



اُنِيْوَرْسِيْتِيْ تِكْنُوْلُوْجِيْ مَآرَا  
UNIVERSITI  
TEKNOLOGI  
MARA

UNIVERSITI TEKNOLOGI MARA PERAK  
KAMPUS TAPAH

FACULTY OF COMPUTER AND MATHEMATICAL SCIENCE  
DIPLOMA IN COMPUTER SCIENCE

CSC186  
OBJECT ORIENTED PROGRAMMING

## COFFEE BEAN MANAGEMENT SYSTEM PROPOSAL

PREPARED BY

NAME	MATRIC NO.
DURRANI AFIQ BIN SAIDIN	2020769853
LUQMAN BIN AGUS SALIM	2020702587

GROUP

A4CS1103A

PREPARED FOR

PN. NURKHAIRIZAN BINTI KHAIRUDIN

DATE OF SUBMISSION

23 APRIL 2021

## TABLE OF CONTENT

OBJECTIVES .....	2
ANALYSIS.....	3
DESIGN .....	6
REFERENCES .....	21

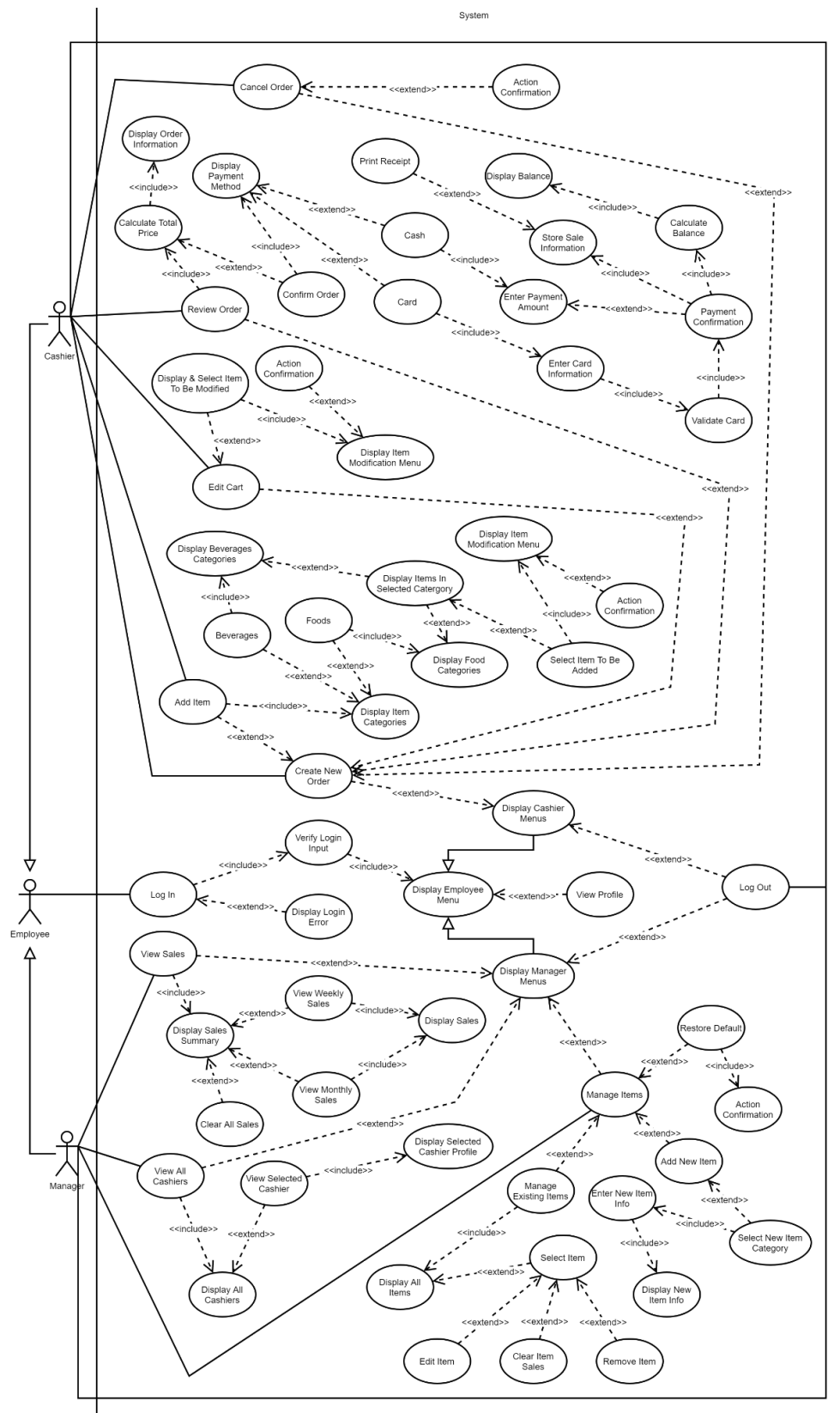
## 1. OBJECTIVES

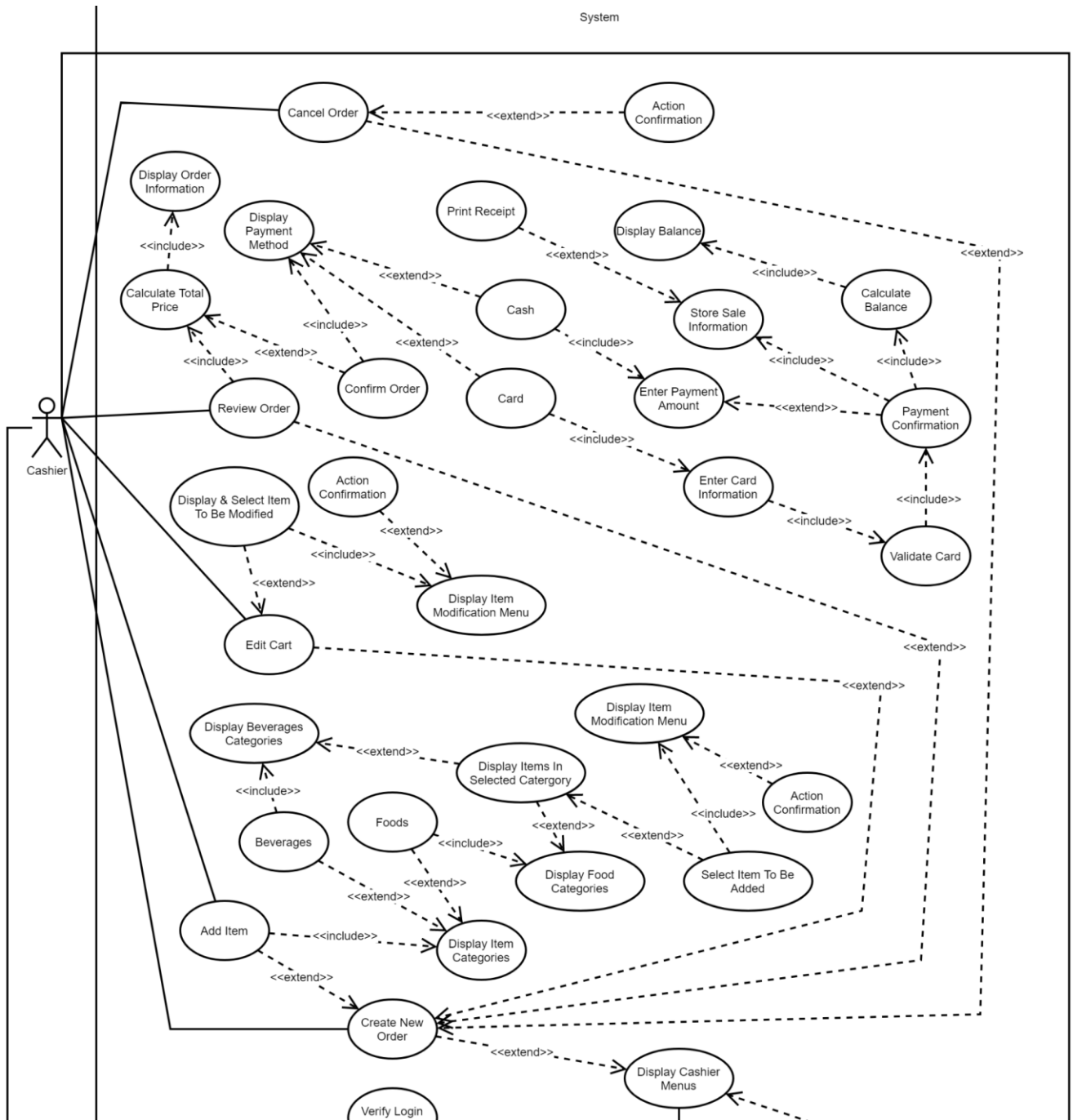
Tapah Road Coffee Shop has been increasing in popularity recently. Many customers around the country came to the cafe to get a taste of their premium coffee. The owner also wishes to diversify their products. This results in the workers getting overwhelmed and having difficulty keeping track of high amounts of orders. The cafe owner also wishes to get more accurate data of their overall sales. With this data, it will help the manager to keep track of the cashier commissions. It will also further help the manager to analyse their product sales to aid their business strategy. Hence, the cafe owner has requested for a system to be developed to automate their business process. This system will be written in Java.

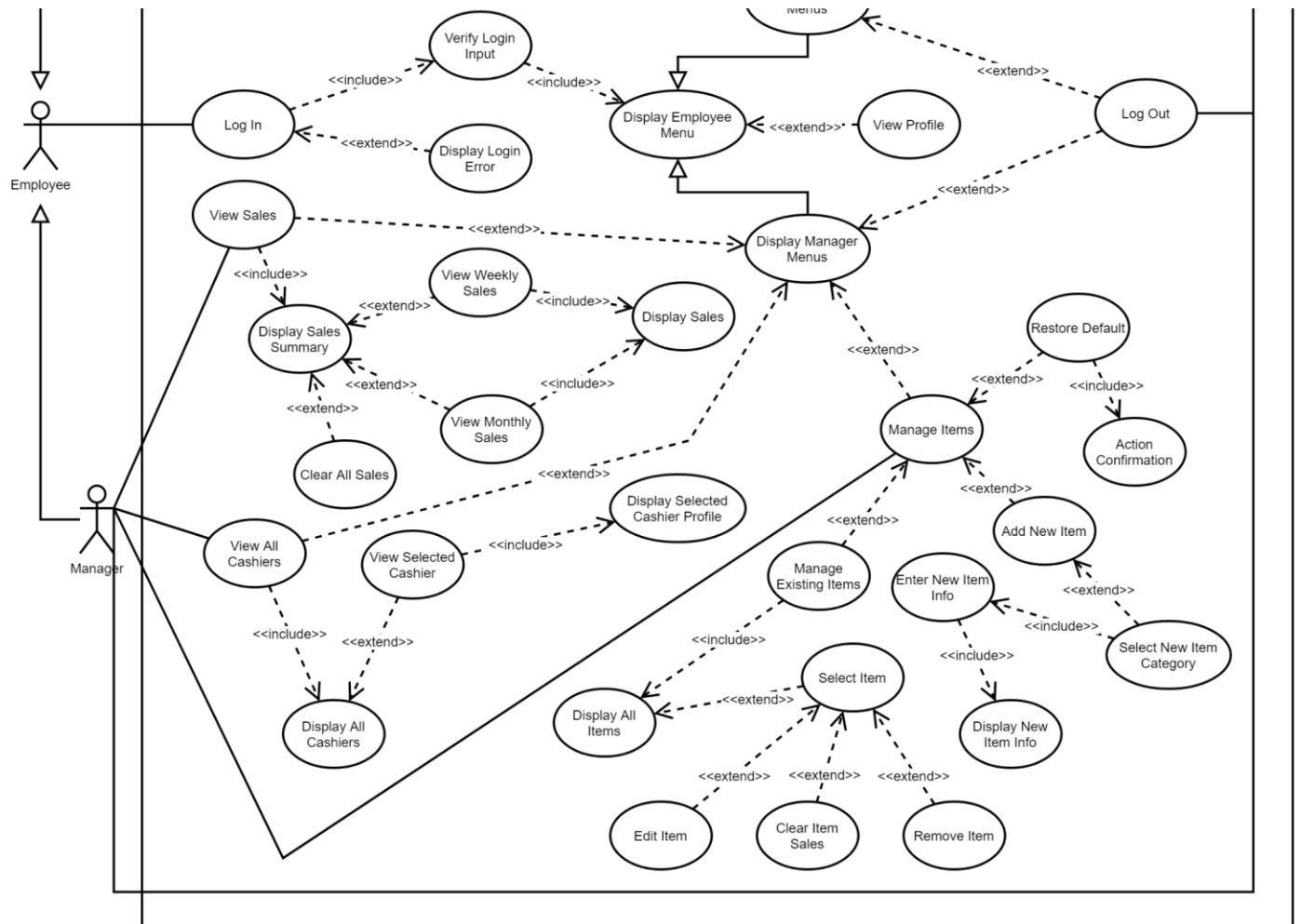
List of requested functionality:

1. Display the type of coffee and other food available and price.
2. Enter order information
3. Calculate & display total cost.
4. Calculate & display monthly collection by the type of coffee and staff.

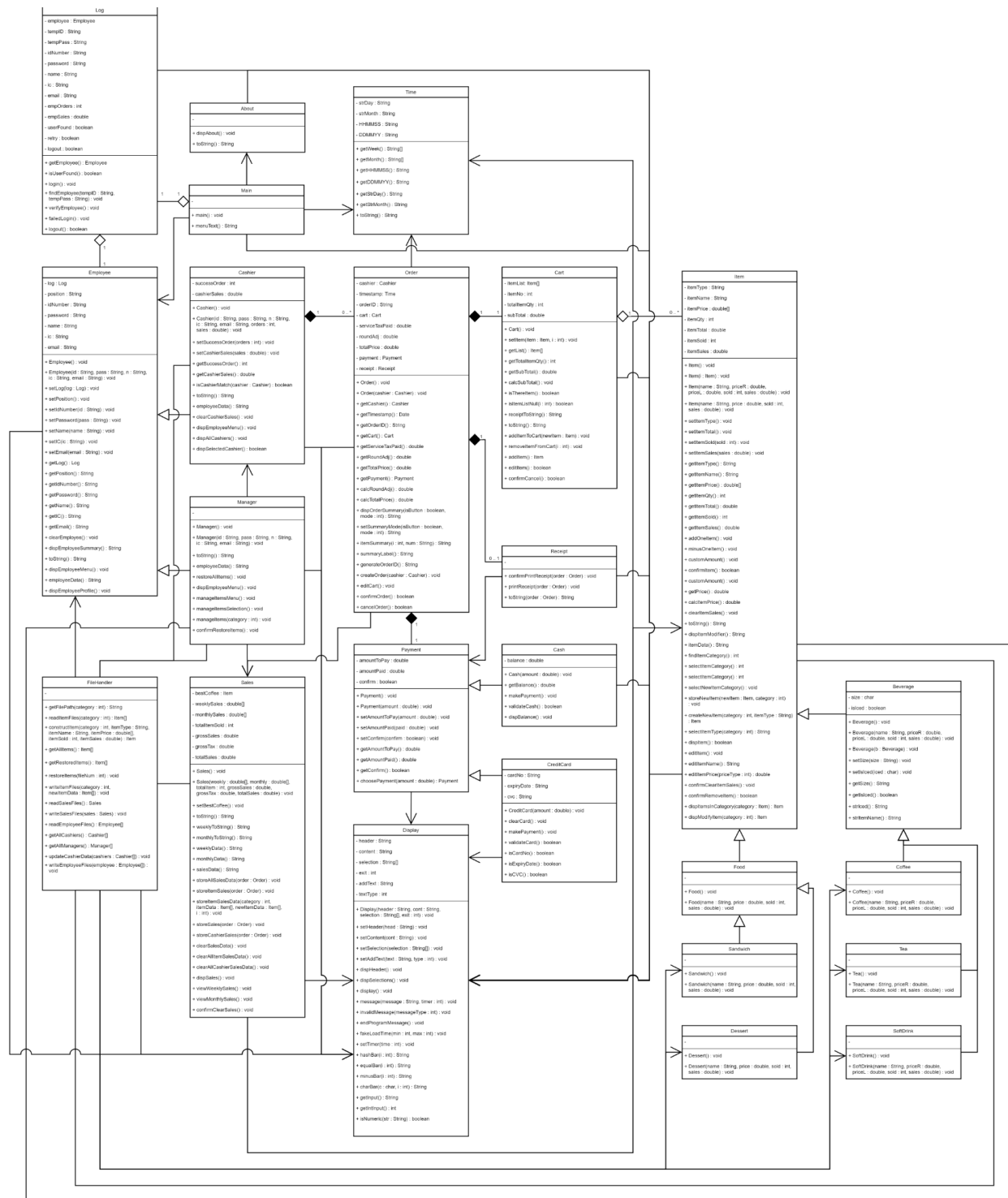
## 2. ANALYSIS (UML CASE DIAGRAM)



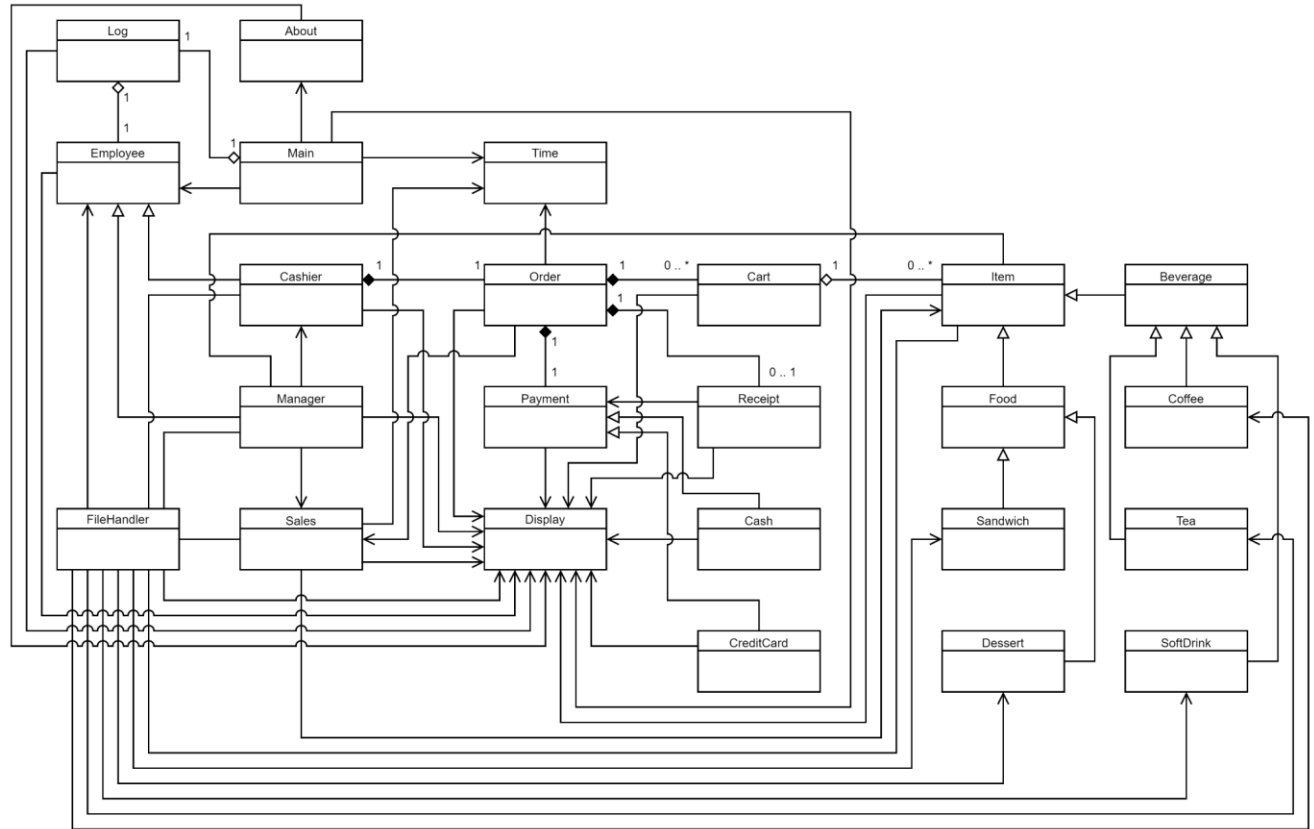




### 3. DESIGN (UML CLASS DIAGRAM)



## Simplified UML Class Diagram





## Main Class

Main
-
+ main() : void + menuText() : String

## About Class

About
-
+ dispAbout() : void + toString() : String

## Time Class

Time
- strDay : String - strMonth : String - HHMMSS : String - DDMMYY : String
+ getWeek() : String[] + getMonth() : String[] + getHHMMSS() : String + getDDMMYY() : String + getStrDay() : String + getStrMonth() : String + toString() : String

## Log Class

Log
<ul style="list-style-type: none"><li>- employee : Employee</li><li>- tempID : String</li><li>- tempPass : String</li><li>- idNumber : String</li><li>- password : String</li><li>- name : String</li><li>- ic : String</li><li>- email : String</li><li>- empOrders : int</li><li>- empSales : double</li><li>- userFound : boolean</li><li>- retry : boolean</li><li>- logout : boolean</li></ul>
<ul style="list-style-type: none"><li>+ getEmployee() : Employee</li><li>+ isUserFound() : boolean</li><li>+ login() : void</li><li>+ findEmployee(tempID : String, tempPass : String) : void</li><li>+ verifyEmployee() : void</li><li>+ failedLogin() : void</li><li>+ logout() : boolean</li></ul>

## Display Class

Display
<ul style="list-style-type: none"><li>- header : String</li><li>- content : String</li><li>- selection : String[]</li><li>- exit : int</li><li>- addText : String</li><li>- textType : int</li></ul>
<ul style="list-style-type: none"><li>+ Display(header : String, cont : String, selection : String[], exit : int) : void</li><li>+ setHeader(head : String) : void</li><li>+ setContent(cont : String) : void</li><li>+ setSelection(selection : String[]) : void</li><li>+ setAddText(text : String, type : int) : void</li><li>+ dispHeader() : void</li><li>+ dispSelections() : void</li><li>+ display() : void</li><li>+ message(message : String, timer : int) : void</li><li>+ invalidMessage(messageType : int) : void</li><li>+ endProgramMessage() : void</li><li>+ fakeLoadTime(min : int, max : int) : void</li><li>+ setTimer(time : int) : void</li><li>+ hashBar(i : int) : String</li><li>+ equalBar(i : int) : String</li><li>+ minusBar(i : int) : String</li><li>+ charBar(c : char, i : int) : String</li><li>+ getInput() : String</li><li>+ getIntInput() : int</li><li>+ isNumeric(str : String) : boolean</li></ul>

## FileHandler Class

FileHandler
-
<div>+ getFilePath(category : int) : String</div> <div>+ readItemFiles(category : int) : Item[]</div> <div>+ constructItem(category : int, itemType : String, itemName : String, itemPrice : double[], itemSold : int, itemSales : double) : Item</div> <div>+ getAllItems() : Item[]</div> <div>+ getRestoredItems() : Item[]</div> <div>+ restoreItems(fileNum : int) : void</div> <div>+ writeItemFiles(category : int, newItemData : Item[]) : void</div> <div>+ readSalesFiles() : Sales</div> <div>+ writeSalesFiles(sales : Sales) : void</div> <div>+ readEmployeeFiles() : Employee[]</div> <div>+ getAllCashiers() : Cashier[]</div> <div>+ getAllManagers() : Manager[]</div> <div>+ updateCashierData(cashiers : Cashier[]) : void</div> <div>+ writeEmployeeFiles(employee : Employee[]) : void</div>

## Employee Class

Employee
- log : Log
- position : String
- idNumber : String
- password : String
- name : String
- ic : String
- email : String

```
+ Employee() : void
+ Employee(id : String, pass : String, n : String, ic : String
  email : String) : void
+ setLog(log : Log) : void
+ setPosition() : void
+ setIdNumber(id : String) : void
+ setPassword(pass : String) : void
+ setName(name : String) : void
+ setIC(ic : String) : void
+ setEmail(email : String) : void
+ getLog() : Log
+ getPosition() : String
+ getIdNumber() : String
+ getPassword() : String
+ getName() : String
+ getIC() : String
+ getEmail() : String
+ clearEmployee() : void
+ dispEmployeeSummary() : String
+ toString() : String
+ dispEmployeeMenu() : void
+ employeeData() : String
+ dispEmployeeProfile() : void
```

## Manager & Cashier Class

Manager
-
+ Manager() : void + Manager(id : String, pass : String, n : String, ic : String, email : String) : void + toString() : String + employeeData() : String + restoreAllItems() : void + dispEmployeeMenu() : void + manageltemsMenu() : void + manageltemsSelection() : void + manageltems(category : int) : void + confirmRestoreItems() : void

Cashier
- successOrder : int - cashierSales : double
+ Cashier() : void + Cashier(id : String, pass : String, n : String, ic : String, email : String, orders : int, sales : double) : void + setSuccessOrder(orders : int) : void + setCashierSales(sales : double) : void + getSuccessOrder() : int + getCashierSales() : double + isCashierMatch(cashier : Cashier) : boolean + toString() : String + employeeData() : String + clearCashierSales() : void + dispEmployeeMenu() : void + dispAllCashiers() : void + dispSelectedCashier() : boolean

## Order Class

Order
- cashier : Cashier
- timestamp: Time
- orderID : String
- cart : Cart
- serviceTaxPaid : double
- roundAdj : double
- totalPrice : double
- payment : Payment
- receipt : Receipt

```
+ Order() : void
+ Order(cashier : Cashier) : void
+ getCashier() : Cashier
+ getTimestamp() : Date
+ getOrderID() : String
+ getCart() : Cart
+ getServiceTaxPaid() : double
+ getRoundAdj() : double
+ getTotalPrice() : double
+ getPayment() : Payment
+ calcRoundAdj() : double
+ calcTotalPrice() : double
+ dispOrderSummary(isButton : boolean,
mode : int) : String
+ setSummaryMode(isButton : boolean,
mode : int) : String
+ itemSummary(i : int, num : String) : String
+ summaryLabel() : String
+ generateOrderID() : String
+ createOrder(cashier : Cashier) : void
+ editCart() : void
+ confirmOrder() : boolean
+ cancelOrder() : boolean
```

## Cart Class

Cart
<ul style="list-style-type: none"><li>- itemList: Item[]</li><li>- itemNo : int</li><li>- totalItemQty : int</li><li>- subTotal : double</li></ul>
<ul style="list-style-type: none"><li>+ Cart() : void</li><li>+ setItem(item : Item, i : int) : void</li><li>+ getList() : Item[]</li><li>+ getTotalItemQty() : int</li><li>+ getSubTotal() : double</li><li>+ calcSubTotal() : void</li><li>+ isThereItem() : boolean</li><li>+ isItemListNull(i : int) : boolean</li><li>+ receiptToString() : String</li><li>+ toString() : String</li><li>+ addItemToCart(newItem : Item) : void</li><li>+ removeItemFromCart(i : int) : void</li><li>+ addItem() : Item</li><li>+ editItem() : boolean</li><li>+ confirmCancel() : boolean</li></ul>



## Item Class

Item	
<ul style="list-style-type: none"> <li>- itemType : String</li> <li>- itemName : String</li> <li>- itemPrice : double[]</li> <li>- itemQty : int</li> <li>- itemTotal : double</li> <li>- itemSold : int</li> <li>- itemSales : double</li> </ul>	
<ul style="list-style-type: none"> <li>+ Item() : void</li> <li>+ Item(i : Item) : void</li> <li>+ Item(name : String, priceR : double, priceL : double, sold : int, sales : double) : void</li> <li>+ Item(name : String, price : double, sold : int, sales : double) : void</li> <li>+ setItemType() : void</li> <li>+ setItemTotal() : void</li> <li>+ setItemSold(sold : int) : void</li> <li>+ setItemSales(sales : double) : void</li> <li>+ getItemType() : String</li> <li>+ getItemName() : String</li> <li>+ getItemPrice() : double[]</li> <li>+ getItemQty() : int</li> <li>+ getItemTotal() : double</li> <li>+ getItemSold() : int</li> <li>+ getItemSales() : double</li> <li>+ addOneItem() : void</li> </ul>	<ul style="list-style-type: none"> <li>+ minusOneItem() : void</li> <li>+ customAmount() : void</li> <li>+ confirmItem() : boolean</li> <li>+ customAmount() : void</li> <li>+ getPrice() : double</li> <li>+ calcItemPrice() : double</li> <li>+ clearItemSales() : void</li> <li>+ toString() : String</li> <li>+ dispItemModifier() : String</li> <li>+ itemData() : String</li> <li>+ findItemCategory() : int</li> <li>+ selectItemCategory() : int</li> <li>+ selectItemCategory() : int</li> <li>+ selectNewItemCategory() : void</li> <li>+ storeNewItem(newItem : Item, category : int) : void</li> <li>+ createNewItem(category : int, itemType : String) : Item</li> <li>+ selectItemType(category : int) : String</li> <li>+ dispItem() : boolean</li> <li>+ editItem() : void</li> <li>+ editItemName() : String</li> <li>+ editItemPrice(priceType : int) : double</li> <li>+ confirmClearItemSales() : void</li> <li>+ confirmRemoveItem() : boolean</li> <li>+ dispItemsInCategory(category : Item) : Item</li> <li>+ dispModifyItem(category : int) : Item</li> </ul>

## Beverage, Coffee, Tea, SoftDrink Classes

Beverage
- size : char
- isIced : boolean
+ Beverage() : void
+ Beverage(name : String, priceR : double, priceL : double, sold : int, sales : double) : void
+ Beverage(b : Beverage) : void
+ setSize(size : String) : void
+ setIced(iced : char) : void
+ getSize() : String
+ getIced() : boolean
+ strIced() : String
+ strItemName() : String

Coffee
-
+ Coffee() : void
+ Coffee(name : String, priceR : double, priceL : double, sold : int, sales : double) : void

Tea
-
+ Tea() : void
+ Tea(name : String, priceR : double, priceL : double, sold : int, sales : double) : void

SoftDrink
-
+ SoftDrink() : void
+ SoftDrink(name : String, priceR : double, priceL : double, sold : int, sales : double) : void

## Foods, Sandwich & Dessert Classes

Food
-
+ Food() : void + Food(name : String, price : double, sold : int, sales : double) : void

Dessert
-
+ Dessert() : void + Dessert(name : String, price : double, sold : int, sales : double) : void

Sandwich
-
+ Sandwich() : void + Sandwich(name : String, price : double, sold : int, sales : double) : void

## Payment, Cash, CreditCard & Receipt Class

Payment
<ul style="list-style-type: none"> <li>- amountToPay : double</li> <li>- amountPaid : double</li> <li>- confirm : boolean</li> </ul>
<ul style="list-style-type: none"> <li>+ Payment() : void</li> <li>+ Payment(amount : double) : void</li> <li>+ setAmountToPay(amount : double) : void</li> <li>+ setAmountPaid(paid : double) : void</li> <li>+ setConfirm(confirm : boolean) : void</li> <li>+ getAmountToPay() : double</li> <li>+ getAmountPaid() : double</li> <li>+ getConfirm() : boolean</li> <li>+ choosePayment(amount : double) : Payment</li> </ul>

Receipt
-
<ul style="list-style-type: none"> <li>+ confirmPrintReceipt(order : Order) : void</li> <li>+ printReceipt(order : Order) : void</li> <li>+ toString(order : Order) : String</li> </ul>

Cash
<ul style="list-style-type: none"> <li>- balance : double</li> </ul>
<ul style="list-style-type: none"> <li>+ Cash(amount : double) : void</li> <li>+ getBalance() : double</li> <li>+ makePayment() : void</li> <li>+ validateCash() : boolean</li> <li>+ dispBalance() : void</li> </ul>

CreditCard
<ul style="list-style-type: none"> <li>- cardNo : String</li> <li>- expiryDate : String</li> <li>- cvc : String</li> </ul>
<ul style="list-style-type: none"> <li>+ CreditCard(amount : double) : void</li> <li>+ clearCard() : void</li> <li>+ makePayment() : void</li> <li>+ validateCard() : boolean</li> <li>+ isCardNo() : boolean</li> <li>+ isExpiryDate() : boolean</li> <li>+ isCVC() : boolean</li> </ul>

## Sales Class

Sales
<ul style="list-style-type: none"><li>- bestCoffee : Item</li><li>- weeklySales : double[]</li><li>- monthlySales : double[]</li><li>- totalItemSold : int</li><li>- grossSales : double</li><li>- grossTax : double</li><li>- totalSales : double</li></ul>
<ul style="list-style-type: none"><li>+ Sales() : void</li><li>+ Sales(weekly : double[], monthly : double[], totalItem : int, grossSales : double, grossTax : double, totalSales : double) : void</li><li>+ setBestCoffee() : void</li><li>+ toString() : String</li><li>+ weeklyToString() : String</li><li>+ monthlyToString() : String</li><li>+ weeklyData() : String</li><li>+ monthlyData() : String</li><li>+ salesData() : String</li><li>+ storeAllSalesData(order : Order) : void</li><li>+ storeItemSales(order : Order) : void</li><li>+ storeItemSalesData(category : int, itemData : Item[], newItemData : Item[], i : int) : void</li><li>+ storeSales(order : Order) : void</li><li>+ storeCashierSales(order : Order) : void</li><li>+ clearSalesData() : void</li><li>+ clearAllItemSalesData() : void</li><li>+ clearAllCashierSalesData() : void</li><li>+ dispSales() : void</li><li>+ viewWeeklySales() : void</li><li>+ viewMonthlySales() : void</li><li>+ confirmClearSales() : void</li></ul>

#### **4. REFERENCES**

1. Ahmad, N., Mohtar, I. A., & Zamani, N. A. (2020). Teaching and Learning Module: Object Oriented Programming Using Java.
2. [https://en.wikipedia.org/wiki/List\\_of\\_coffee\\_drinks](https://en.wikipedia.org/wiki/List_of_coffee_drinks)
3. <https://www.teacoffeecup.com/recipe/different-types-of-coffee-explained-espresso-drink-recipes/>