IIT

INSTITUTE OF INFORMATION TECHNOLOGY

Noakhali Science & Technology University

# Course name: Software Metrics Lab

# Course code: SE 3204

| Submitted to | Submitted by |
|---|---|
| Dipok Chandra Das | Joy Bhowmik(Ash1925012M) |
| Assistant Professor | Nayeem Khan(Ash1925027M) |
| IIT, NSTU | Redwan Hossain(Ash1925005M) |

Team members:

- Joy Bhowmik (ASH1925012M)
- Nayem Khan (ASH1925027M)
- Redwan Hossain Polash (ASH1925007M)

Project name: Digital Voting System

Project link: https://github.com/Joy-extreme/Face-recognition-voting/tree/master

## Object Oriented Design

**Weighted Method per Class (WMC):** It is a software metric used to measure the complexity of a class in an object-oriented software system. It is a measure of the total number of methods in a class, weighted by their complexity.

A high WMC value indicates that a class has many methods, and these methods are complex, making it more difficult to understand, maintain, and modify. On the other hand, a low WMC value indicates that a class has fewer methods, and these methods are less complex, which can lead to simpler, more manageable code.

Table 1: Table of weight giving convention to methods

| **Method complexity category** | **Weights** |
|---|---|
| Simple | 3 |
| Medium | 4 |
| Complex | 5 |

Table 2: Table of finding the WMC

| Class name | Methods | Weights |
|---|---|---|
| WebcameCapture.java | web() | 4 |
| CropppingOfImage.java | components () | 4 |
| FaceSimillarityPercentage.java | Percentage_checking(int[][]mat1,int[][]mat2) | 4 |
| HistogramEqui.java | components  (String location) | 4 |
| ImageLoad.java | work() | 5 |
| ImageToPixel | Image_to_pixel_method()<br>convertToArrayLocation(BufferedImage inputImage) | 4<br>5 |

| | | |
|---|---|---|
| | lbp_pixel(int [][]img,int x,int y)<br>get_pixel(int [][]img,int center,int x,int y) | 4<br>4 |
| AdminMainClass.java | main(String[]args) | 5 |
| AdminPanel.java | components() | 5 |
| EndingPageOfVoter.java | components() | 5 |
| PasswordCheck.java | components() | 4 |
| ResultPanel.java | components() | 5 |
| StartingPageOfVoter.java | components() | 5 |
| UserMainClass.java | main(String[]args) | 5 |
| UserPanel.java | components() | 5 |
| VoterPanel.java | components() | 5 |
| WelcomePage.java | components() | 5 |
| | **Total =** | **87** |

Weighted method per class = (total weighted method / No of classes )

= ( 87 / 16) = 5.4375

**Depth of Inheritance Tree (DIT):** It is a software metric used to measure the depth of the inheritance hierarchy for a given class in an object-oriented software system. It is a measure of the number of super classes that a class has, either directly or indirectly. A high DIT value indicates that the class has a long inheritance chain, meaning that it inherits many properties and methods from its super classes.

Table 3: Table of determining the DIT

| **Class Name** | **Value of DIT** |
|---|---|
| WebcameCapture.java | 0 |
| CropppingOfImage.java | 0 |
| FaceSimillarityPercentage.java | 0 |
| HistogramEqui.java | 0 |
| ImageLoad.java | 0 |
| ImageToPixel | 0 |
| AdminMainClass.java | 0 |
| AdminPanel.java | 0 |
| EndingPageOfVoter.java | 0 |
| PasswordCheck.java | 0 |
| ResultPanel.java | 0 |
| StartingPageOfVoter.java | 0 |
| UserMainClass.java | 0 |
| UserPanel.java | 0 |
| VoterPanel.java | 0 |
| WelcomePage.java | 0 |

**Number of Children (NOC):** It is a software metric used to measure the number of immediate subclasses that inherit from a particular class in an object-oriented software system. In other words, it measures the number of classes that are directly derived from a given class.

A high NOC value indicates that a class has many direct subclasses, which can make it more complex and difficult to understand, maintain, and modify.

Table 4: Table of determining the NOC

| Class Name | Value of NOC |
|---|---|
| WebcameCapture.java | 0 |
| CroppplingOfImage.java | 0 |
| FaceSimillarityPercentage.java | 0 |
| HistogramEqui.java | 0 |
| ImageLoad.java | 0 |
| ImageToPixel | 0 |
| AdminMainClass.java | 0 |
| AdminPanel.java | 0 |
| EndingPageOfVoter.java | 0 |
| PasswordCheck.java | 0 |
| ResultPanel.java | 0 |
| StartingPageOfVoter.java | 0 |
| UserMainClass.java | 0 |
| UserPanel.java | 0 |
| VoterPanel.java | 0 |
| WelcomePage.java | 0 |

**Coupling Between Object Classes (CBO):** It is a measure of how tightly coupled a group of objects or classes are to each other. It refers to the level of dependence between two or more objects, and the extent to which changes in one object require changes in the other.

High coupling means that objects are highly dependent on each other and that a change in one object is likely to affect many other objects in the system. On the other hand, low coupling means that objects are relatively independent of each other, and a change in one object is unlikely to affect other objects in the system.

Table 5: Table for finding CBO

| Class Name | Value of CBO | Names of coupled classes |
|---|---|---|
| WebcameCapture.java | 0 | |
| CropppingOfImage.java | 0 | |
| FaceSimillarityPercentage.java | 0 | |
| HistogramEqui.java | 0 | |

| | | |
|---|---|---|
| ImageLoad.java | 8 | WebcameCapture, CropppingOfImage, HistogramEqui, ImageToPixel, FaceSimillarityPercentage |
| ImageToPixel | 0 | |
| AdminMainClass.java | 0 | |
| AdminPanel.java | 0 | |
| EndingPageOfVoter.java | 0 | |
| PasswordCheck.java | 0 | |
| ResultPanel.java | 0 | |
| StartingPageOfVoter.java | 0 | |
| UserMainClass.java | 0 | |
| UserPanel.java | 1 | ImageLoad |
| VoterPanel.java | 0 | |
| WelcomePage.java | 0 | |

CBO per class = (8 / 16) = 0.5

**Response For a Class (RFC):** It is a software metric used to measure the number of unique methods that can be executed in response to a message received by an object of a class. In simpler terms, it measures the number of methods that can be invoked by an object in response to a message.

The higher the value of RFC, the more complex the class is, and the more potential paths of execution it has.

Table 6: Table for determining RFC

| Class Name | Value of RFC | Local methods + Invoked methods |
|---|---|---|
| WebcameCapture.java | 1 | web() |
| CropppingOfImage.java | 1 | components () |
| FaceSimillarityPercentage.java | 1 | Percentage_checking(int[][]mat1,int[][]mat2) |
| HistogramEqui.java | 1 | components  (String location) |
| ImageLoad.java | 9 | work()<br>components()<br>Image_to_pixel_method()<br>components()<br>components()<br>Percentage_checking(int[][]mat1,int[][]mat2) |
| ImageToPixel | 4 | Image_to_pixel_method()<br>convertToArrayLocation(BufferedImage inputImage)<br>lbp_pixel(int [][]img,int x,int y)<br>get_pixel(int [][]img,int center,int x,int y) |
| AdminMainClass.java | 1 | main(String[]args) |
| AdminPanel.java | 1 | components() |

| | | |
|---|---|---|
| EndingPageOfVoter.java | 1 | components() |
| PasswordCheck.java | 1 | components() |
| ResultPanel.java | 1 | components() |
| StartingPageOfVoter.java | 1 | components() |
| UserMainClass.java | 1 | main(String[]args) |
| UserPanel.java | 2 | components()<br>work() |
| VoterPanel.java | 1 | components() |
| WelcomePage.java | 1 | components() |

**Message Passing Coupling (MPC):** It is a software metric used to measure the degree of coupling between objects in an object-oriented software system. It measures the number of messages that are passed between objects in a class, indicating the extent to which objects are dependent on one another to perform their respective functions.

High MPC values indicate that there are many messages being passed between objects, which can result in a tightly coupled system that is difficult to maintain, modify, and test.

Table 6: Table for determining MPC

| **Class Name** | **Value of MPC** | **Invoked methods** |
|---|---|---|
| WebcameCapture.java | 0 | |
| CroppingOfImage.java | 0 | |
| FaceSimillarityPercentage.java | 0 | |
| HistogramEqui.java | 0 | |
| ImageLoad.java | 8 | components()<br>Image_to_pixel_method()<br>components()<br>components()<br>Percentage_checking(int[][]mat1,int[][]mat2) |
| ImageToPixel | 0 | |
| AdminMainClass.java | 0 | |
| AdminPanel.java | 0 | |
| EndingPageOfVoter.java | 0 | |
| PasswordCheck.java | 0 | |
| ResultPanel.java | 0 | |
| StartingPageOfVoter.java | 0 | |
| UserMainClass.java | 0 | |
| UserPanel.java | 1 | work() |
| VoterPanel.java | 0 | |

| | | |
|---|---|---|
| WelcomePage.java | 0 | |