**Assignment Task 2**

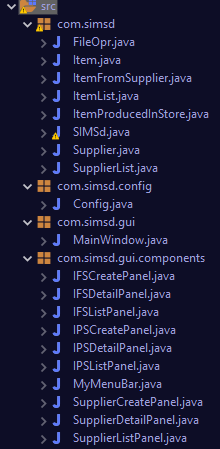
**Project Design**

**\*** This document is for the project ‘**Supermarket Inventory Management System for deli section (SIMSd)’** which includes:

1. the project design,
2. UML of used classes,
3. functionality of each method,
4. the purpose of each attribute,
5. Draft design of UI.

# Project Design

SIMSd will be developed in JAVA language with Swing for GUI, and Data will be stored in local JSON file. It will be a structured program that follow enforced programming language conventions and programming standards. Source file and package structure are as figure below:



com.simsd : for major class files of app

com.simsd.config: System config files

com.simsd.gui: App Window GUI files

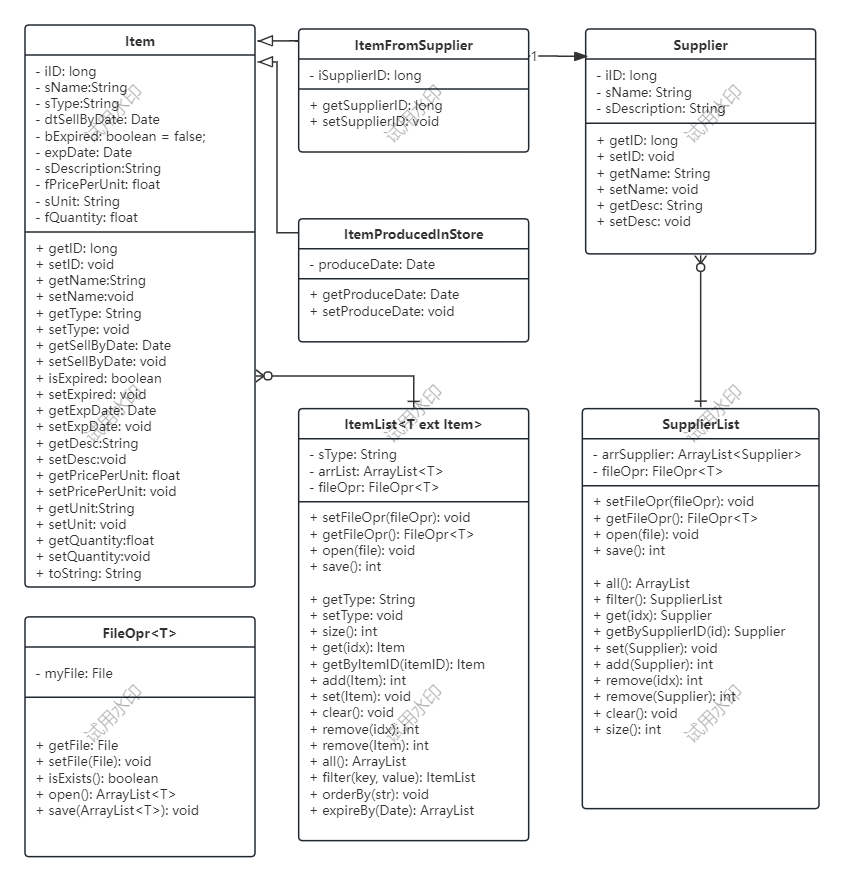
com.simsd.gui.components: GUI components

For Learning Outcome 4, **ArrayList** are wildly used in app as main Data Structure.

When starts, class **FileOpr** helps App to load the data from 3 local JSON files into class **Items** and class **Supplier**. There are 2 type of item, **ItemFromSupplier** and **ItemProducedInStore** which are both inherited from class **Item**. All loaded items are organized and stored in List, **ItemList** and **SupplierList** for app further use.

For functional, There’d be CRUD operations for both kind of Items and Supplier. Every time data modified, it will be writed into the JSON via FileOpr. As well , basic search functions are provided.

# UML



# Functionality and Attribute Description

Below are the description of some major function of class: FileOpr, ItemList and SupplierList.

**FileOpr**

Attr:

1. File myFile: the local data JSON file.

Function:

1. getFile() and setFile(File): the getter and setter of variable myFile.
2. isExists(): check if myFile exists or not. Return true if the myFile exists.
3. Open(File): open and read the file as specified data type, return an ArrayList of them.
4. Save(ArrayList): save the input ArrayList into myFile.

**ItemList**

Attr:

1. sType: Indicate what type of item the ItemList instance will hold.
2. arrList: The ArrayList that holds the item.
3. fileOpr: the File Operater instance that read and save the items data from local JSON.

Function:

1. ItemList(File, type): The Constructor will initialise the FileOpr and read the JSON for all the item of specified type.
2. Open(): let the fileOpr to read data into proper type arrList.
3. Save(): let the fileOpr to write arrList into proper JSON
4. Add(Item): add item into arrList
5. Remove(inx): remove item from arrList by index
6. Remove(Item): remove item from arrList by item
7. Clear(): empty the arrList.
8. getByItemID(id): return item from arrList by id
9. Get(int): return item from arrList by index
10. All(): return all items from arrList
11. Size(): return amount of items in arrList
12. Update(int, item): update the item in arrList by index
13. Update(item): update the item in arrList by itself
14. Filter(key, value): search items by key and value in arrList.
15. expireBefore(Date): return items resulte set that’d expired before a certain date.

**SupplierList**

Attr:

1. arrSupplier: The arraylist that holds the suppliers.
2. fileOpr: the File Operator instance that read and save the suppliers data from local JSON.

Function:

1. SupplierList(File): The Constructor which will read suppliers data from File.
2. Open(): let the fileOpr to read data into arrSupplier.
3. Save(): let the fileOpr to write data into JSON.
4. Add(Supplier): add supplier into arrSupplier.
5. Remove(index): remove supplier from arrSupplier by index.
6. Remove(Supplier): remove the specified supplier from arrSupplier.
7. Clear(): clean all suppliers from arrSupplier.
8. Get(int): get supplier from arrSupplier by index.
9. getBySupplierID(long): get supplier from arrSupplier by supplier ID
10. All(): return all suppliers in the arrSupplier.
11. Size(): return arrSupplier’s size.
12. Filter(key, value): search supplier by key and value in arrSupplier.
13. Set(Supplier): update the specified supplier.

# Purpose of attribute

We’ll descript some major attributes of class: Supplier, Item, ItemFromSupplier and ItemProducedInStore.

**Different kinds of relationship**:

* ItemFromSupplier and ItemProducedInStore are inheritated from Item.
* Supplier and ItemFromSupplier is one-to-one Foreign Key.

**Supplier**

1. Long iID: auto created supplier id, it’s actually the datetime in which the supplier created.
2. String sName: Supplier’s name.
3. String sDescription: Description of the supplier.

**Methods in this class are only getters and setters.**

**Item**

1. Long iID: auto created item id, it’s the datetime that item created.
2. String sName: Item’s name.
3. String sType: Item’s type.
4. Date dtSellByDate: date by which the item should be sold
5. Bool bExpired: Is the item expired.
6. String sDescription: item’s description.
7. Float fPricePerUnit: item’s unit price.
8. String sUnit: item’s unit
9. Float fQuantity: item’ s stocked quantity.

**Methods in this class are only getters and setters.**

**ItemFromSupplier extends Item**

1. iSupplierID: foreign key, supplier’s id.

**Methods in this class are only getters and setters.**

**ItemProducedInStore extends Item**

1. Date produceDate: the date that item was produced.

**Methods in this class are only getters and setters.**

# UI Sample screens

1. Index Page and Menu.

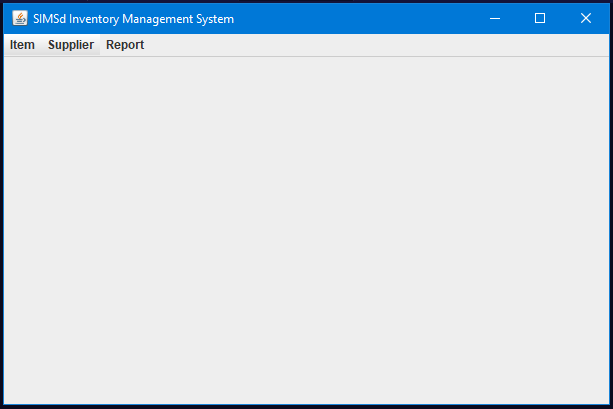


Fig 1.0 Index Screen

|  |  |
| --- | --- |
| Fig 1.1 Item From Supplier Menu | Fig 1.2 Item Produced In Store Menu |
| Fig 1.3 Supplier Menu |  |

1. Item From Supplier - List, Create, Edit, Delete

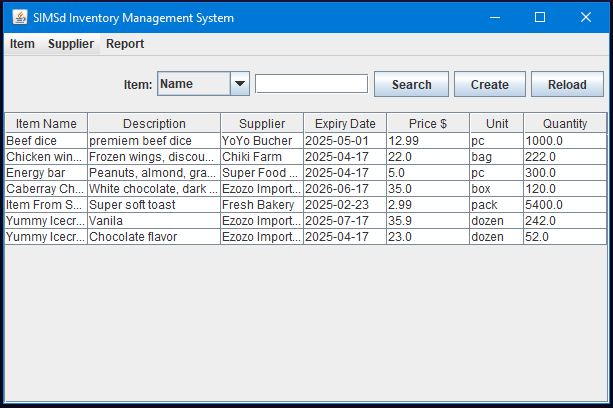


Fig 2.0 Item From Supplier List

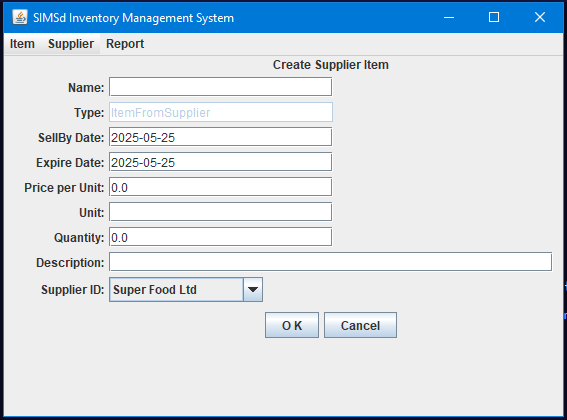


Fig 2.1 Item From Supplier Create

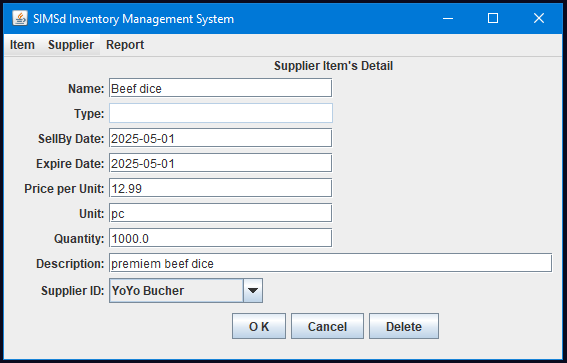


Fig 2.2 Item From Supplier Edit

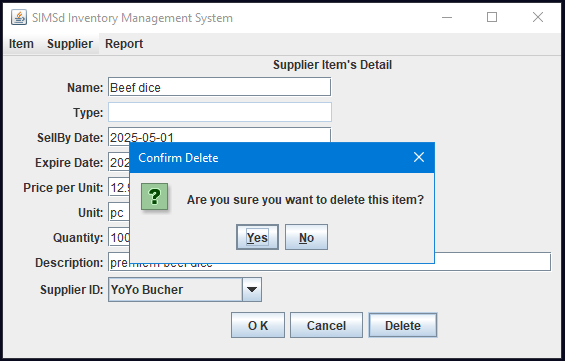


Fig 2.3 Item From Supplier- Delete Confirm

1. Item Produced In Store - List, Create, Edit, Delete

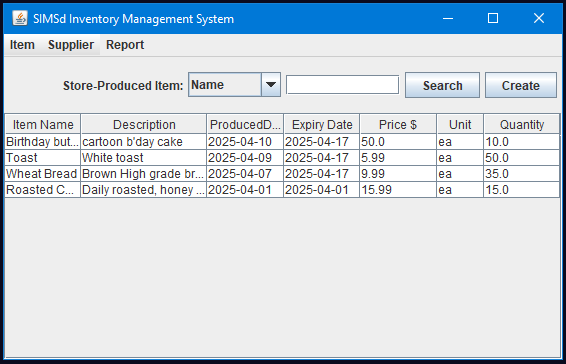


Fig 3.0 Item Produced in Store - List

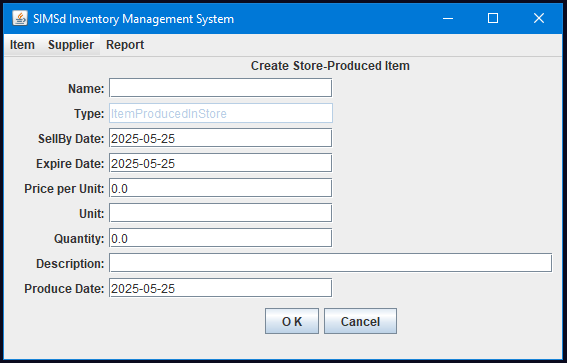


Fig 3.1 Item Produced in Store - Create

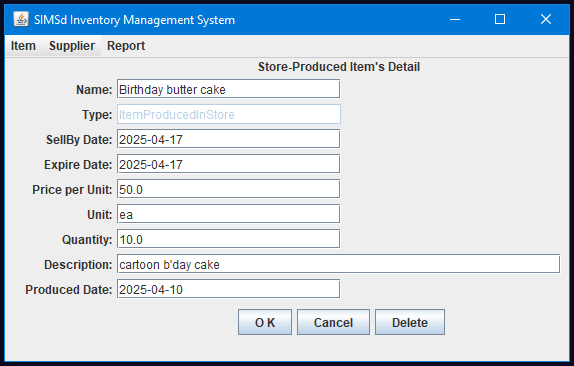


Fig 3.2 Item Produced in Store - Edit

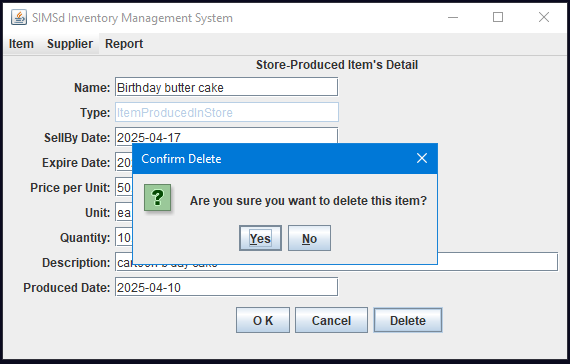


Fig 3.3 Item Produced in Store - Delete confirm

1. Report

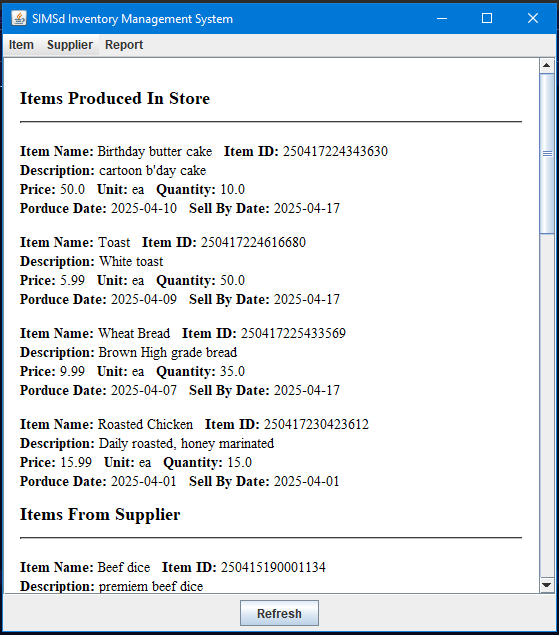


Fig 4.0 Report