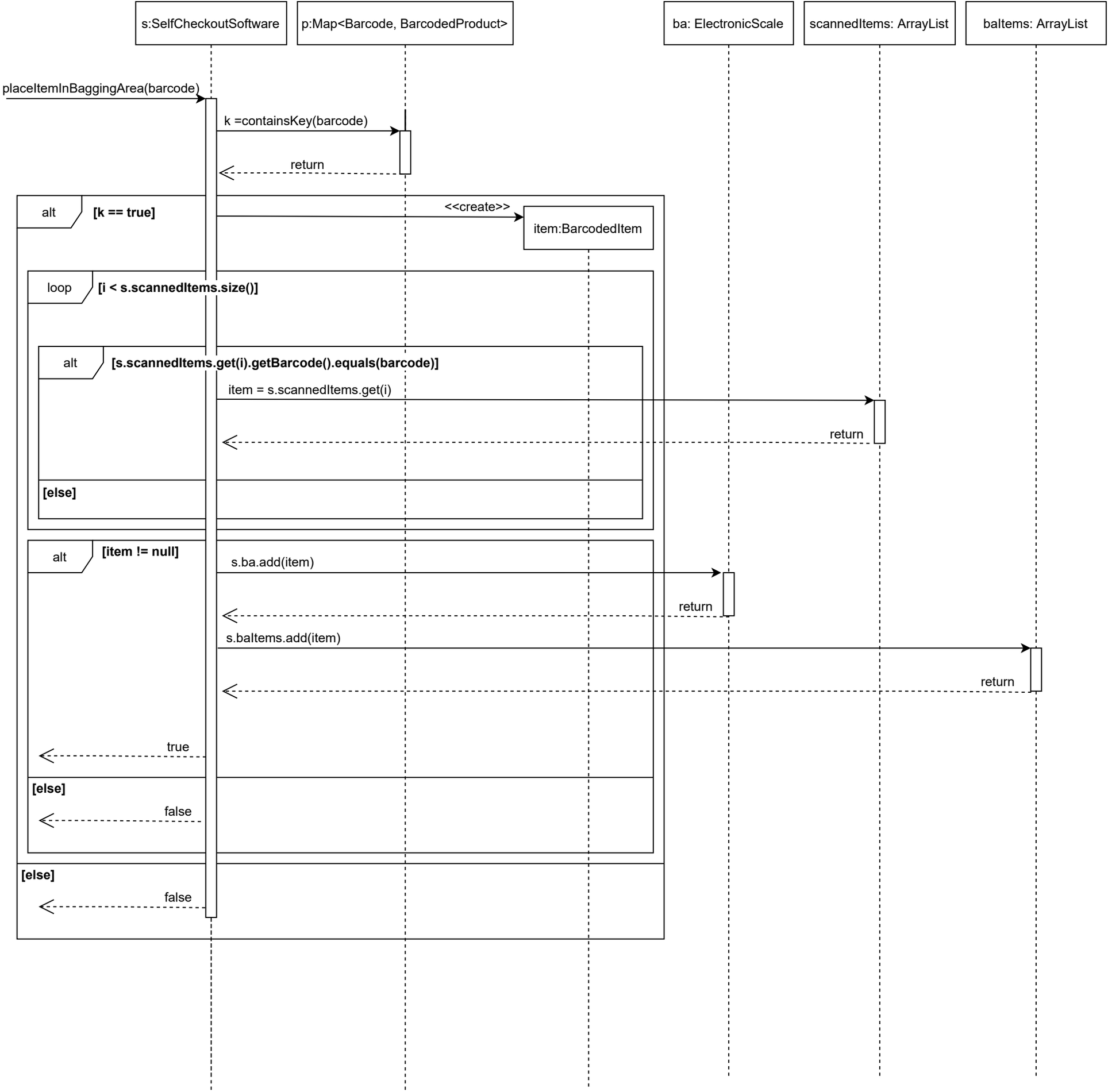
Although we do understand our design could have been better (we could have separated some more classes to avoid high coupling) at the time we thought this was the best way to proceed at the time especially with large amounts of changes and not knowing the full scope of the project beforehand. This is what made our design good as it was easy to adapt to changes. However, due to the high coupling these changes did create other problems in the system as well.

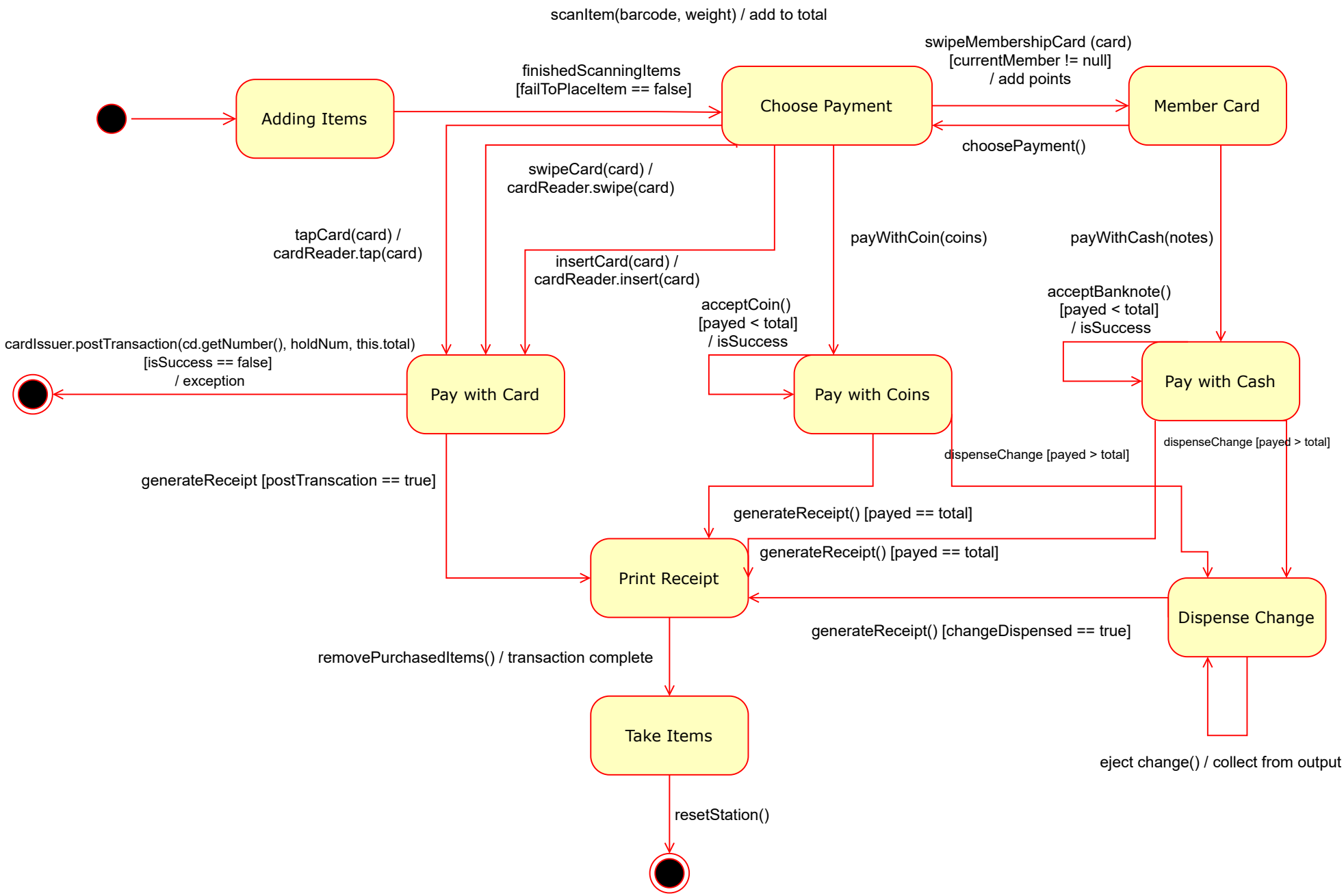

SelfCheckoutSoftware
-addingItems : boolean
-amountPaid : BigDecimal
-approvedWeightDifference : double
-attendentLoggedIn : boolean
-baggingAreaItems : ArrayList<BarcodedItem>
-baggingAreaPluItems : ArrayList<PLUCodedItem>
-banknoteSlotListener : BanknoteSlotListenerStub
-banknoteStorageUnitListener : BanknoteStorageUnitListenerStub
-banknoteValidatorListener : BanknoteValidatorListenerStub
-blocked : boolean
-cardReaderListener : CardReaderListenerStub
-changeDue : BigDecimal
-coinDispenserListener : CoinDispenserListenerStub
-coinValidatorListener : CoinValidatorListenerStub
+currentAttendant : Attendant
-currentMember : String
-inkLeft : int
-inventoryDatabase : Map<Product, Integer>
-lowInk : boolean
-lowPaper : boolean
-numberOfBags : int
-paperLeft : int
-personalBags : ArrayList<BarcodedItem>
-pluItems : ArrayList<PLUCodedItem>
-pluProductDatabase : Map<PriceLookupCode, PLUCodedProduct>
-productDatabase : Map<Barcode, BarcodedProduct>
-receipt : String
-scannedItems : ArrayList<BarcodedItem>
-shutDown : boolean
-station : SelfCheckoutStation
-total : BigDecimal
+ SelfCheckoutSoftware(SelfCheckoutStation)
- accept(BanknoteBanknote) : boolean
- acceptCoin(Coin) : boolean
+ addInkToPrinter(int) : boolean
+ addMember(String, String) : void
+ addOwnBag(double) : boolean
+ addPaperToPrinter(int) : boolean
+ addPLUItem(PriceLookupCode, double) : boolean
+ addPLUProduct(PLUCodedProduct, int) : void
- addPoints(int) : void
+ addProduct(BarcodedProduct, int) : void
+attendantFindProductName(PriceLookupCode) : String
+attendantLogin(String) : void
+attendantLogOut() : void
+attendantLookUpProductCode(String) : PriceLookupCode
+buildTextAreaString() : String
+changeToAttendantGUI() : void
+changeToCheckOutGUI() : void
+checkLowPaper() : void
+checkWeight() : boolean
+disableGUI() : void
+ discreteChange(BigDecimal) : boolean
+emptyBanknoteStorage() : boolean
+emptyCoinStorage() : boolean
+enterMembershipInfo(String) : boolean
+enterNumberOfBags(int) : void
+ faiITsPlaceItem() : boolean
+ finishedScanningItems() : boolean
+ generateReceipt() : void
+ getAmountPaid() : BigDecimal
+getBaggingArea() : ArrayList<BarcodedItem>
+ getBaggingAreaPlu() : ArrayList<PLUCodedItem>
+getBaggingAreaWeight() : double
+getInkLeft() : int
+ getInventoryDB() : Map<Product, Integer>
+getLowInk() : boolean
+getLowPaper() : boolean
+getMaxWeightDiff() : double
+getPaperLeft() : int
+ getPersonalBags() : ArrayList<BarcodedItem>
+ getPluItems() : ArrayList<PLUCodedItem>
+ getProductDB() : Map<Barcode, BarcodedProduct>
+ getProductPLUDB() : Map<PriceLookupCode, PLUCodedProduct>
+ getReceipt() : String
+ getScannedItems() : ArrayList<BarcodedItem>
+getStation() : SelfCheckoutStation
+ getTotal() : BigDecimal
+ insertCard(Card, String) : boolean
+isBlocked() : boolean
+ isShutDown() : boolean
+loadBanknoteDispenser(Banknote...) : boolean
+loadCoinDispenser(Coin...) : boolean
+loadMainGUI() : void
+ lookUpProductCode(String) : PriceLookupCode
- payWithCard(CardData) : boolean
+ payWithCash(ArrayList<Banknote>) : boolean
+ payWithCoin(ArrayList<Coin>) : boolean
+ placeItemInBaggingArea(Barcode) : boolean
+ placePluItemInBaggingArea(PriceLookupCode) : boolean
+registerAttendent(String) : void
+ removeItemBaggingArea(BarcodedItem) : boolean
+ removeMember(String) : void
+ removeOwnBag(BarcodedItem) : boolean
+ removePluItem(PLUCodedItem) : boolean
+ removePluItemBaggingArea(PLUCodedItem) : boolean
+ removePluProduct(PriceLookupCode) : boolean
+ removeProduct(Barcode) : boolean
+ removeScannedItem(BarcodedItem) : boolean
+ resetStation() : void
+returnToAddingItems() : void
+ scanItem(Barcode, double) : boolean
+setBlocked(boolean) : boolean
+setMaxWeightDiff(double) : boolean
+setShutDown(boolean) : void
+shutDownStation() : boolean
+ startUpStation() : void
+ swipeCard(Card, BufferedImage) : boolean
+ swipeMembershipCard(Card) : boolean
+ tapCard(Card) : boolean
AbstractDeviceListener
+disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
BanknoteDispenserListener
+banknoteAdded(BanknoteDispenser, Banknote) : void
+banknoteRemoved(BanknoteDispenser, Banknote) : void
+banknotesEmpty(BanknoteDispenser) : void
+banknotesFull(BanknoteDispenser) : void
+banknotesLoaded(BanknoteDispenser, Banknote...) : void
+banknotesUnloaded(BanknoteDispenser, Banknote...) : void
BanknoteSlotListener
+ banknoteEjected(BanknoteSlot) : void
+ banknoteInserted(BanknoteSlot) : void
+ banknoteRemoved(BanknoteSlot) : void
BanknoteStorageUnitListener
+banknoteAdded(BanknoteStorageUnit) : void
+banknotesFull(BanknoteStorageUnit) : void
+banknotesLoaded(BanknoteStorageUnit) : void
+banknotesUnloaded(BanknoteStorageUnit) : void
BanknoteValidatorListener
+invalidBanknoteDetected(BanknoteValidator) : void
+validBanknoteDetected(BanknoteValidator, Currency, int) : void
BarcodeScannerListener
+barcodeScanned(BarcodeScanner, Barcode) : void
CardReaderListener
+cardDataRead(CardReader, CardData) : void
+cardInserted(CardReader) : void
+cardRemoved(CardReader) : void
+cardSwiped(CardReader) : void
+cardTapped(CardReader) : void
CoinDispenserListener
+coinAdded(CoinDispenser, Coin) : void
+coinRemoved(CoinDispenser, Coin) : void
+coinsEmpty(CoinDispenser) : void
+coinsFull(CoinDispenser) : void
+coinsLoaded(CoinDispenser, Coin...) : void
+coinsUnloaded(CoinDispenser, Coin...) : void
BanknoteDispenserListenerStub
- banknoteAdded : boolean
- banknoteFull : boolean
- banknoteLoaded : boolean
- disabled : boolean
+banknoteAdded(BanknoteDispenser, Banknote) : void
+banknoteRemoved(BanknoteDispenser, Banknote) : void
+banknotesEmpty(BanknoteDispenser) : void
+banknotesFull(BanknoteDispenser) : void
+banknotesLoaded(BanknoteDispenser, Banknote...) : void
+banknotesUnloaded(BanknoteDispenser, Banknote...) : void
+disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+isBanknoteAdded() : boolean
+isBanknoteFull() : boolean
+isBanknoteLoaded() : boolean
+isDisabled() : boolean
BanknoteSlotListenerStub
- disabled : boolean
- ejected : boolean
- inserted : boolean
- removed : boolean
+ banknoteEjected(BanknoteSlot) : void
+ banknoteInserted(BanknoteSlot) : void
+ banknoteRemoved(BanknoteSlot) : void
+ disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+ enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+ isDisabled() : boolean
+ isEjected() : boolean
+ isInserted() : boolean
+ isRemoved() : boolean
BanknoteStorageUnitListenerStub
-added : boolean
-disabled : boolean
-full : boolean
-loaded : boolean
+banknoteAdded(BanknoteStorageUnit) : void
+banknotesFull(BanknoteStorageUnit) : void
+banknotesLoaded(BanknoteStorageUnit) : void
+banknotesUnloaded(BanknoteStorageUnit) : void
+disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+isAdded() : boolean
+isDisabled() : boolean
+isFull() : boolean
+isLoaded() : boolean
BanknoteValidatorListenerStub
-disabled : boolean
-validBanknote : boolean
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+invalidBanknoteDetected(BanknoteValidator) : void
+isDisabled() : void
+isValidBanknote() : void
+validBanknoteDetected(BanknoteValidator, Currency, int) : void
BarcodeScannerListenerStub
-disabled : boolean
-latestScannedBarcode : Barcode
+barcodeScanned(BarcodeScanner, Barcode) : void
+disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+getIsDisabled() : boolean
+getLatestScannedBarcode() : Barcode
CardReaderListenerStub
-cardInReader : boolean
-isDisabled : boolean
-lateerCard : CardData
+cardDataRead(CardReader, CardData) : void
+cardInserted(CardReader) : void
+cardRemoved(CardReader) : void
+cardSwiped(CardReader) : void
+cardTapped(CardReader) : void
+disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+getIsCardInReader() : boolean
+getIsDisabled() : boolean
+getLatestCard() : CardData
CoinDispenserListenerStub
-coinRemoved : boolean
-coinsAdded : boolean
-coinsFull : boolean
-coinsLoaded : boolean
-disabled : boolean
-noCoins : boolean
+coinAdded(CoinDispenser, Coin) : void
+coinRemoved(CoinDispenser, Coin) : void
+coinsEmpty(CoinDispenser) : void
+coinsFull(CoinDispenser) : void
+coinsLoaded(CoinDispenser, Coin...) : void
+coinsUnloaded(CoinDispenser, Coin...) : void
+disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+getIsEmpty() : boolean
+getIsFull() : boolean
CoinSlotListener
+coinInserted(CoinSlot) : void
CoinStorageUnitListener
+coinAdded(CoinStorageUnit) : void
+coinsFull(CoinStorageUnit) : void
+coinsLoaded(CoinStorageUnit) : void
+coinsUnloaded(CoinStorageUnit) : void
CoinTrayListener
+coinAdded(CoinTray) : void
CoinValidatorListener
+invalidCoinDetected(CoinValidator) : void
+validCoinDetected(CoinValidator, BigDecimal) : void
ElectronicScaleListener
+outOfOverload(ElectronicScale) : void
+overload(ElectronicScale) : void
+weightChanged(ElectronicScale, double) : void
ReceiptPrinterListener
+inkAdded(ReceiptPrinter) : void
+outOfInk(ReceiptPrinter) : void
+outOfPaper(ReceiptPrinter) : void
+paperAdded(ReceiptPrinter) : void
CoinSlotListenerStub
-coinInserted : boolean
-disabled : boolean
+coinInserted(CoinSlot) : void
+disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+isDisabled() : boolean
CoinStorageUnitListenerStub
-coinsAdded : boolean
-coinsFull : boolean
-coinsLoaded : boolean
-disabled : boolean
+coinAdded(CoinStorageUnit) : void
+coinsFull(CoinStorageUnit) : void
+coinsLoaded(CoinStorageUnit) : void
+coinsUnloaded(CoinStorageUnit) : void
+disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+getIsFull() : boolean
CoinTrayListenerStub
-coinAdded : boolean
-disabled : boolean
+coinAdded(CoinTray) : void
+disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
CoinValidatorListenerStub
-disabled : boolean
-validCoinDetected : boolean
+disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+getIsValid() : boolean
+invalidCoinDetected(CoinValidator) : void
+validCoinDetected(CoinValidator, BigDecimal) : void
ElectronicScaleListenerStub
-currentWeightInGrams : double
-disabled : boolean
-overload : boolean
+disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+getCurrentWeight() : double
+getIsDisabled() : boolean
+getIsOverload() : boolean
+outOfOverload(ElectronicScale) : void
+overload(ElectronicScale) : void
+weightChanged(ElectronicScale, double) : void
ReceiptPrinterListenerStub
+paperAdded(ReceiptPrinter) : void
- inkAdded : boolean
- paperAdded : boolean
+ disabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+enabled(AbstractDevice<? extends AbstractDeviceListener>) : void
+inkAdded(ReceiptPrinter) : void
+isInkAdded() : void
+isPaperAdded() : void
+outOfInk(ReceiptPrinter) : void
+outOfPaper(ReceiptPrinter) : void
CardIssuersDatabase
+CREDIT_CARD_ISSUER : CardIssuer
+DEBIT_CARD_ISSUER : CardIssuer
+GIFT_CARD_ISSUER : CardIssuer
-CardIssuersDatabase()
Main
+ main(String[])
AttendantDatabase
+REGISTERED_ATTENDANTS : Map<String, Attendant>
-AttendantDatabase()
Attendant
-attendantID : String
+Attendant(String)
+addInkToPrinter(SelfCheckoutSoftware, int) : boolean
+addPaperToPrinter(SelfCheckoutSoftware, int) : boolean
+approveWeightDiscrepency(SelfCheckoutSoftware, double) : boolean
+attendantShutDownStation(SelfCheckoutSoftware) : boolean
+attendantStartUpStation(SelfCheckoutSoftware) : boolean
+blockStation(SelfCheckoutSoftware) : boolean
+getAttendantID() : String
+lookupProdCode(SelfCheckoutSoftware, String) : PriceLookupCode
+lookupProdName(SelfCheckoutSoftware, PriceLookupCode) : String
+removeItemFromPurchase(SelfCheckoutSoftware, BarcodedItem) : bool
+setAttendantID(String) :void
+unBlockStation(SelfCheckoutSoftware) : boolean
MemberDatabase
+REGISTERED_MEMBERS : Map<String, Member>
-MemberDatabase()
Member
-card : Card
-name : String
-points : int
+Member(String, String)
+addPoints(int) : void
+getMemberCard(): Card
+getName() : String
+getPoints() : int
+redeemPoints(int) : void

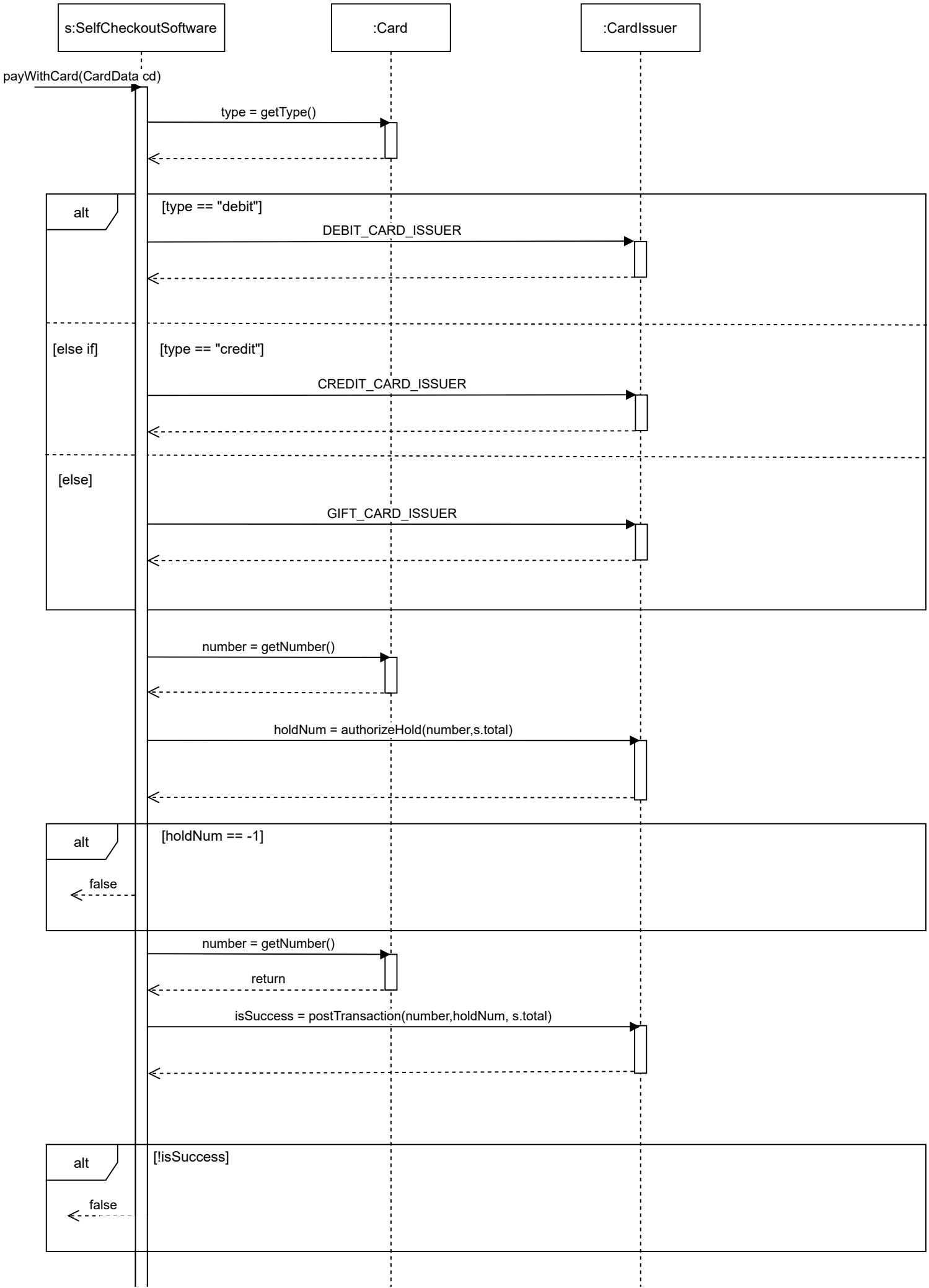

addOwnBag(weight) / add weight to scale
scanItem(barcode, weight) / add to total
addPLUItem(plu, weight) / add to total
attendantApprovesWeightDiff()
/ update discrepancy
Main Screen
Scanning/Entering
Items
scanItem(barcode, weight) / add to total
addPLUItem(plu, weight) / add to total
Fail To place Item
swipeMembershipCard (card)
[currentMember != null]
/ add points
Member Card
Bag Item
finishScanningItems()
[failToPlaceItem == false]
placeItemInBaggingArea(barcode)
/ update weight
payWithCash (notes)
payWithCard (card)
payWithCoins (coins)
failToPlaceItem() / lock system
Paying
Change
dispenseChange [payed > total] / dispense coins/notes
removePurchasedItems() [payed == true]
/ generateReceipt()
removePurchasedItems() [changeDispensed == true]
/ generateReceipt()
In Storage
resetStation() / reset all values

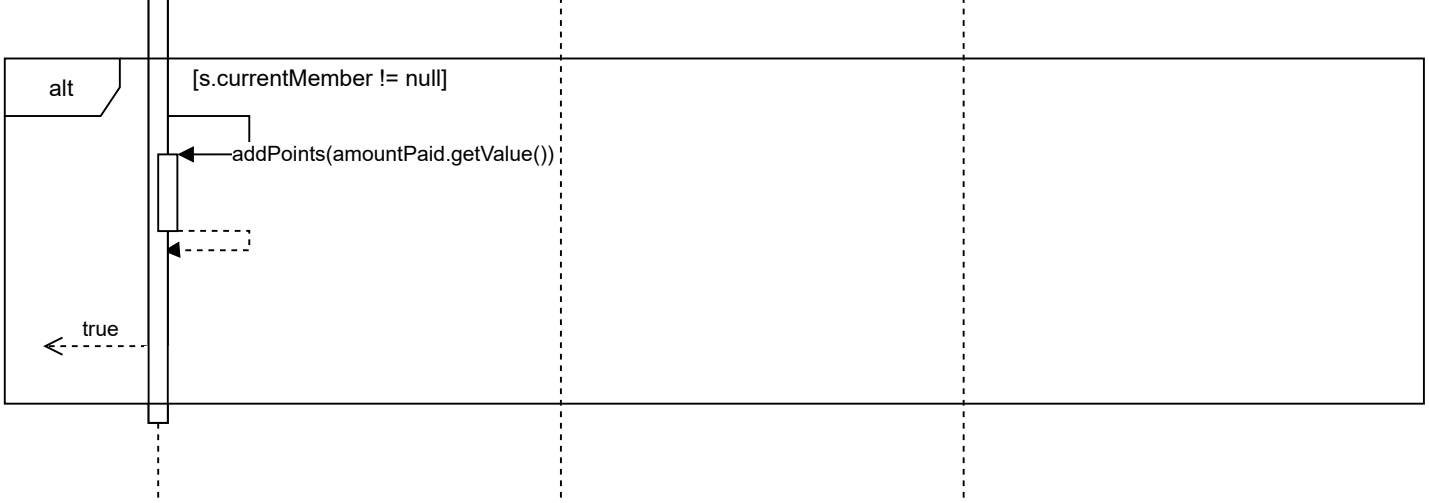

addOwnBag(weight)
removeScannedItem(item)
scanItem(barcode, weight)
/ add to total
attendantStartUpStation(control)
System Ready
addPLUItem(barcode, weight)
/ add to total
Scanning/Adding
Items
attendantApprovesWeightDiff()
removePluItem()
scanItem(barcode,weight)
addPLUItem(barcode,weight)
lookUpProductCode(productName)
placeItemInBaggingArea(barcode)
placePluItemInBaggingArea(plu))
Fail to Place Item
System Shutdown
addPLUItem(barcode, weight)
/ add to total
Look Up product
lookUpProductCode(productName)
Bagging Area
failToPlaceItem() / lock system
finishedScanningItems()
[failtoPlaceItem() == false]
returnToAddingItems() / addingItems = true
Checkout

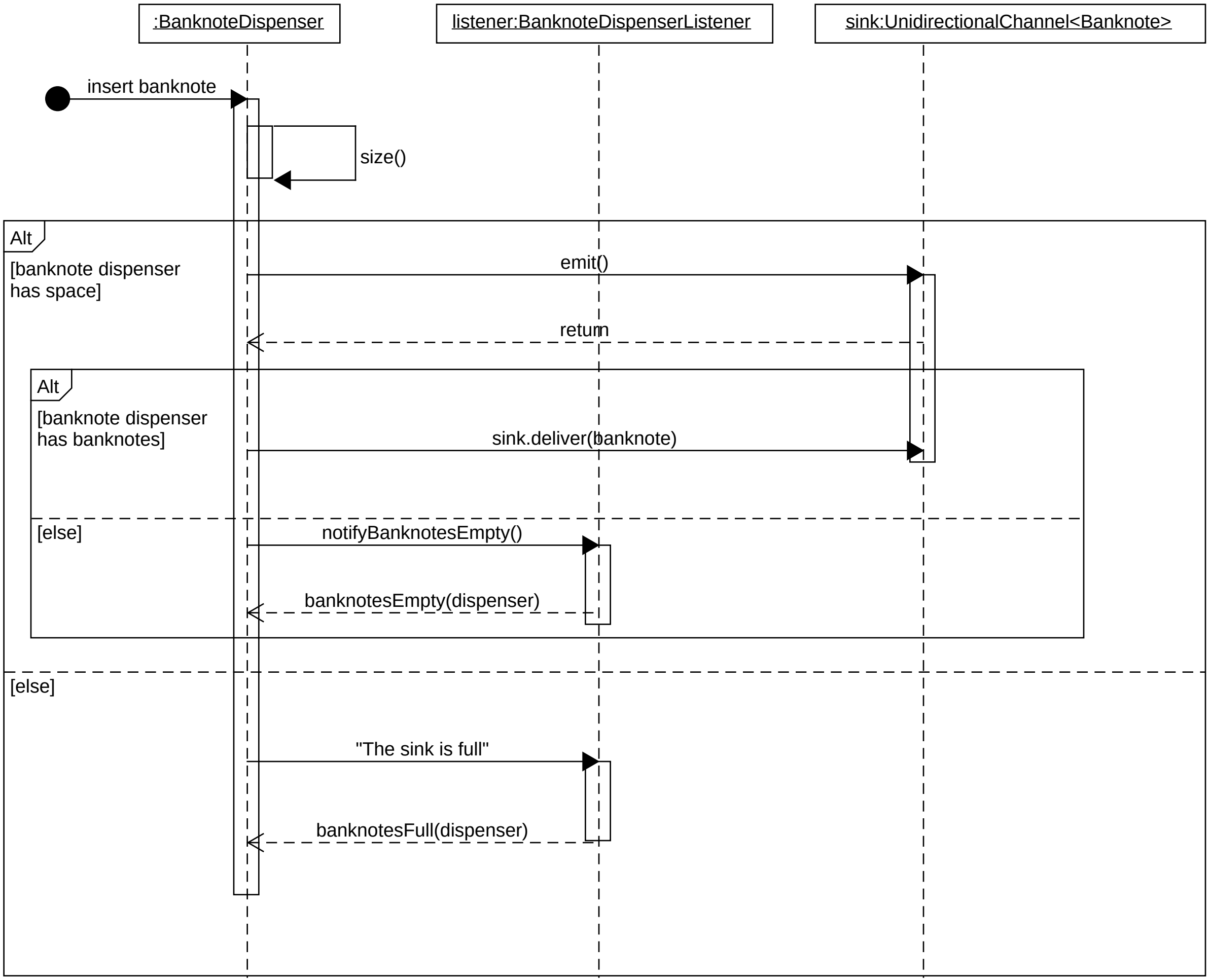

s:SelfCheckoutSoftware
prodDB:map<Barcode, BarcodedProduct>
invDB:map <Product,Integer>
:BarcodeScanner
:BigDecimal
scanItem(barcode, weight)
key = containsKey(barcode)
return
alt
[key]
product = get(barcode)
return
inventoryLeft = get(product)
return
alt
[inventoryLeft]
false
[else]
<<create>>
item:BarcodedItem (barcode, weight)
scan(item)
getPrice()
put(product, inventoryLeft-1)
add(item)
true
[else]
false

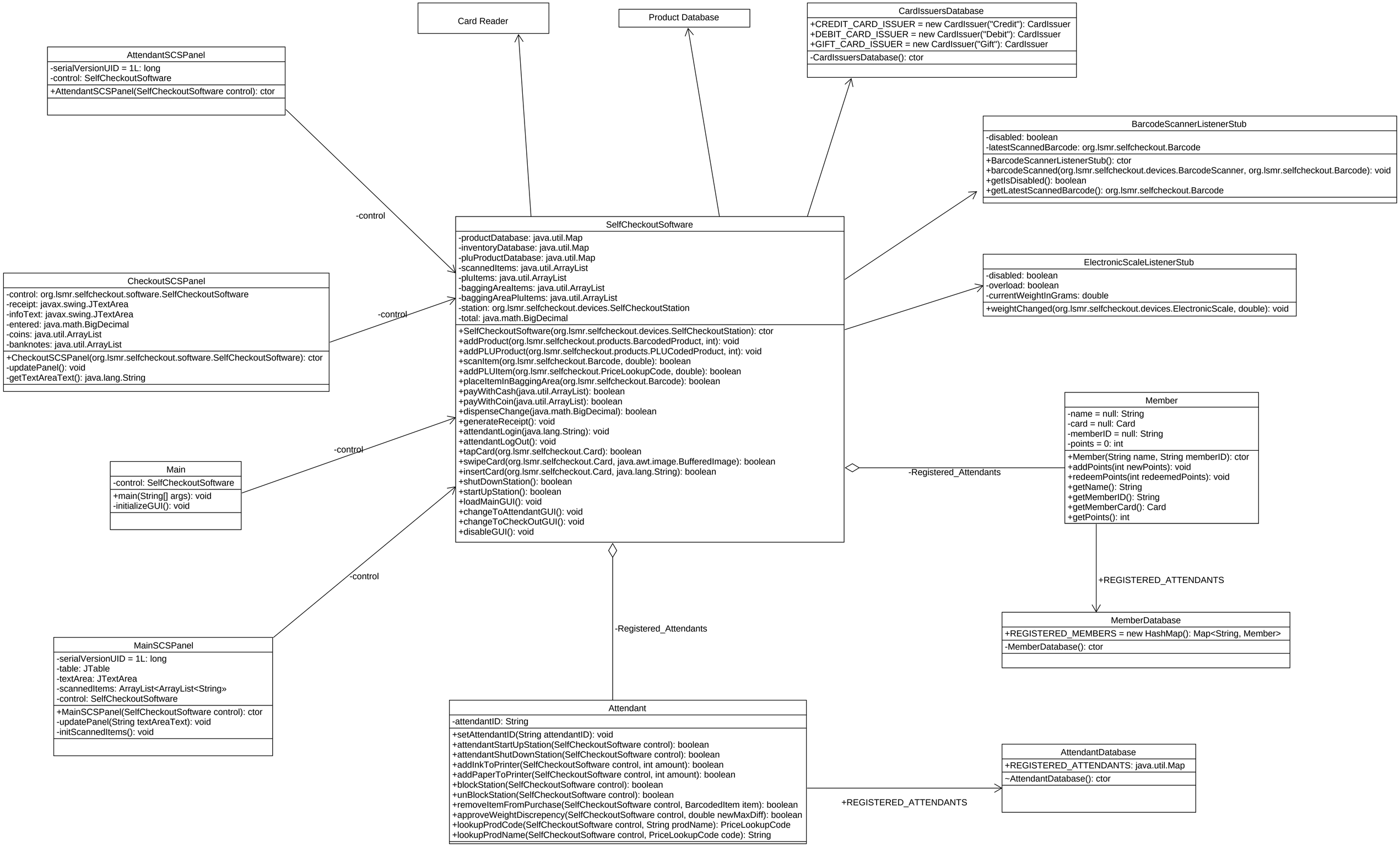

s:SelfCheckoutSoftware
p:Map<Barcode, BarcodedProduct>
ba: ElectronicScale
scannedItems: ArrayList
baItems: ArrayList
placeItemInBaggingArea(barcode)
k =containsKey(barcode)
return
alt
[k == true]
<<create>>
item:BarcodedItem
loop
[i < s.scannedItems.size()]
alt
[s.scannedItems.get(i).getBarcode().equals(barcode)]
item = s.scannedItems.get(i)
return
[else]
alt
[item != null]
s.ba.add(item)
return
s.baItems.add(item)
return
true
[else]
false
[else]
false

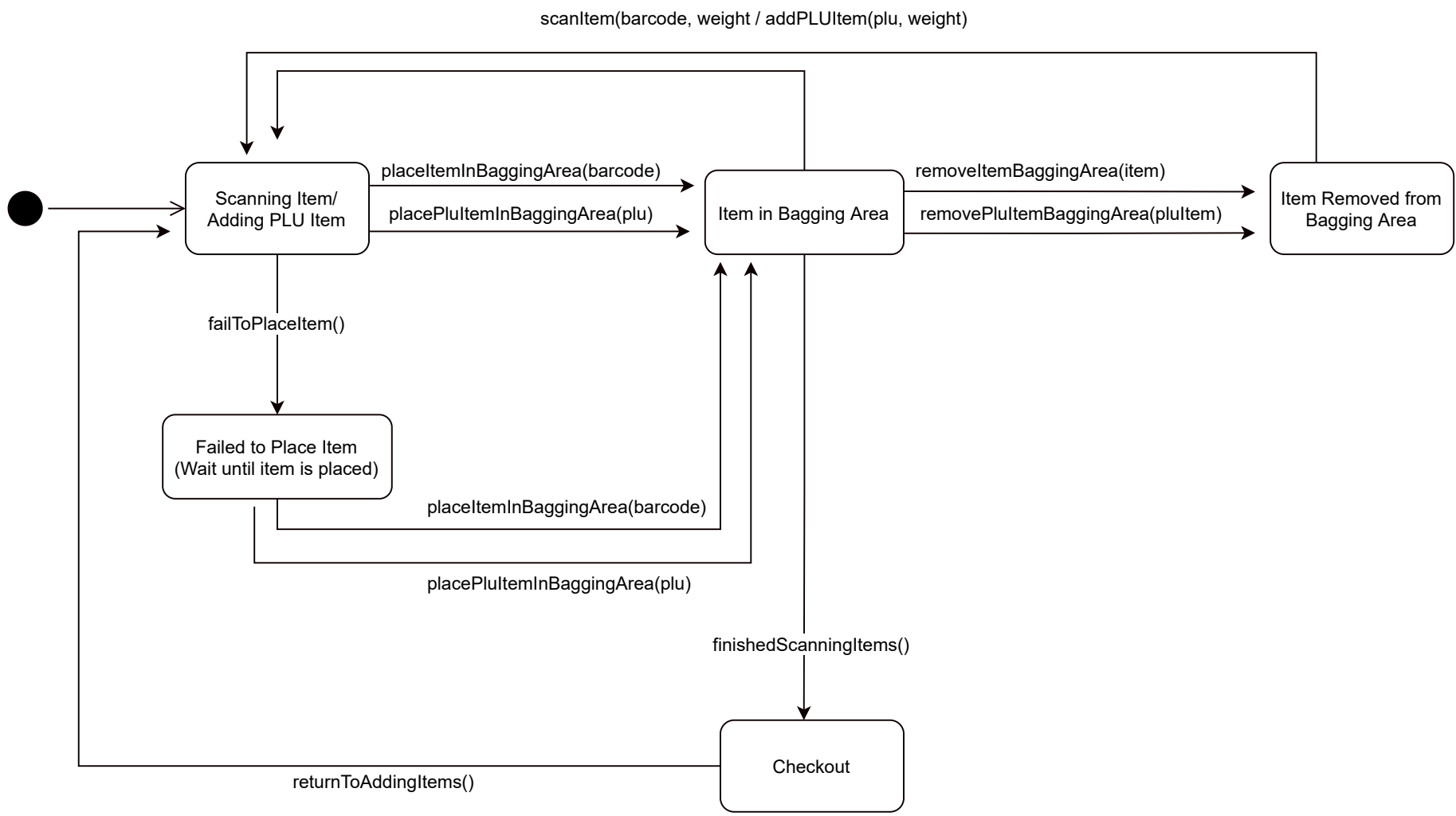

scanItem(barcode, weight) / add to total
Adding Items
finishedScanningItems
[failToPlaceItem == false]
Choose Payment
swipeMembershipCard (card)
[currentMember != null]
/ add points
Member Card
choosePayment()
swipeCard(card) /
cardReader.swipe(card)
tapCard(card) /
cardReader.tap(card)
insertCard(card) /
cardReader.insert(card)
payWithCoin(coins)
payWithCash(notes)
acceptCoin()
[payed < total]
/ isSuccess
acceptBanknote()
[payed < total]
/ isSuccess
cardIssuer.postTransaction(cd.getNumber(), holdNum, this.total)
[isSuccess == false]
/ exception
Pay with Card
Pay with Coins
Pay with Cash
dispenseChange [payed > total]
dispenseChange [payed > total]
generateReceipt [postTranscation == true]
generateReceipt() [payed == total]
generateReceipt() [payed == total]
Print Receipt
Dispense Change
generateReceipt() [changeDispensed == true]
removePurchasedItems() / transaction complete
Take Items
eject change() / collect from output
resetStation()

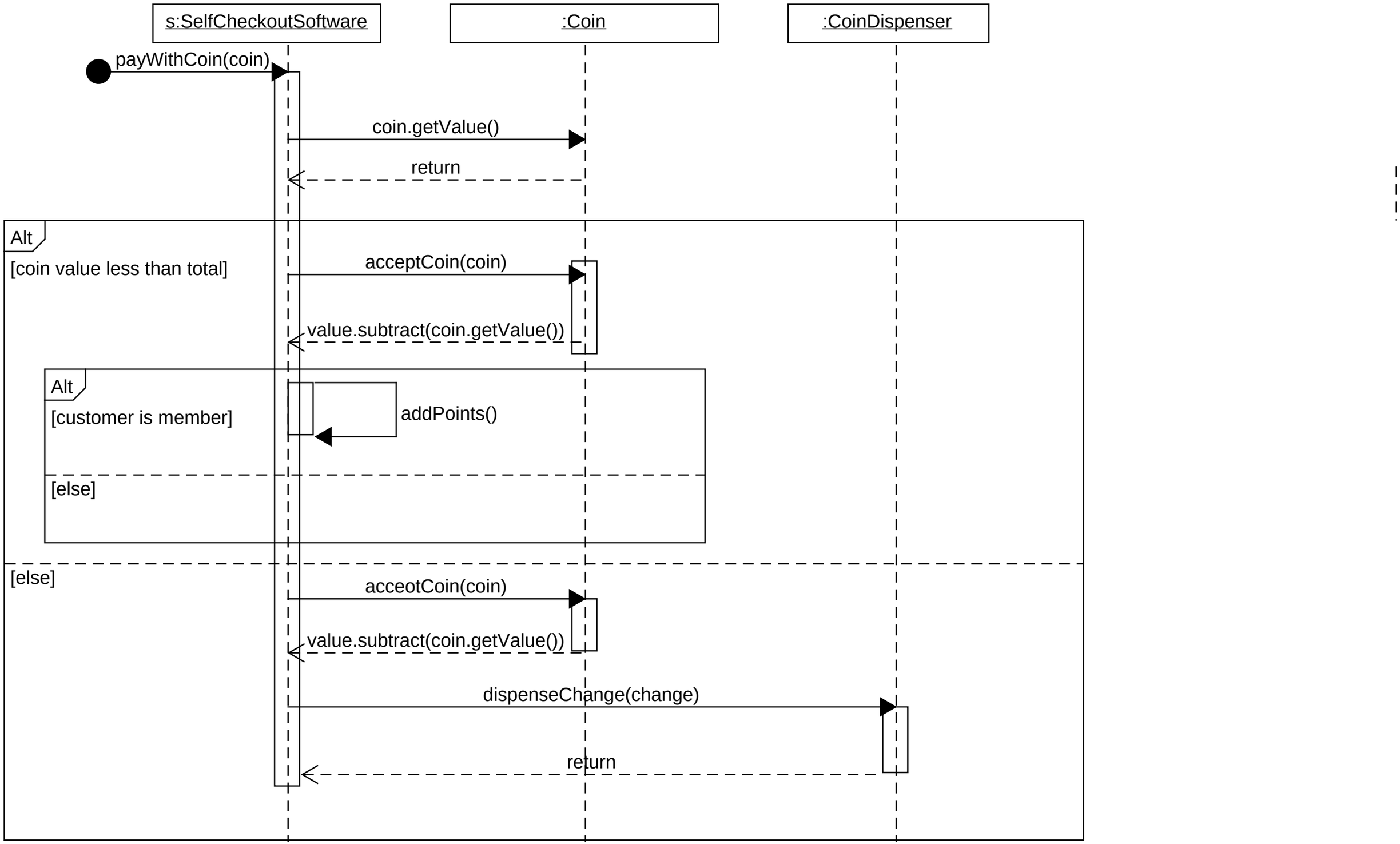

s:SelfCheckoutSoftware
:Card
:CardIssuer
payWithCard(CardData cd)
type = getType()
alt
[type == "debit"]
DEBIT_CARD_ISSUER
[else if]
[type == "credit"]
CREDIT_CARD_ISSUER
[else]
GIFT_CARD_ISSUER
number = getNumber()
holdNum = authorizeHold(number,s.total)
alt
[holdNum == -1]
false
number = getNumber()
return
isSuccess = postTransaction(number,holdNum, s.total)
alt
[!isSuccess]
false


alt
[s.currentMember != null]
addPoints(amountPaid.getValue())
true


:BanknoteDispenser
listener:BanknoteDispenserListener
sink:UnidirectionalChannel<Banknote>
insert banknote
size()
Alt
[banknote dispenser has space]
emit()
return
Alt
[banknote dispenser has banknotes]
sink.deliver(banknote)
[else]
notifyBanknotesEmpty()
banknotesEmpty(dispenser)
[else]
"The sink is full"
banknotesFull(dispenser)


Card Reader
Product Database
CardIssuersDatabase
+CREDIT_CARD_ISSUER = new CardIssuer("Credit"): CardIssuer
+DEBIT_CARD_ISSUER = new CardIssuer("Debit"): CardIssuer
+GIFT_CARD_ISSUER = new CardIssuer("Gift"): CardIssuer
-CardIssuersDatabase(): ctor
AttendantSCSPanel
-serialVersionUID = 1L: long
-control: SelfCheckoutSoftware
+AttendantSCSPanel(SelfCheckoutSoftware control): ctor
BarcodeScannerListenerStub
-disabled: boolean
-latestScannedBarcode: org.lsmr.selfcheckout.Barcode
+BarcodeScannerListenerStub(): ctor
+barcodeScanned(org.lsmr.selfcheckout.devices.BarcodeScanner, org.lsmr.selfcheckout.Barcode): void
+getIsDisabled(): boolean
+getLatestScannedBarcode(): org.lsmr.selfcheckout.Barcode
-control
SelfCheckoutSoftware
-productDatabase: java.util.Map
-inventoryDatabase: java.util.Map
-pluProductDatabase: java.util.Map
-scannedItems: java.util.ArrayList
-pluItems: java.util.ArrayList
-baggingAreaItems: java.util.ArrayList
-baggingAreaPluItems: java.util.ArrayList
-station: org.lsmr.selfcheckout.devices.SelfCheckoutStation
-total: java.math.BigDecimal
+SelfCheckoutSoftware(org.lsmr.selfcheckout.devices.SelfCheckoutStation): ctor
+addProduct(org.lsmr.selfcheckout.products.BarcodedProduct, int): void
+addPLUProduct(org.lsmr.selfcheckout.products.PLUCodedProduct, int): void
+scanItem(org.lsmr.selfcheckout.Barcode, double): boolean
+addPLUItem(org.lsmr.selfcheckout.PriceLookupCode, double): boolean
+placeItemInBaggingArea(org.lsmr.selfcheckout.Barcode): boolean
+payWithCash(java.util.ArrayList): boolean
+payWithCoin(java.util.ArrayList): boolean
+dispenseChange(java.math.BigDecimal): boolean
+generateReceipt(): void
+attendantLogin(java.lang.String): void
+attendantLogOut(): void
+tapCard(org.lsmr.selfcheckout.Card): boolean
+swipeCard(org.lsmr.selfcheckout.Card, java.awt.image.BufferedImage): boolean
+insertCard(org.lsmr.selfcheckout.Card, java.lang.String): boolean
+shutDownStation(): boolean
+startUpStation(): boolean
+loadMainGUI(): void
+changeToAttendantGUI(): void
+changeToCheckOutGUI(): void
+disableGUI(): void
ElectronicScaleListenerStub
-disabled: boolean
-overload: boolean
-currentWeightInGrams: double
+weightChanged(org.lsmr.selfcheckout.devices.ElectronicScale, double): void
CheckoutSCSPanel
-control: org.lsmr.selfcheckout.software.SelfCheckoutSoftware
-receipt: javax.swing.JTextArea
-infoText: javax.swing.JTextArea
-entered: java.math.BigDecimal
-coins: java.util.ArrayList
-banknotes: java.util.ArrayList
+CheckoutSCSPanel(org.lsmr.selfcheckout.software.SelfCheckoutSoftware): ctor
-updatePanel(): void
-getTextAreaText(): java.lang.String
-control
Member
-name = null: String
-card = null: Card
-memberID = null: String
-points = 0: int
+Member(String name, String memberID): ctor
+addPoints(int newPoints): void
+redeemPoints(int redeemedPoints): void
+getName(): String
+getMemberID(): String
+getMemberCard(): Card
+getPoints(): int
-Registered_Attendants
Main
-control: SelfCheckoutSoftware
+main(String[] args): void
-initializeGUI(): void
-control
+REGISTERED_ATTENDANTS
MemberDatabase
+REGISTERED_MEMBERS = new HashMap(): Map<String, Member>
-MemberDatabase(): ctor
MainSCSPanel
-serialVersionUID = 1L: long
-table: JTable
-textArea: JTextArea
-scannedItems: ArrayList<ArrayList<String»
-control: SelfCheckoutSoftware
+MainSCSPanel(SelfCheckoutSoftware control): ctor
-updatePanel(String textAreaText): void
-initScannedItems(): void
-Registered_Attendants
Attendant
-attendantID: String
+setAttendantID(String attendantID): void
+attendantStartUpStation(SelfCheckoutSoftware control): boolean
+attendantShutDownStation(SelfCheckoutSoftware control): boolean
+addInkToPrinter(SelfCheckoutSoftware control, int amount): boolean
+addPaperToPrinter(SelfCheckoutSoftware control, int amount): boolean
+blockStation(SelfCheckoutSoftware control): boolean
+unBlockStation(SelfCheckoutSoftware control): boolean
+removeItemFromPurchase(SelfCheckoutSoftware control, BarcodedItem item): boolean
+approveWeightDiscrepency(SelfCheckoutSoftware control, double newMaxDiff): boolean
+lookupProdCode(SelfCheckoutSoftware control, String prodName): PriceLookupCode
+lookupProdName(SelfCheckoutSoftware control, PriceLookupCode code): String
+REGISTERED_ATTENDANTS
AttendantDatabase
+REGISTERED_ATTENDANTS: java.util.Map
~AttendantDatabase(): ctor


scanItem(barcode, weight / addPLUItem(plu, weight)
Scanning Item/
Adding PLU Item
placeItemInBaggingAreaBarcode(barcode)
placePluItemInBaggingArea(plu)
Item in Bagging Area
removeItemBaggingArea(item)
removePluItemBaggingArea(pluItem)
Item Removed from
Bagging Area
failToPlaceItem()
Failed to Place Item
(Wait until item is placed)
placeItemInBaggingArea(barcode)
placePluItemInBaggingArea(plu)
finishedScanningItems()
Checkout
returnToAddingItems()


s:SelfCheckoutSoftware
:Coin
:CoinDispenser
payWithCoin(coin)
coin.getValue()
return
Alt
[coin value less than total]
acceptCoin(coin)
value.subtract(coin.getValue())
Alt
[customer is member]
addPoints()
[else]
[else]
acceotCoin(coin)
value.subtract(coin.getValue())
dispenseChange(change)
return