**Web Code Metrics - Catero**

**Web Metric at System Level** Catero

Tools used for metrics:

1) Source Monitor

2)Cyvis metric tool

**1) Source Monitor**

The freeware program **SourceMonitor** lets you see inside your software source code to find out how much code you have and to identify the relative complexity of your modules. For example, you can use SourceMonitor to identify the code that is most likely to contain defects and thus warrants formal review.

|  |  |
| --- | --- |
| Baseline | Description |
| Files # | 20 |
| Lines # | 1163 |
| Statements # | 737 |
| % branches | 6.4 |
| Calls # | 340 |
| % Comments | 24.3 |
| Classes # | 22 |
| Methods/Class # | 2.77 |
| Avg.statements/methods | 6.15 |
| Max complexity | 14\* |
| Max depth | 6 |
| Avg .depth | 1.51 |
| Avg.complexity | 1.79\* |

|  |  |
| --- | --- |
| Invalidate.java | Description |
| Lines # | 42 |
| Statements # | 24 |
| % branches | 0.0 |
| Calls # | 9 |
| % Comments | 33.3 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 3 |
| Max complexity | 1\* |
| Max depth | 2 |
| Avg .depth | 0.92 |
| Avg.complexity | 1.00\* |

|  |  |
| --- | --- |
| OrderHistory.java | Description |
| Lines # | 43 |
| Statements # | 24 |
| % branches | 0 |
| Calls # | 7 |
| % Comments | 34.9 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 2 |
| Max complexity | 1\* |
| Max depth | 2 |
| Avg .depth | 0.67 |
| Avg.complexity | 1.00\* |

|  |  |
| --- | --- |
| OrderMail.java | Description |
| Lines # | 80 |
| Statements # | 56 |
| % branches | 1.8 |
| Calls # | 28 |
| % Comments | 20.0 |
| Classes # | 2 |
| Methods/Class # | 3 |
| Avg.statements/methods | 4 |
| Max complexity | 3\* |
| Max depth | 3 |
| Avg .depth | 1.03 |
| Avg.complexity | 1.67\* |

|  |  |
| --- | --- |
| OrderPlaced.java | Description |
| Lines # | 50 |
| Statements # | 31 |
| % branches | 6.5 |
| Calls # | 11 |
| % Comments | 28.0 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 4 |
| Max complexity | 3\* |
| Max depth | 3 |
| Avg .depth | 1.03 |
| Avg.complexity | 1.67\* |

|  |  |
| --- | --- |
| PlaceOrder.java | Description |
| Lines # | 50 |
| Statements # | 31 |
| % branches | 3.2 |
| Calls # | 16 |
| % Comments | 28.0 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 5 |
| Max complexity | 2\* |
| Max depth | 3 |
| Avg .depth | 1.26 |
| Avg.complexity | 1.33\* |

|  |  |
| --- | --- |
| RemoveFromCart.java | Description |
| Lines # | 54 |
| Statements # | 33 |
| % branches | 18.2 |
| Calls # | 15 |
| % Comments | 25.9 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 6 |
| Max complexity | 7\* |
| Max depth | 4 |
| Avg .depth | 1.67 |
| Avg.complexity | 3\* |

|  |  |
| --- | --- |
| ShowCart.java | Description |
| Lines # | 62 |
| Statements # | 41 |
| % branches | 9.8 |
| Calls # | 18 |
| % Comments | 22.6 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 7.67 |
| Max complexity | 5\* |
| Max depth | 3 |
| Avg .depth | 1.46 |
| Avg.complexity | 2.33\* |

|  |  |
| --- | --- |
| ShowCart.java | Description |
| Lines # | 62 |
| Statements # | 41 |
| % branches | 9.8 |
| Calls # | 18 |
| % Comments | 22.6 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 7.67 |
| Max complexity | 5\* |
| Max depth | 3 |
| Avg .depth | 1.46 |
| Avg.complexity | 2.33\* |

|  |  |
| --- | --- |
| ShowProduct.java | Description |
| Lines # | 41 |
| Statements # | 23 |
| % branches | 0 |
| Calls # | 7 |
| % Comments | 34.1 |
| Classes # | 1 |
| Avg.statements/methods | 2 |
| Max complexity | 1\* |
| Max depth | 2 |
| Avg .depth | 0.70 |
| Avg.complexity | 1.00\* |

|  |  |
| --- | --- |
| Statements # | 24 |

|  |  |
| --- | --- |
| ViewRegisteredCustomer.java | Description |
| Lines # | 42 |
| Statements # | 24 |
| % branches | 0.0 |
| Calls # | 7 |
| % Comments | 33.3 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 2 |
| Max complexity | 1\* |
| Max depth | 2 |
| Avg .depth | 0.67 |
| Avg.complexity | 1\* |

|  |  |
| --- | --- |
| ViewProduct.java | Description |
| Lines # | 41 |
| Statements # | 23 |
| % branches | 0 |
| Calls # | 7 |
| % Comments | 34.1 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 2 |
| Max complexity | 2\* |
| Max depth | 2 |
| Avg .depth | 0.7 |
| Avg.complexity | 1\* |

|  |  |
| --- | --- |
| AddProduct.java | Description |
| Lines # | 127 |
| Statements # | 90 |
| % branches | 14.4 |
| Calls # | 57 |
| % Comments | 9.4 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 20.67 |
| Max complexity | 14\* |
| Max depth | 6 |
| Avg .depth | 2.51 |
| Avg.complexity | 5.33\* |

|  |  |
| --- | --- |
| AddToCart.java | Description |
| Lines # | 64 |
| Statements # | 43 |
| % branches | 9.3 |
| Calls # | 25 |
| % Comments | 20.3 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 8.33 |
| Max complexity | 5\* |
| Max depth | 4 |
| Avg .depth | 1.84 |
| Avg.complexity | 2.33\* |

|  |  |
| --- | --- |
| AddminLogin.java | Description |
| Lines # | 51 |
| Statements # | 32 |
| % branches | 6.3 |
| Calls # | 15 |
| % Comments | 27.5 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 5 |
| Max complexity | 3\* |
| Max depth | 3 |
| Avg .depth | 1.25 |
| Avg.complexity | 1.67\* |

|  |  |
| --- | --- |
| CustomerGmail.java | Description |
| Lines # | 41 |
| Statements # | 23 |
| % branches | 0 |
| Calls # | 9 |
| % Comments | 34.1 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 3 |
| Max complexity | 1\* |
| Max depth | 2 |
| Avg .depth | 0.96 |
| Avg.complexity | 1\* |

|  |  |
| --- | --- |
| CustomerLogin.java | Description |
| Lines # | 56 |
| Statements # | 35 |
| % branches | 8.6 |
| Calls # | 17 |
| % Comments | 26.8 |
| Classes # | 1 |
| Methods/Class # | 3 |
| Avg.statements/methods | 6.33 |
| Max complexity | 4\* |
| Max depth | 4 |
| Avg .depth | 1.72 |
| Avg.complexity | 2\* |

**2) Cyvis metric tool**

Cyvis metric tool has been used for the project ‘Catero’ to calculate the complexities of the web application.

The following document contains the report generated by Cyvis metric tool which consists tables of the following:

* Total number of classes, total number of methods in each class and in whole project.
* Total number of lines that each class consists of.
* Total cyclomatic Complexities for each method in a class.
* Graphical representation of above data.

**CyVis Output**

controller,AddProduct,doPost,15,290,,doGet,3,12,,<init>,1,14,

controller,Forgetpassword,doGet,5,162,,doPost,3,8,,<init>,1,6,

controller,OrderMail,doGet,4,141,,doPost,3,8,,<init>,1,6,

controller,AddToCart,doGet,6,118,,doPost,3,8,,<init>,1,6,

controller,ShowCart,doGet,5,96,,doPost,3,8,,<init>,1,6,

controller,CustomerRegistration,doPost,4,90,,doGet,3,12,,<init>,1,6,

controller,CheckEmail,checkEmail,4,71,,doPost,3,21,,doGet,3,8,,<init>,1,6,

controller,CustomerLogin,doPost,5,81,,doGet,3,12,,<init>,1,6,

controller,DeleteProduct,doGet,5,82,,doPost,3,8,,<init>,1,6,

controller,PlaceOrder,doGet,4,81,,doPost,3,8,,<init>,1,6,

controller,RemoveFromCart,doGet,6,71,,doPost,3,8,,<init>,1,6,

controller,AdminLogin,doPost,4,61,,doGet,3,12,,<init>,1,6,

controller,OrderPlaced,doGet,4,48,,doPost,3,8,,<init>,1,6,

controller,ViewProductDescription,doPost,4,40,,doGet,3,12,,<init>,1,6,

controller,CustomerGmailLogin,doGet,3,42,,doPost,3,8,,<init>,1,6,

controller,InvalidateJava,doGet,3,35,,doPost,3,8,,<init>,1,6,

controller,ViewProduct,doGet,3,24,,doPost,3,8,,<init>,1,6,

controller,ViewAllRegisteredCustomer,doGet,3,24,,doPost,3,8,,<init>,1,6,

controller,ShowProduct,doGet,3,24,,doPost,3,8,,<init>,1,6,

**Diagramatic Representation:**

