Design & Code Size of SPL I

for

Shop Assistant

Submitted To

Dipok Chandra Das

Assistant Professor

Institute of Information Technology (IIT)

Noakhali Science and Technology University

Submitted by

Prosanto Deb (ASH1925005M)

Sultana Marjan (BKH1925010F)

Shahriar Ahmed (ASH1925015M)

Md. Alamgir Hossain (ASH1925016M)

Ratno Kumer Tripura (ASH1825042M)

Course Title: Software Metrics Lab

Course Code: SE 3204

Date of Submission: 05/03/2023

Table of Contents

Table of Figures

1. Project Information

Project Name	**Shop Assistant**
Supervised By	**Dipanita Saha** Assistant Professor Institute of Information Technology(IIT) Noakhali Science and Technology University
Team Members	**Prosanto Deb** (ASH1925005M)
	Sultana Marjan (BKH1925010F)
	Shahriar Ahmed (ASH1925015M)
	Md Alamgir Hossain (ASH1925016M)
	Ratno Kumer Tripura (ASH1825042M)
GitHub Link	Shop Assistant
Project Report	Shop Assistant Report

2. Code Size

Definition: Program code is an integral component of software. Such code includes source code, intermediate code, byte code, and even executable code. We look at approaches for directly measuring code size. We must take great care to clarify what we are counting and how we are counting it. Here, we have developed a program that will calculate these metrics if we give the location of the source files. An output.txt file will be attached with the assignment.

Process:

- Count of the number of physical lines (including blank lines).
- Count of all lines except blank lines and comments
- Count of all statements except comments (statements taking more than one-line count as only one line)

- Count of all lines except blank lines, comments, declarations and headings
- Count of all statements except blank lines, comments, declarations and headings
- Count of only the ESs (Executable Statement), not including exception conditions

NLOC= Non commented line
CLOC= Comment lines of program text
Total Size(LOC) = NCLOC + CLOC

Measure of density of comments in a program is (CLOC / LOC).

Values:

```
Enter your directory : E:\2-1\Java\Shop-Assistant 2.0\For Developers (NetBeans Formate)\src
```

Figure 1 Command for our program

```
|-------------------------------------------------------------------------|
|-----------------------------Summary-------------------------------------|
|-------------------------------------------------------------------------|
| Total number of files                                              16 |
| Total File Size in kilo bytes                          177.302734375 |
|-------------------------------------------------------------------------|
| Total number of physical lines (including blank lines)        :    5011 |
| Total number of blank lines                                   :     850 |
| Total number of commented lines                               :     467 |
| Total number of non commented lines                           :    4544 |
| Total Density of comments                                     :  9.32 % |
| Total number of statements                                    :    2317 |
| Total number of headings                                      :     259 |
| Total number of lines except blanks, comments & headings      :    3435 |
| Total number of characters                                    :  181558 |
| Average number of characters per line                         :   36.23 |
|-------------------------------------------------------------------------|
```

Figure 2 Final Report for all files.

Sample Output
Output.txt

3. Design Size

Definition: We can measure the size of a design in a manner similar to that used to measure code size. We will count design elements rather than LOCs. The elements that we count depend on the abstractions used to express the design, and the design aspects of interest. Thus, the appropriate size measure depends on the design methodology, the artifacts developed, and the level of abstraction.

Thus, we will measure size in terms of packages, design patterns, classes, interfaces, abstract classes, operations, and methods.

Process:

- **Packages**: Number of sub packages, number of classes, interfaces (Java), or abstract classes (C++).
- **Design Patterns**: Number of different design patterns used in a design.
- **Classes, interfaces, or abstract classes:** Number of public methods or operations, number of attributes.
- **Methods or Operations:** Number of parameters, number of overloaded versions of a method or operation One of the earliest object-oriented design size measures was the weighted methods per class (WMC) measure.

Values

3.1 Package

Package No.	**Package Name**	**Class No.**	**Class Name**
P1	Main	C1	MainPage
P2	Pictures		N/A

P3	ShopAssistant	C2	AddCostPage
		C3	AdvancedStockCheckPage
		C4	CreatePurchaseInvoicePage
		C5	CreateSalesInvoicePage
		C6	DailyIncomeCostPage
		C7	DueCheckPage
		C8	HomePage
		C9	LoginPage
		C10	SettingsPage
		C11	StaffAttendancePage
		C12	StockCheckPage

P4	Templates	C13	DashBoardTemplate
		C14	FrameSetup
		C15	InvoiceGeneratorTemplate
		C16	StartingTemplate
	Total Number of Classes	**16**	

3.2 Sub Package

Number of Sub Package	**0**

3.3 Classes

Class No.	Class Name	Method Signature	Number of Para-meters	Total Para-meters	Method Weights
C1	MainPage	main(String[] args)	1	1	2

		setComponentPanel()	0		1
				Weighted Method Per Class	3
C2	AddCostPage	setAddCostFeatures()	0	0	1
		setInputPanel()	0		2
		setMainPanel()	0		1
		setPageButton()	0		3
				Weighted Method Per Class	7
C3	AdvancedStock CheckPage	setAdvancedStockCheckFeatures()	0	1	1
		setMainPanel()	0		2
		setManualDate()	0		3
		setModelNumberCombobox(String date)	1		4
		setOutputPanel()	0		3
		setPageButton()	0		2
				Weighted Method Per Class	15
C4	CreatePurchase InvoicePage	setPageButton()	0	0	1
		setQuantityTextFieldListener()	0		2
				Weighted Method Per Class	3
C5	CreateSales InvoicePage	setPageButton()	0	0	1
		setQuantityTextFieldListener()	0		1
				Weighted Method Per Class	2
C6	DailyIncome CostPage	setDailyIncomeCostFeatures()	0	2	1
		setIncomeCostInformation	2		2

		(String keyword,String date)			
		setInputPanel()	0		1
		setMainPanel()	0		2
		setManualDate()	0		2
		setPageButton()			1
				Weighted Method Per Class	9
C7	DueCheckPage	checkDue(String selectedItem,String filename)	2	3	1
		checkMoneyDigit(char c)	1		2
		refresh()	0		3
		setCompanyDue()	0		4
		setCustomerDue()	0		5
		setDueCheckFeatures()	0		1
		setMainPanel()	0		2
		setOutputPanel()	0		3
		setPageButton()	0		4
				Weighted Method Per Class	25
C8	HomePage	setHTMLText(String filename)	1	1	1
		setMainPanel()	0		2
		setPageButton()	0		3
		setSeachFeatures()	0		4
		setSearchPanel()	0		5
				Weighted Method Per Class	15

C9	LoginPage	setComponetPanel()	0	0	1
		setLoginFeatures()	0		2
				Weighted Method Per Class	3
C10	SettingsPage	setAddNewMemberSectionVisible (boolean visibility)	1	4	5
		setAddNewModelRemoveMemberPanel ()	0		4
		setAddNewModelSectionVisible (boolean visibility)	1		3
		setCenterPanel()	0		2
		setMainPanel()	0		1
		setMemberNames()	0		1
		setMemberNames()	0		2
		setPasswordChangeAddNewStaffPanel( )	0		3
		setPasswordSectionVisible(boolean visibility)	1		4
		setRemoveMemberSectionVisible (Boolean visibility)	1		5
		setSettingsPageFeatures()	0		1
				Weighted Method Per Class	31
C11	StaffAttendance Page	isAttendanceGiven(String name,String date)	2	2	1
		setCenterPanel()	0		2

		setMainPanel()	0		1
		setMemberNames()	0		1
		setOutputTextArea()	0		2
		setPageButton()	0		1
		setStaffAttendanceFeatures()	0		1
		setintputPanel()	0		1
				Weighted Method Per Class	10
C12	StockCheckPage	setMainPanel()	0	0	1
		setOutputPanel()	0		1
		setPageButton()	0		1
		setStockCheckFeatures()	0		2
				Weighted Method Per Class	5
C13	DashBoard Template	getButton(String buttonName)	1	1	1
		setActionListeners()	0		2
		setContainer()	0		3
		setFrame()	0		4
		setHorizontalPanelComponents()	0		5
		setPanels()	0		1
		setVerticalPanelComponents()	0		2
				Weighted Method Per Class	18
C14	FrameSetup	getPassword()	0	9	1
		getRemainingQuantity(String modelNumber,String filename)	2		2

		getRemainingQuantityUsingDate(String date,String modelNumber,String filename)	3		3
		getTotalStock(String modelNumber,String purchaseFilename,String salesFilename)	3		4
		isDigit(char character)	1		5
		setAppIcon()	0		1
		setContainer()	0		2
		setDate()	0		3
		setFrame()	0		4
		setProductTypes()	0		5
		setReminder()	0		1
Weighted Method Per Class					31
C15	InvoiceGenerator Template	addHTMLbegintext(String keyword)	1	9	1
		addHTMLendtext(String keyword)	1		1
		addHTMLmidtext()	0		2
		addToCart(String fileName)	1		2
		checkAllFilledUp()	0		3
		getButton(String buttonName)	1		3
		setButtons()	0		4
		setContainer()	0		4
		setDue()	0		5
		setFrame()	0		5

		setId(String fileName)	1		1
		setInputPanel(String keyword,String fileName)	2		1
		setListeners(String keyword,String fileName)	2		2
		setOutputPanel()	0		2
		setPanels()	0		3
		setPayment()	0		3
				Weighted Method Per Class	42
C16	Starting Template	setContainer()	0	0	1
		setPanels()	0		2
		setPicturePanel()	0		3
				Weighted Method Per Class	6
				Total Weighted Method Per Class	**225**

3.4 Weighted Methods

Average Weighted Method Per Class	**225/16 = 14.06**

3.5 Interfaces

Number of Interfaces	**0**

3.6 Abstract Class

Number of Abstract Classes	**0**

3.7 Private Methods

Number of Private Methods	**0**

3.8 Public Methods

Number of Public Methods	**98**

3.9 Weighted Methods

Total Number of weighted methods	**24**

3.10 Parameters

Number of Parameters	**33**

3.11 Parameters per method

Number of average parameters per method	**0.34**

3.12 Polymorphism

Number of overloading methods	**0**
Number of overriding methods	**3**

3.13 Design Pattern & Code Smell

Design pattern used	**Builder Pattern, Composite**

Code smells Occurred	**Long method, Large Class, Temporary Field, Inappropriate Intimacy**