**1.Abstraction**

**Code Complexity Tool** Group ID=28

Producing a high-quality software product economically is one of the main fundamental objectives of every software development practice. Moreover, code quality plays a prominent role when talking about the quality of the software. Considering the above matters, measuring the code complexity of a specific file is important to software developers particularly in maintaining the quality of the product. Basing on this factor, a web application was developed targeting the measuring of code complexity and PHP language was employed as the technological base of the product followed by techniques such as HTML, CSS, jQuery, js, and MySQL. This web application can be utilized to measure the complexity of a single or multiple JAVA file(s), zip file or a folder. This paper presents the web application and discusses the six-software metrics; namely size, variables, methods, inheritance, coupling and control structures which were combined together to complete this product.

Key words—code Complexity, Size, Variables, Methods, Inheritance, Coupling, Control Structures

**2.Introduction**

Codes are harder to change if it is difficult to read and understand. As software developer wants easiest way to do it.so we think about make a new web-based application to fix it. We created a tool called the code complexity measurement tool as our project.

Software complexity could be defined as the principle driver of cost, reliability, and performance of software. Software complexity is the level of challenge in analyzing, maintaining, texting, designing and modifying software in other words, , software complexity is an issue that is in the whole software development process and each phase of software development life cycle (SDLC).

We only concentrate on developing C++ and Java based web application. And we used PHP language for develop our web application.

Why we developed web-based application? Because we can easily update, and we will able to develop this application for other languages. And, user no need of update. When this application standalone system user wants to update.

We measure the complexity introduced due to following factors,

* Size, Variables and Methods
* Inheritance
* Coupling
* Control structures

There have been many code complexities measuring tools proposed by software developers accompanied with several case studies which indicates how the proposed system relate to the quality of the software, since the early stages of software. McCabe’s cyclomatic complexity is one of the well-known complexity measures of all time among other metrics. And other metrics tools are Halstead's software metric, Klemola’s KLCID complexity Metric, Wang’s cognitive functional complexity.

A major force behind these efforts is to increment our capability to predict the effort, quality, coding efficiency, cost or all of these.

**3.RELATED WORK/LITERATURE REVIEW**

When we find details about Software complexity tool, we have understood, this application is deployed in wide area. We divided our functions with each person. We divided our project to four parts. Those are

a. Size, variables, and methods

b. Inheritance

c. Coupling

d. Control structures

Firstly, we selected a language for doing this project. We selected php web-based application to do this project. Because our group members have a good knowledge about PHP.

We choose sublime text for code editor. It is a clean, functional, and fast code editor. Not only does it have incredible built in features, but it has support for many plugins, snippets, and many other things. It has lots of shortcuts also. It was a lot easier for our work.

We used WAMP server. It is a Web development platform on Windows that allows you to create dynamic Web applications. WAMP Stands for Windows, Apache, MySQL, and PHP. It is a variation of LAMP for Windows systems and is often installed as a software bundle. We used it because we already used it very well for our previous projects and we had finished our works very successfully by doing in it. We used it for web development and internal testing. It automatically installs everything you need to intuitively develop Web applications. You will be able to tune your server without even touching its setting files.

And we used GITHUB to integrate our application. Github is a web-based platform used for version control. It simplifies the process of working with other team members and makes it easy to collaborate on our application. Each team member can work on files and easily merge their changes in with the master branch of the project. In quarantine situation these days, GITHUB was the only saviour for us.

Those are the most important tools/systems that we used.

When we were doing this project, we had faced some problems and as a group we found the solutions and answers for problems.

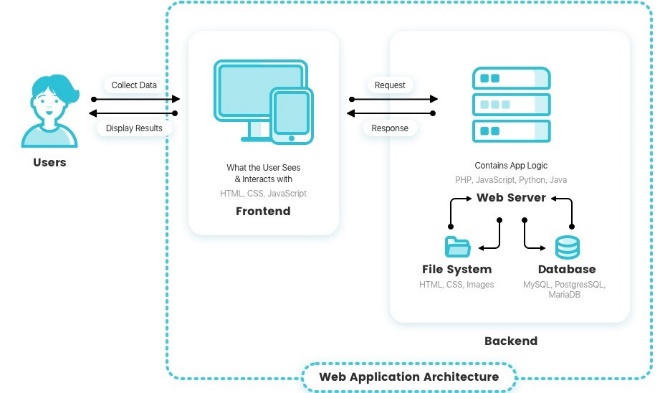
Size, variables, and methods part is one Function. It divides into three parts. We faced some problems when calculating some columns in sprint 1. But he could manage it ourselves. In sprint 2 also he had to work than others because he had three parts of his function. Inheritance part is another function. In sprint 1, all the other functions are same in printing the code line by line. But in inheritance part should print only the class names. It is the first problem, which we faced, when we were doing it. We managed it by searching the forums and IT related sites. Control Structure part is the next function. We also faced some problems when calculating some columns. Coupling part is the major problem that we faced in this project. In sprint 1 we had lots of columns to calculate. We also could not understand some calculations in this part. We asked other groups also. Every group had the same issues in this part. After a lot of effort, we did that activity with the help of the lecturers and senior students.

After all we hosted our web-based application at a hosting platform.

**4.Proposed System**

**4.1. Introduction**

Our System is a dynamic web application. Simply Web application is an application that runs through web browsers like Google Chrome, Firefox, Internet Explorer. The program runs on the localhost in its pc using wamp server or xamp server. Also, we can post our web application onto the Internet using Web hosting. Then our Web application is stored, on special high-performance computers called servers. When Internet users want to view your website, all simply they have to type web application address into their browser.

Client computers will then connect to the hosted server and our web application will be loaded in their browsers. There are two main categories of programming for creating Web Applications. There are client side(frontend) and server side(backend). Both client and server-side programming are necessary to make a website. Client-Side(Frontend) functions run on the client machine (browser) and deals with the user interface/display and any other processing that can happen on client machine like reading/writing cookies simply client-side collects user inputs and mostly deal with user input aspects.

Html, CSS, bootstrap, JavaScript, jquery,datatable are used to develop and implement this web application. HTML (Hypertext Mark-up Language) is a mark-up language. The structure of Web pages describes using HTML.HTML elements are represented by its pre-declared tags like head, body, title Browsers do not display the HTML tags. CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on the screen.CSS is a powerful technology to style web pages. There are Inline and External CSS. External stylesheets are stored in.CSS files.

There are 3 types of JavaScript namely they are Client-Side JavaScript (CSJS), server-Side JavaScript (SSJS), Core JavaScript. In client-side uses, CSJS and core js.Js Runs on the user's computer and Source code visible to the user. Used for validation and functionality for user events such as button clicks, link navigation. Also Depends on browser and version.

Bootstrap, the world’s most popular framework for building responsive, quickly produce responsive, clean looking websites. t does this simply by providing you with a ton of pre-built CSS and JavaScript. **It Saves lots of time** and **Easy to use** with **Responsive features and Consistent design. Also Compatible with browsers** and **Open Source Framework.** The objective of jQuery is to make it much easier and faster to use JavaScript. It is a library that is light-weight and very fast. jQuery will make coding very faster and also result in simple and clean code.

Data Tables **is a** powerful JavaScript library for adding interaction features to HTML tables like generation reports, CSV,excel sheets, and pagination, and search options. Data Tables' huge range of options **will be accustomed**customize the way that **it’ll** present its interface, **and therefore the** features available, to **the** user. **this can be** done through its configuration options, which are set at initialization time.





**4.2) Method**

The main objective is of this web-based code complexity measuring tool is to reduce the maintenance cost of the software. This software uses the lines of code to evaluate the complexity Using this web application can measure the complexity following factors:

a. Size, variables, and methods

b. Inheritance

c. Coupling

d. Control structures

e. Total Factor

In this application, first of all, the user has to upload file or files which want to calculate. Then it files lines are traversed in foreach loop. If we consider Size, Variable, and methods factor we stored keywords, identifiers, operators, a string literal, and check each line that has a numerical value and gets count it. And Also, primitive data type and Composite data types are stored in separate arrays.

**4.3) Tools**

To implement this web application, we used the Visual Studio Code and Sublime Text AS IDE’S.

Visual Studio Code

Visual Studio Code could be a streamlined code editor with support for development operations like version control and debugging. Visual Studio Code features a lightning-fast ASCII text file editor, perfect for day-to-day use. With support for many languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more

Sublime Text

Sublime Text also has tons of plugins you can find through Package Control. But it's only a text editor and not an IDE.

**4.4) Flowcharts**

**4.4.1. Flowchart for Upload Files**

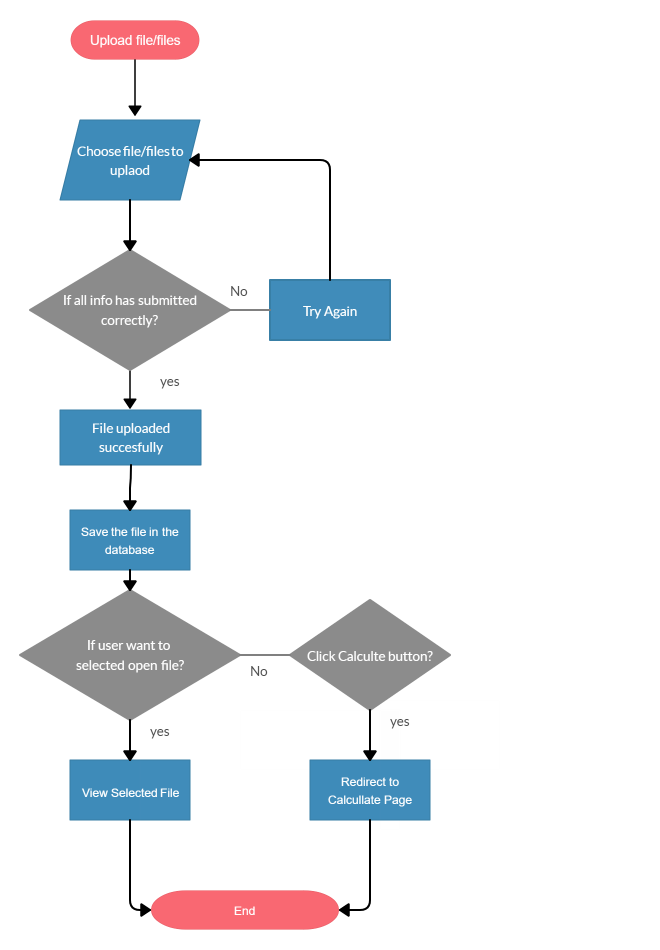


Figure:4.4.1

**4.4.2. Flowchart for edit weight values**

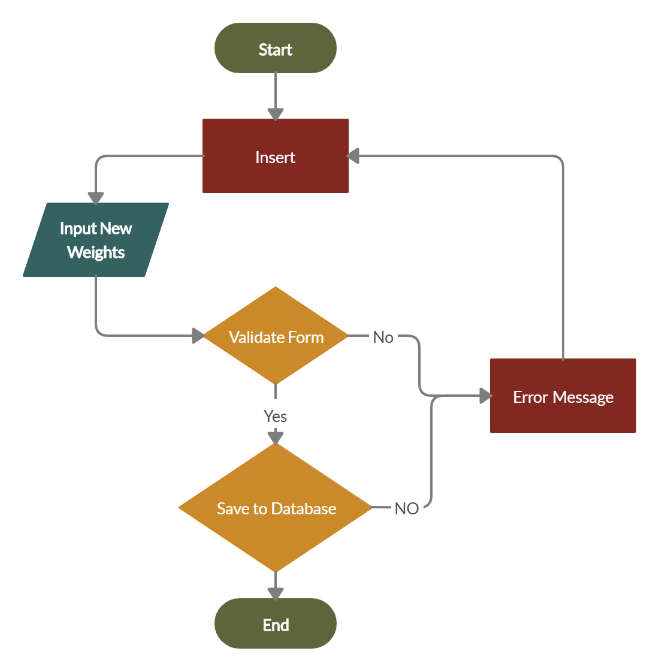


Figure:4.4.2

**4.4.3. Flowchart for Size, variables, and methods Calculation**

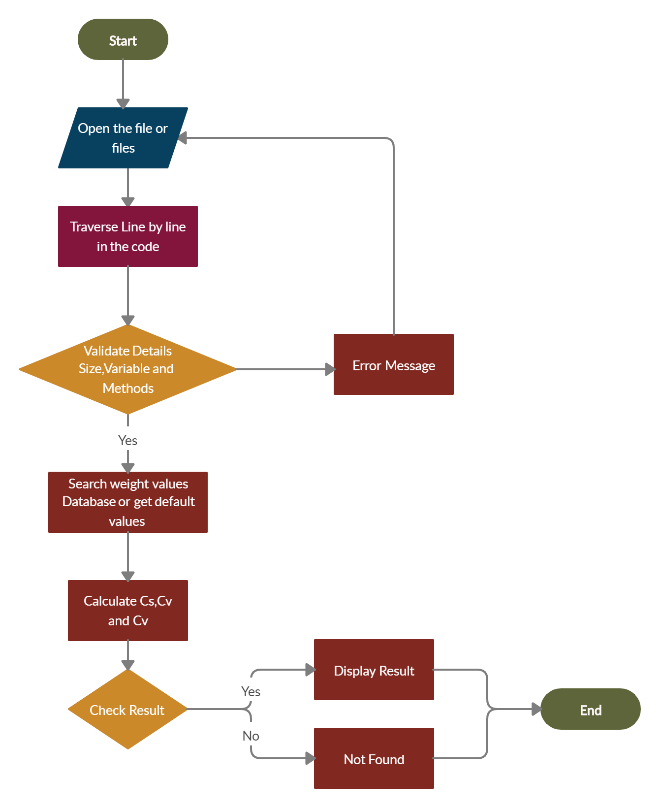
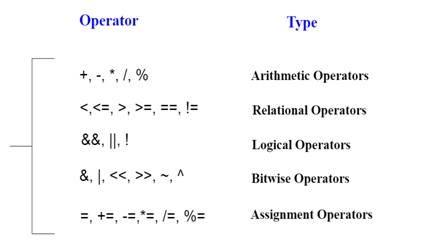


Figure:4.4.3

According to these flowcharts first of all users should upload files or files. Then using for each loop Line by line traverse in the whole file. All the logic to calculate complexity in this loop. When we consider size factor

* Keywords
* Identifiers
* Operators (Arithmetic operators, Relation operators, Logical operators, Bitwise operators, Miscellaneous operators, Assignment operators)
* Numerical Values
* String Literal



are stored in arrays. After validating details then search weight values user entered (last row fetched) or get a default value. Then Calculate Cs, Cv values. After checking the result, the result is displayed in a table with related program lines.

**Cs = (Wkw \* Nkw) + (Wid \* Nid) + (Wop \* Nop) + (W n v \* Nnv) + (Wsl \* Nsl)**

Complexity of a program statement due to its size -Cs

In programming there are two type of variables based on the scope

1. Global variables - Variables which are declared/defined inside a class, but outside a method.

2. Local variables - Variables declared/defined in a method or constructor.

A weight of two is assigned for each global variable and a weight of one is assigned for each local variable as default values.

Based on the data type, variables are divided into two types

1. Primitive data type variables

2. Composite data type variables

A weight of two is assigned for each Composite data type variables and a weight of one is assigned for each local variable as Primitive data type variables.

**Cv = Wvs [(Wpdtv \* Npdtv) + (Wcdtv \* Ncdtv)]**

Method return types are divided into two categories as follows:

1.Primitive return type a weight of one

2. Composite return type a weight of two

**Cm = Wmrt + (Wpdtp \* Npdtp) + (Wcdtp \* Ncdtp)**

**4.4.4. Flowchart for Inheritance**

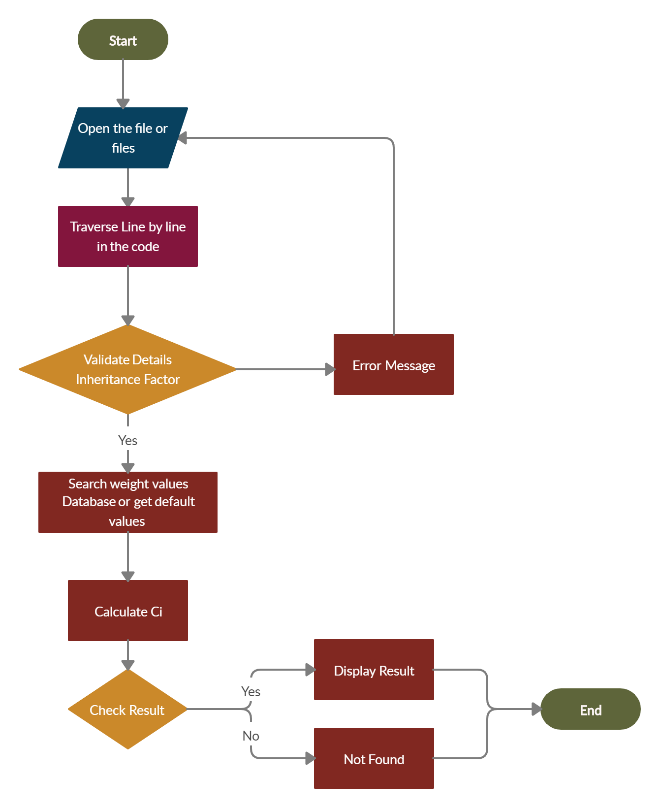


Figure:4.4.4

Inheritance is an important aspect of OOP (Object Oriented Programming). It is the mechanism in which one class is allow to inherit the features (properties and methods) of another class. Inheritance represents the **IS-A relationship** which is also known as a parent-child relationship.

* **Class**
* **Super Class**
* **Sub Class**
* **Reusability**

The extends keyword indicates **that you just** are making **a new** class that derives from an existing class.

Inheritance complexity of a program statement of a class (Ci) =

Weight assigned for that class due to its inheritance pattern

**4.4.5. Flowchart for Coupling**

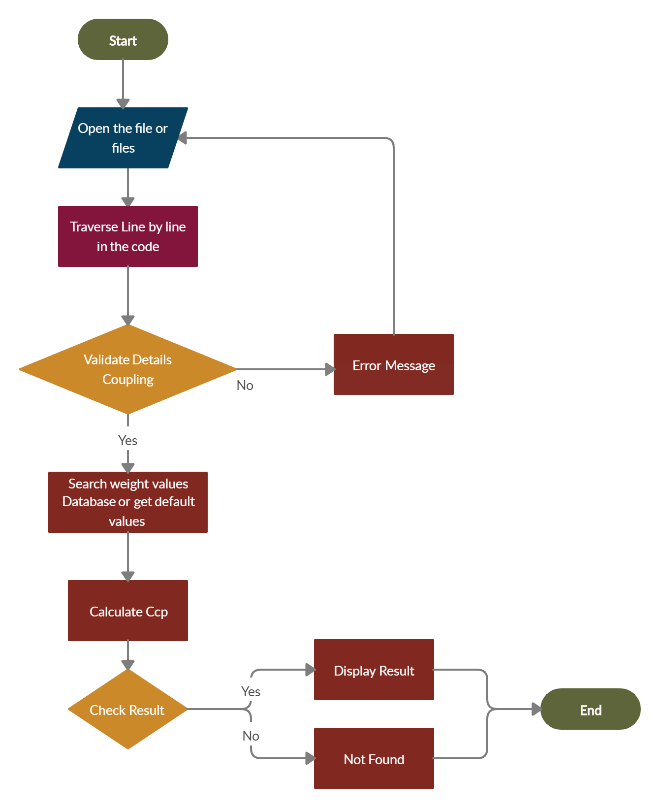


Figure:4.4.5

Coupling is that the principle of "separation of concerns". this implies that one object doesn't directly change or modify the state or behaviour of another object. In programming coupling is viewed as how much each program module depends on different modules and is additionally the term used to portray associating at least two frameworks.

|  |  |
| --- | --- |
| Inherited Pattern | Weight |
| A class without any inheritance (direct or indirect) | 0 |
| A class inheriting (directly or indirectly) from one user-defined class | 1 |
| A class inheriting (directly or indirectly) from two user-defined classes | 2 |
| A class inheriting (directly or indirectly) from three user-defined classes | 3 |
| A class inheriting (directly or indirectly) from more than three user-defined classes | 4 |

**Ccp =(Wr \* Nr) + (Wmcms \* Nmcms) + (Wmcmd \* Nmcmd) + (Wmcrms \* Nmcrms) + (Wmcrmd \* Nmcrmd) + (Wrmcrms \* Nrmcrms) + (Wrmcrmd \*Nrmcrmd) + (Wrmcms \* Nrmcms) + (Wrmcmd \* Nrmcmd) + (Wmrgvs \*Nmrgvs) + (Wmrgvd \* Nmrgvd) + (Wrmrgvs \* Nrmrgvs) + (Wrmrgvd \* Nrmrgvd)**

**4.4.6. Flowchart for Control Structures**

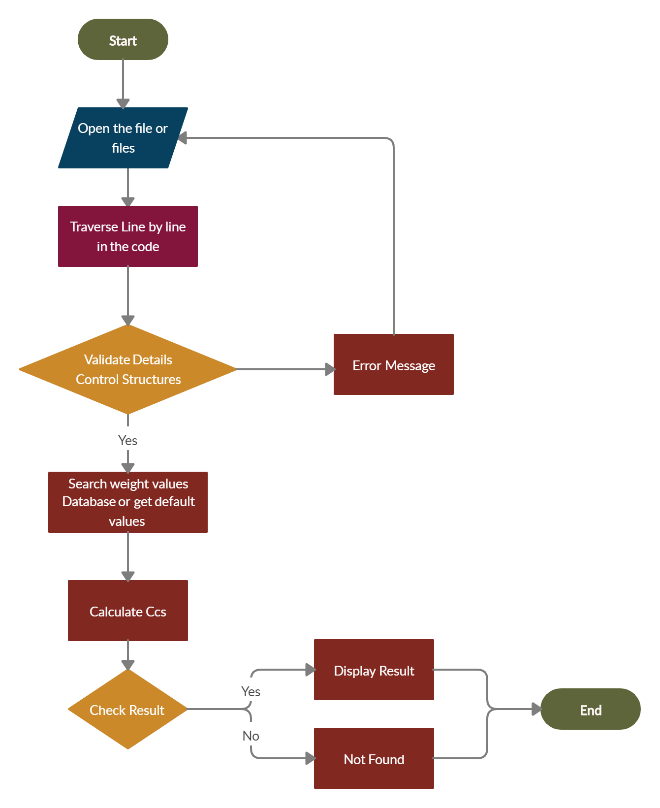


Figure:4.4.6

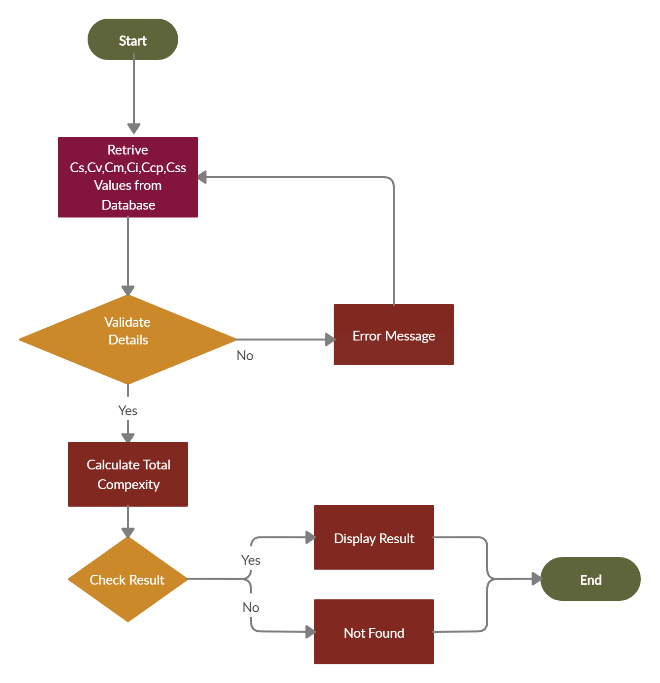
A control structure is sort of a block of programming that analyses variables and chooses a direction within which to travel supported given parameters. Mainly there have 3 fundamental control structures such as;

Figure:4.4.7

1. Sequence control structure.
2. Decision control structure.
3. Loop control structure.

|  |  |
| --- | --- |
| Control Structure Type | Weight |
| A conditional control structure such as an *if* or *else-if* condition | 2 |
| An iterative control structure such as a *for*, *while*, or *do-while* loop | 3 |
| The *switch* statement in a *switch-case* control structure | 2 |
| Each *case* statement in a *switch-case* control structure | 1 |

Ccs = (Wtcs \* NC) + Ccspps

**4.4.7. Flowchart for Total Factor**

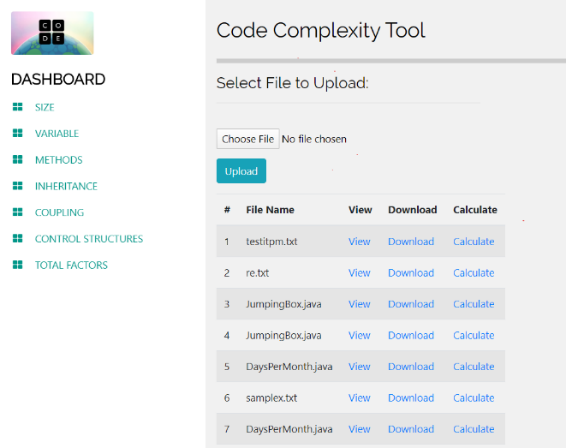
As the above the total complexity of a program also be measured by adding all the factors which are ;

* size.
* variable.
* method.
* inheritance.
* coupling.
* control structure.

TCps = Cs + Cv + Cm + Ci + Ccp + Ccs

The flows are for measuring the complexity of code due to size, variables, methods, inheritance, control structures coupling, and total complexity. According to these flowcharts first of all users should upload files or files. Then using for each loop Line by line traverse in the whole file. All the logic to calculate complexity in this loop. Keywords, Identifiers, Operators (Arithmetic operators, Relation operators, Logical operators, Bitwise operators, Miscellaneous operators, Assignment operators),, String Literal, Primitive data type variables, Composite data type variables are stored in arrays. After validating details then search weight values user entered (last row fetched) or get a default value. Then Calculate Cs, Cv values. After checking the result, the result is displayed in a table with related program lines.

**4.5. Creation of User Interfaces**

As mentioned in the proposed web application should has 10 main Interfaces. Those interfaces are Home (Index Page), Size-Variable-Methods, inheritance, coupling. Control structure and Total factors and Generating report and also, mainly every interface include a calculate button and file upload button which user want to upload files which want to calculate code complexity. Also, there is an option to change weight in Size-Variable-Methods, inheritance, coupling. Control structure factors using the form. When users submit the form then click update weights button user can analyze code complexity using the latest updated weight. Then calculations are done by the latest updated values.

Index page/Home Page

Figure:4.5.1

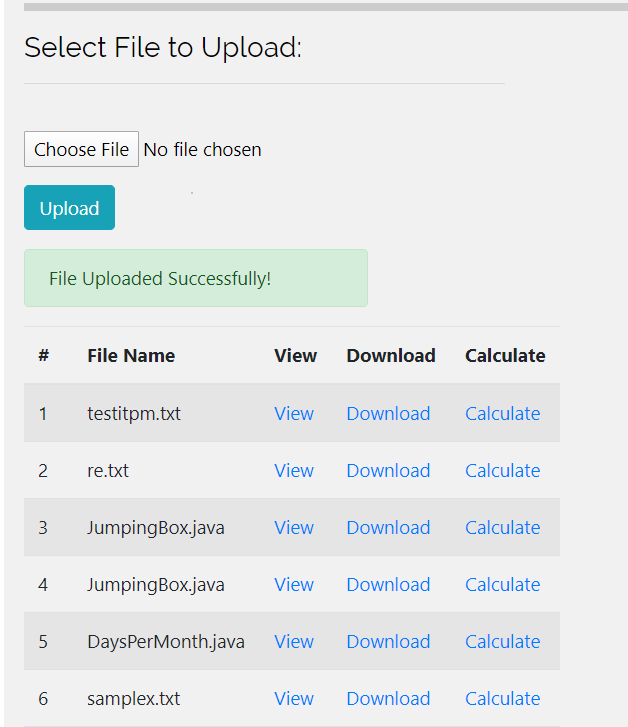


Figure:4.5.2

After Successful File Upload Multiple files are shown in the grid and file name and lines are stored in the database for retrieval in calculating total factors

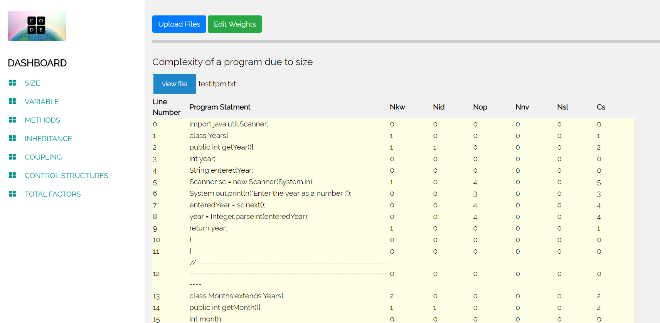


Figure:4.5.3

Size Calculation Interface

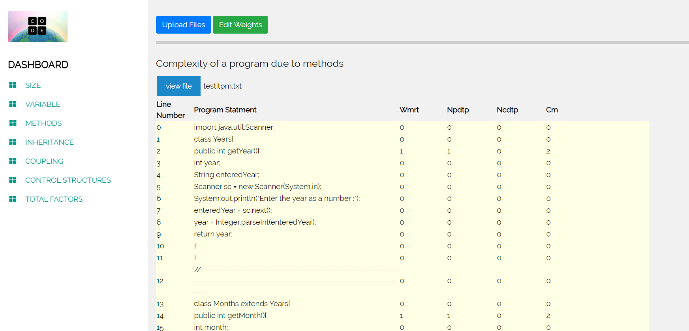
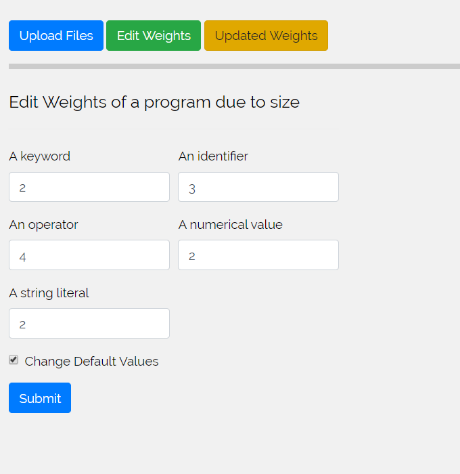
with default Weights

Figure:4.5.4



Variable Calculation Interface with default Weights

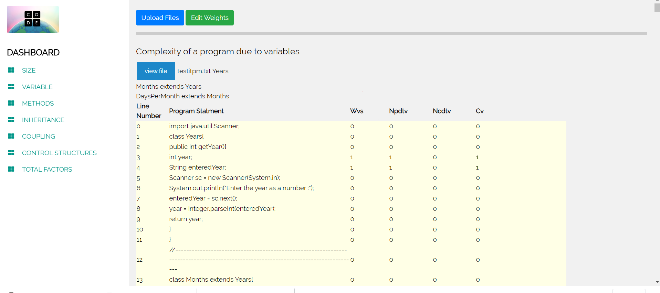


Figure:4.5.5

Figure:4.5.8 After Successful Submission of values.

Method Calculation Interface

with default Weights

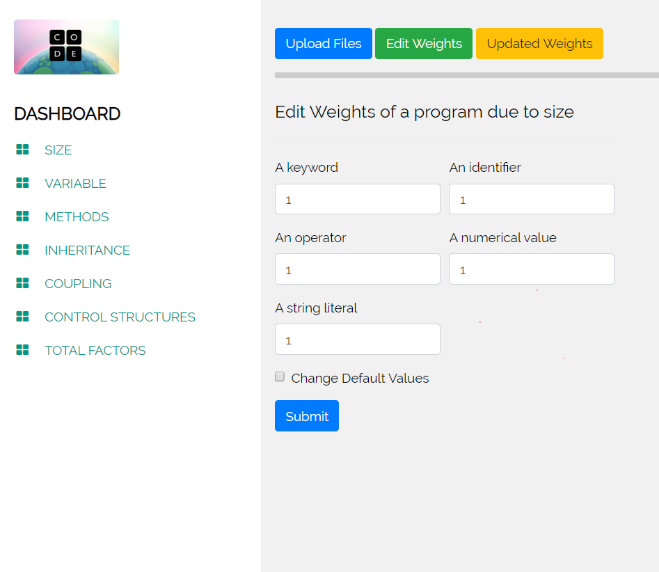
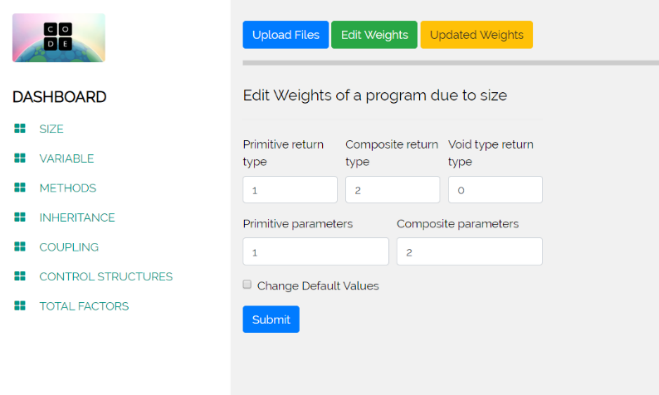
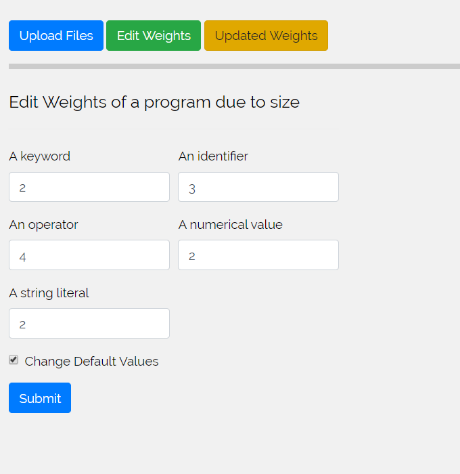
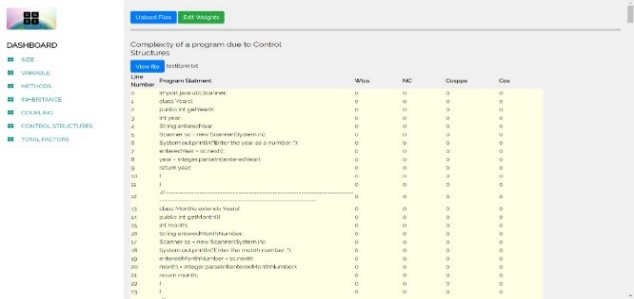
In these interface displays the calculation of each factor with default weight values. And by clicking edit weights user can update weights. And using view file button user can view the updated file. In updated file calculation are done by new added values to the database. It fetches last row of the table.

Figure:4.5.9

Figure:4.5.8 After Succesfull Submition of values.

Figure:4.5.7 Form to change weights



Control Structures Calculation Interface

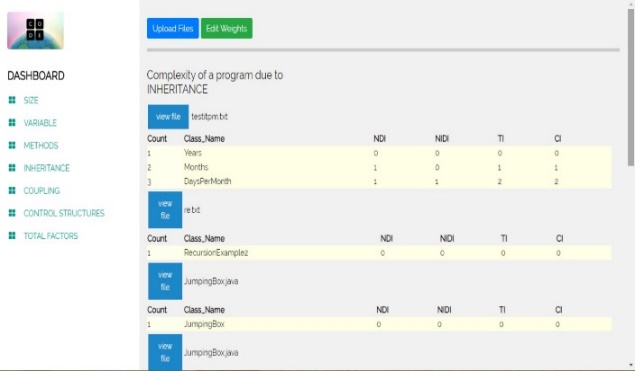
with default Weights

Figure:4.5.10

Inheritance Calculation Interface

with default Weights.

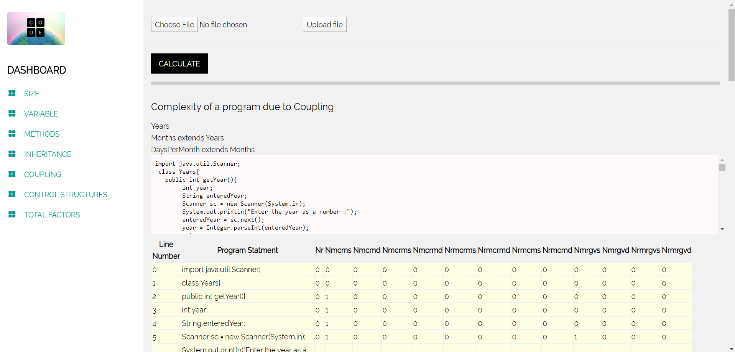


Figure:4.5.11

Coupling Calculation Interface

with default Weights

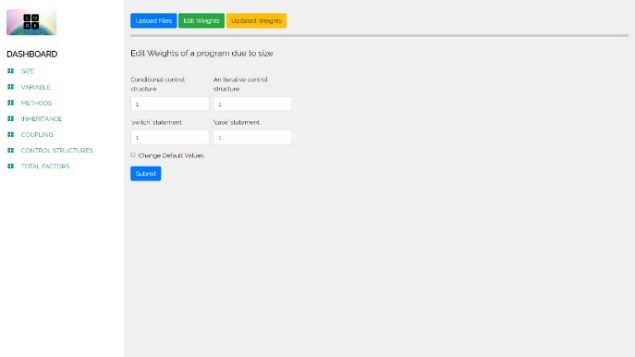
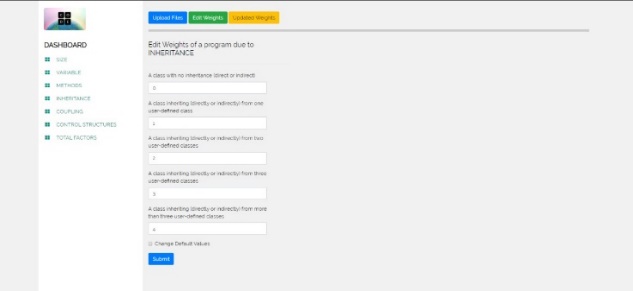
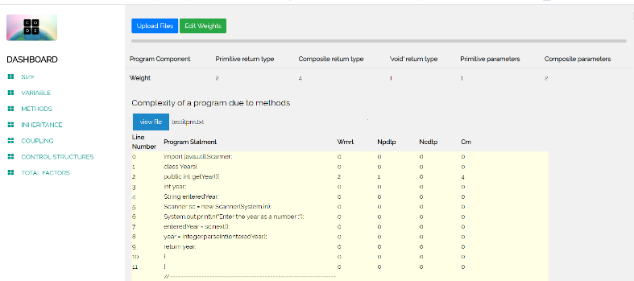


Figure:4.5.12

****



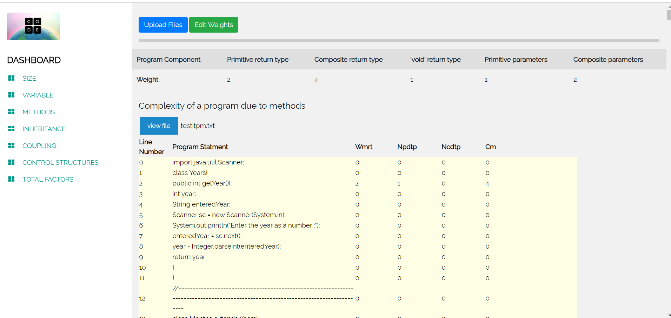
****

Figure:4.5.15 Interface with updated weights.

Figure:4.5.13

Forms To change weights

Forms to change weights

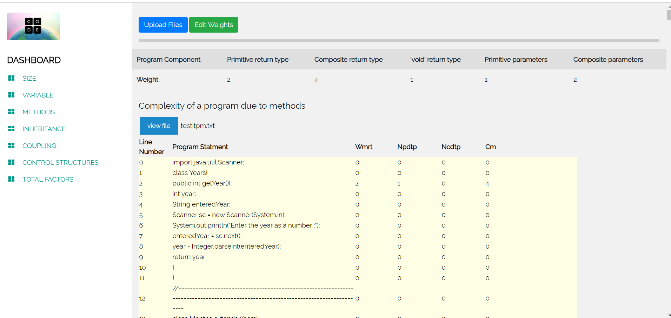


Figure:4.5.14 Interface with updated weights.

**5.Methodology**

**5.1) Method**

waterfall model is the software process model that we used to implement this software. This implementation is a progressive design process and in the development of the software step by step we were able to go efficiently and sometimes we have to iterate some steps, and go back to last step again. Our main developing language is PHP server side and also we worked with MySQL, js server side, server-Side JavaScript (SSJS).

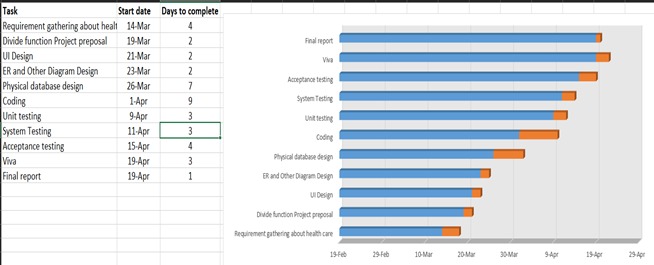
And here is our project grant chart, 

Figure:5.1 Gannt Chart

We choose to develop this software as a web application. Because then our product not base on a single person or company, anyone who needs this can use via internet. We thought that would helpful for those who needed. We always try to develop this software as a more efficient to the user and it will be more user friendly in many ways.

5.2) Tools

**5.2.1) integrated development environment (IDE)**

We choose sublime text for code editor. It is a clean, functional, and fast code editor. Not only does it have incredible built in features, but it has support for many plugins, snippets, and many other things. It has lots of shortcuts also. It was a lot easier for our work.

**5.2.2) WAMP server**

We used WAMP server. It is a Web development platform on Windows that allows you to create dynamic Web applications. WAMP Stands for Windows, Apache, MySQL, and PHP. It is a variation of LAMP for Windows systems and is often installed as a software bundle. We used it because we already used it very well for our previous projects and we had finished our works very successfully by doing in it. We used it for web development and internal testing. It automatically installs everything you need to intuitively develop Web applications. You will be able to tune your server without even touching its setting files.

**5.2.3) GITHUB** 

And we used GITHUB to integrate our application. GitHub is a web-based platform used for version control. It simplifies the process of working with other team members and makes it easy to collaborate on our application. Each team member can work on files and easily merge their changes in with the master branch of the project. In quarantine situation these days, GITHUB was the only savior for us.

**5.3) User Interfaces development**

We came up with total of 7 interfaces, Those interfaces are Home, Size-Variable-Methods (SVM), inheritance, coupling. Control structure and total factors. Each and

Every interface includes a submit and calculate buttons which are allow us to import a file or multiple files.

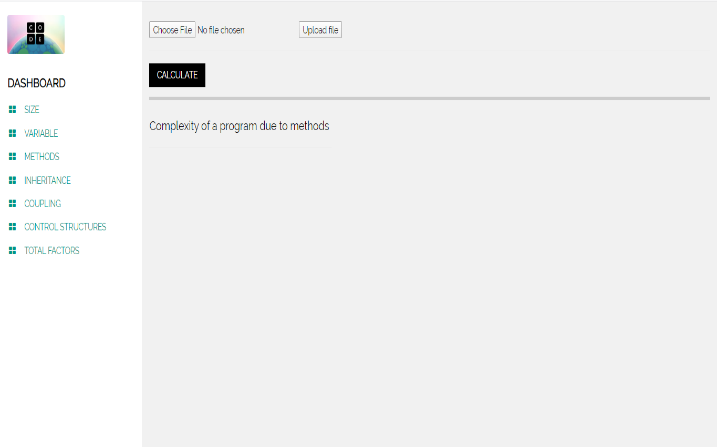


Figure:5.2 interface

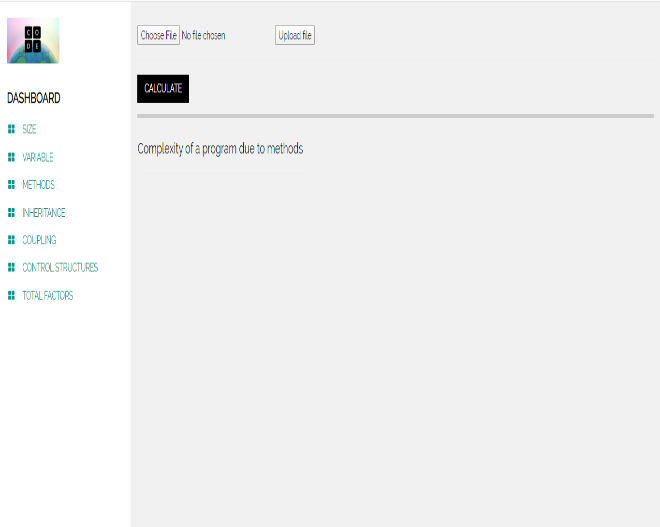
****

Figure:5.3 interface’s upload button and chose file button

Those interfaces are created by using HTML CSS, bootstrap, jQuery.

**5.4) Implementation of the backend.**

**5.4.1) Uploading a single file**

Basically, we developed this software to import a single file and then after the success of single file uploading, we update the system to import multiple files or folders or zip flies.

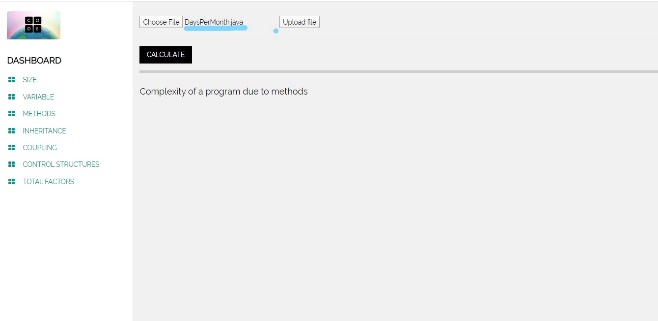


Figure:5.4.1(1) interface’s uploading single file

Code we implement to run above interface

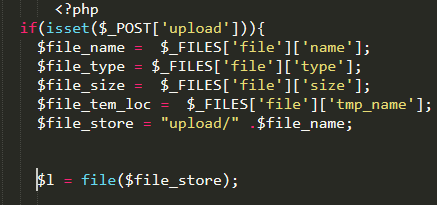


Figure:5.4.1(2) source code for uploading file

And also implanted code for multiple files below here

Figure:5.4.1(3) source code for uploading multiple file

After file(s) uploaded we have to show the content of the file(s) for that we use this code,

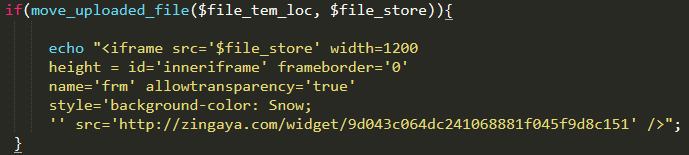


Figure:5.4.1(3) source code for display file content

This part was a one of the requirements that request to be in implementation during the project, as its mention in the project proposed specification. User can see what he/she upload into the system then check correct file or files are uploaded.

**5.4.2 Implementation**

5.4.2.1 The derivation of complexity due to each factor

First system should be getting the code as line by line and count it .so that to develop that part we *used a foreach loop,*

*foreach ($l as $line\_num => $line) {*

*}*

**5.4.2.1.1) Size**

In the backend we used foreach loops to travers Arrays. Words that need to be search firstly stored in a arrays as global variables, then in the body of code we travers and gets match words.

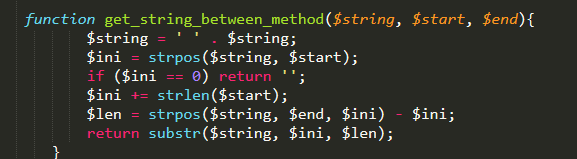
Here is the Arrays that are store for size



Figure:5.4.1(4) source code for display Arrays

To travers and get the result what we want we used another function called *get\_string\_between* ,by using this function we can get a word between two word that we insert.

Here is the function that we used.

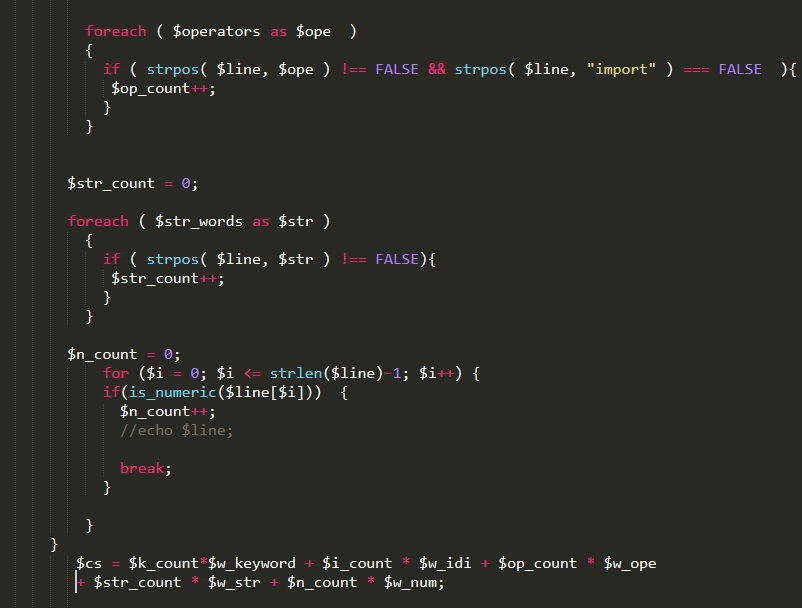


sf

Figure:5.4.1(5) source code of *get\_string\_between* function

Then the implementation of the size , we used that technologies that mention before.

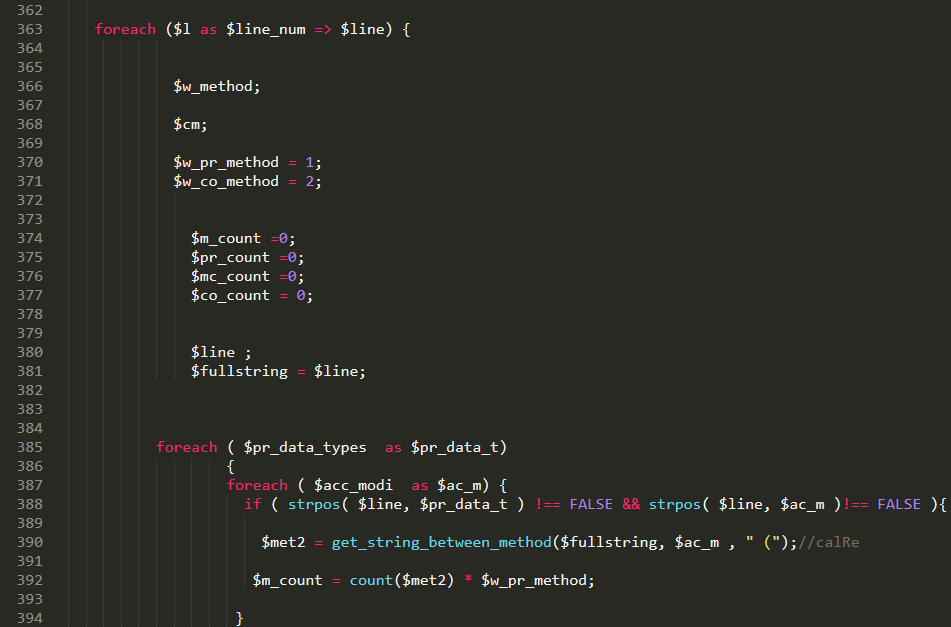


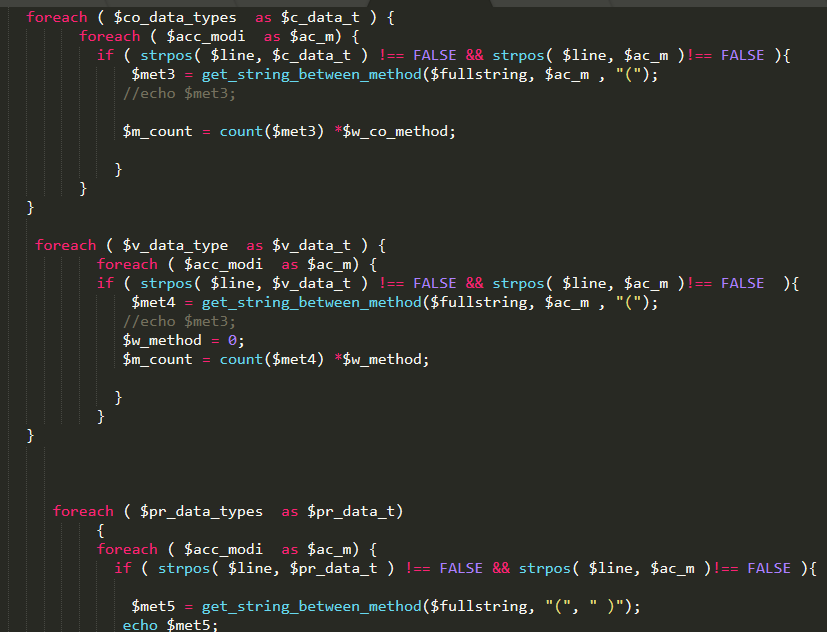
Figure:5.4.1(6) source code of *size* Figure:5.4.1(7) source code of *size*

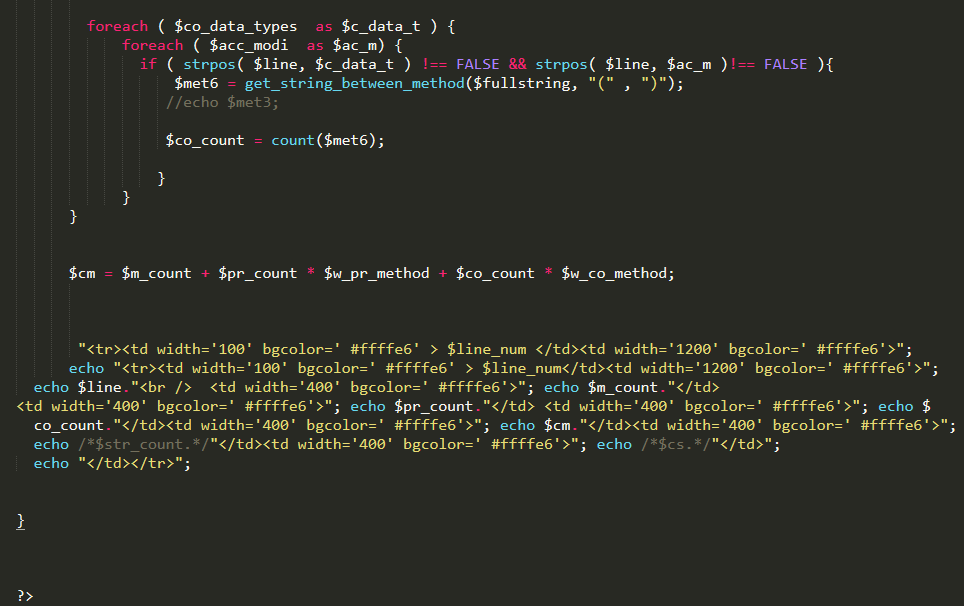
We used above techniques to implement all other function, Arrays foreach loop and the *get\_string\_between* function.

**5.4.2.1.2) Method**

This function developed for calculate methods in the given code ,this should identify the methods and voids are not calculate, and Components like primitive and composite data type parameters are considered and complexity is calculated with regular expressions, sorting the components traversing from line to line through the code.

Figure:5.4.1(8) source code of *method*

Figure:5.4.1(9) source code of *method*

Figure:5.4.1(10) source code of *method*

**5.4.2.1.2) Variable**

This function should identify the Like in size, in variables there are certain components to consider. Primitive and compositive data types and also Global and Local variables are taken into consideration applying the necessary weights to calculate the complexity.

In this function also used *get\_string\_between* function. To get similar words

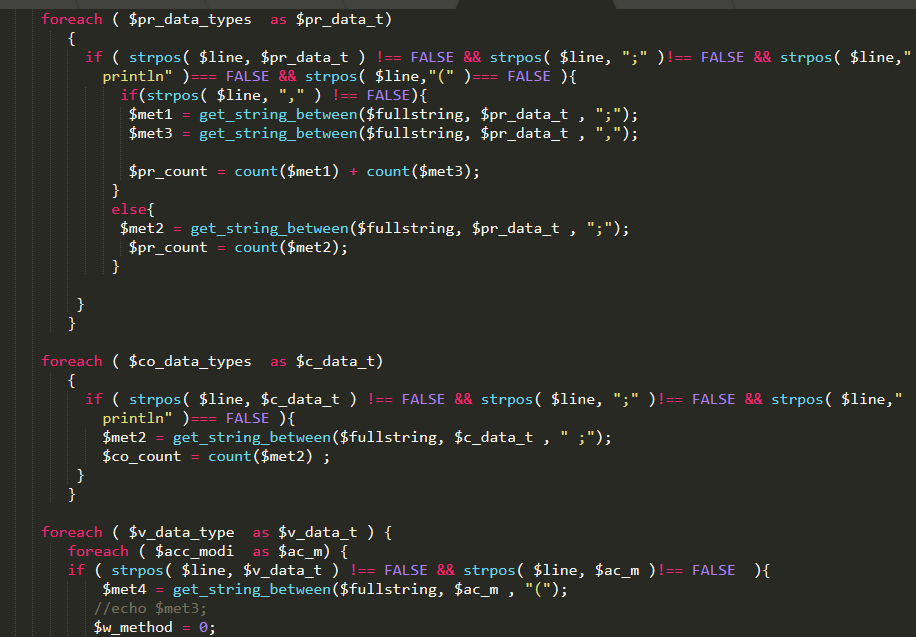
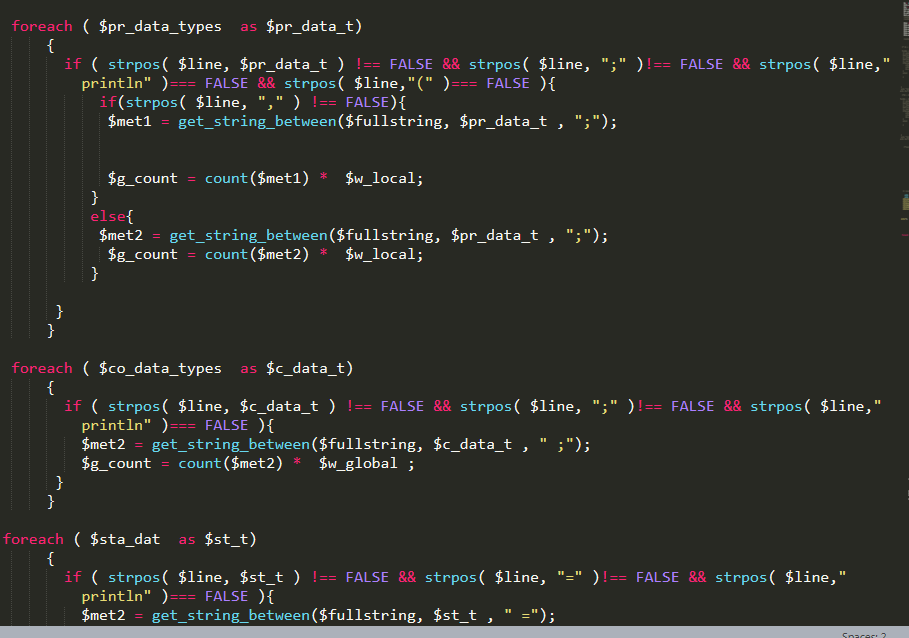


Figure:5.4.1(11) source code of *variable*



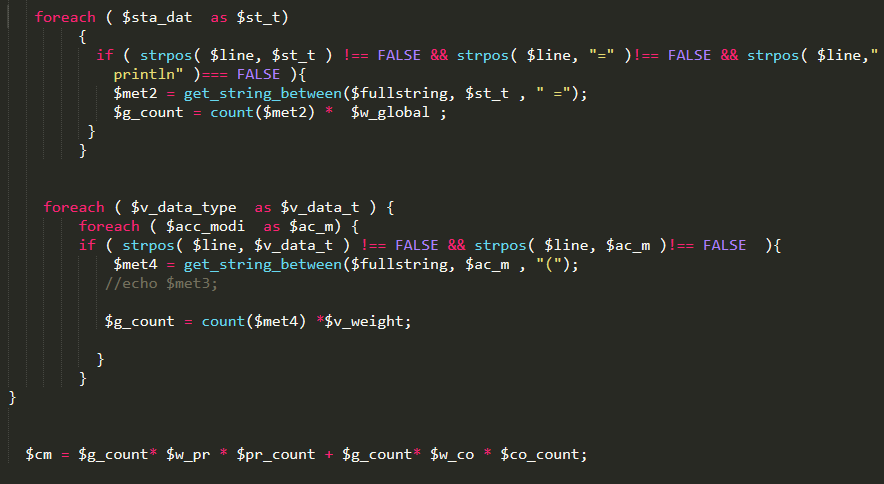


Figure:5.4.1(12) source code of *variable*



Figure:5.4.1(13) source code of *variable*

**5.4.2.1.3) Inheritance**

In this function user can calculate, have a idea about classes that are user -defined classes, whether direct or indirect. For this, classes are sorted through lines, checking the inherited number of user-defined classes and the necessary weights are allocated for the measurement of complexity. This is also calculated with applications of regular expressions due to its inheritance pattern.



Figure:5.4.1(14) source code of *inheritance*

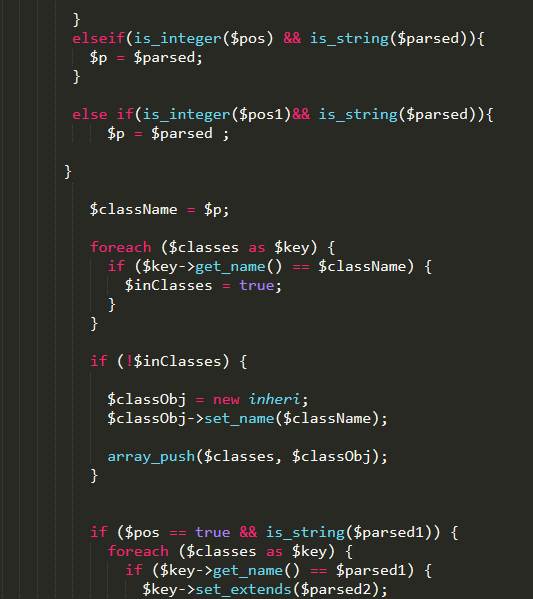


Figure:5.4.1(15) source code of *inheritance*

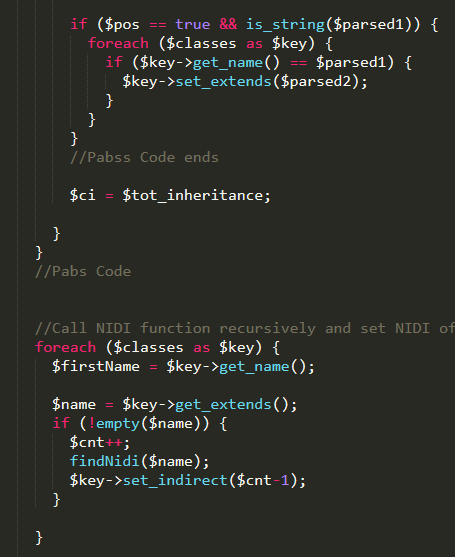


Figure:5.4.1(16) source code of *inheritance*

**5.4.2.1.3) Coupling**

This function develops to Complexity measure the recursive calls methods, and regular methods and global variable calls in same and different files. With regular expressions, these components are sorted and filtered into the table with the weights allocated for each component.

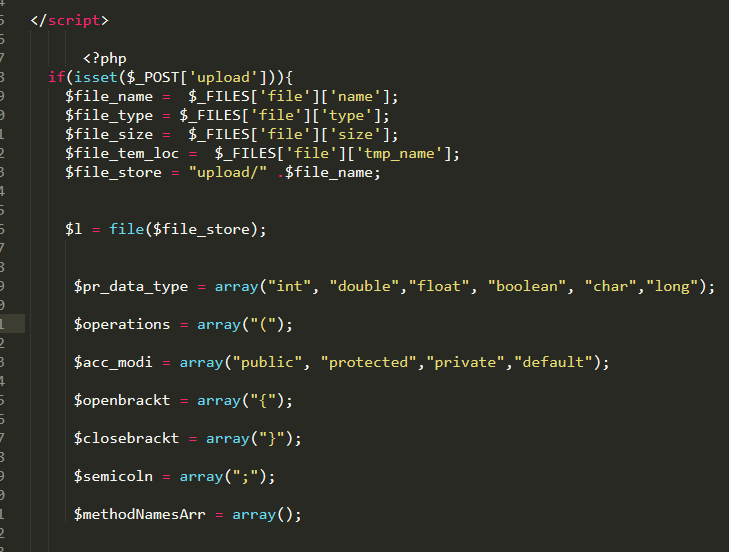


Figure:5.4.1(17) source code of *coupling*



Figure:5.4.1(18) source code of *coupling*

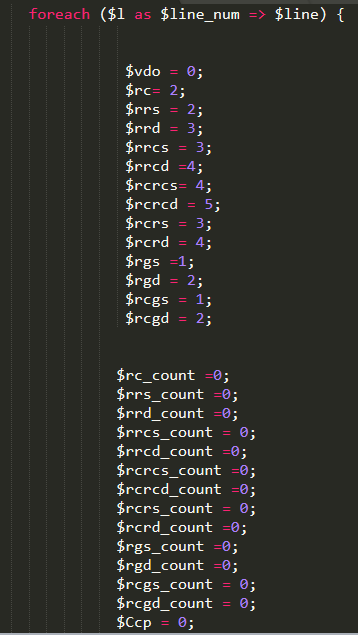


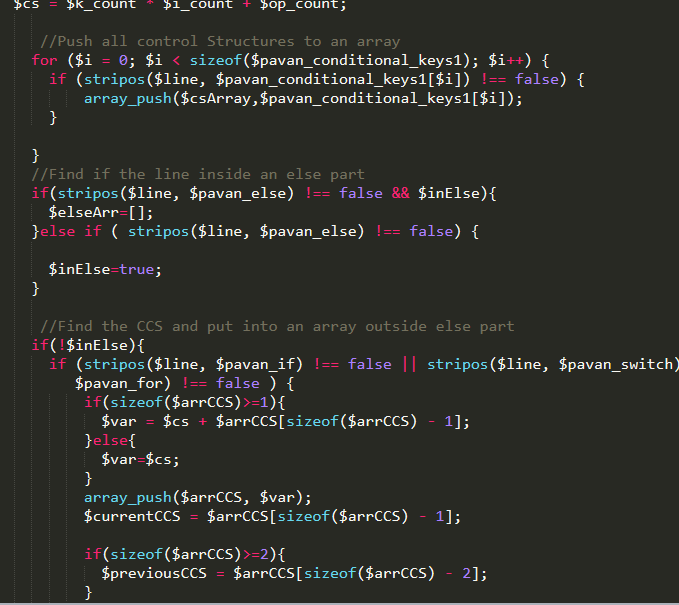
Figure:5.4.1(19) source code of *coupling*



Figure:5.4.1(20) source code of *coupling*

**5.4.2.1.4) Control Structures**

Control structures is the function that focused if-else, for, while, do-while, switch-case factors. Several columns are maintained in the table for the complexity due to the above-mentioned control structure types. With the necessary weights allocated complexity is measured using regular expression functions.



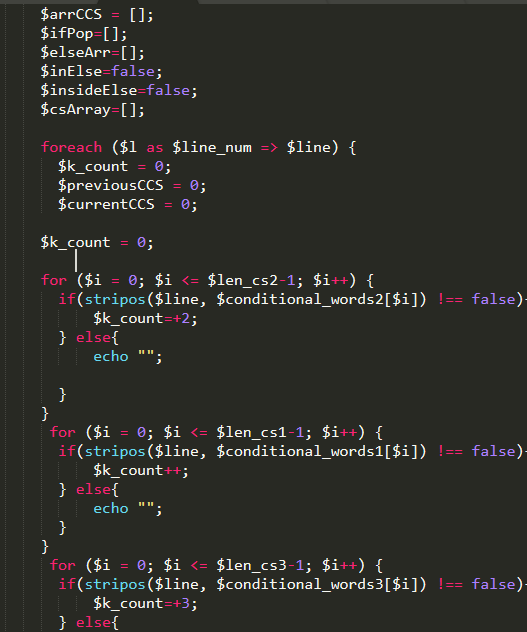


Figure:5.4.1(23) source code of Control Structures

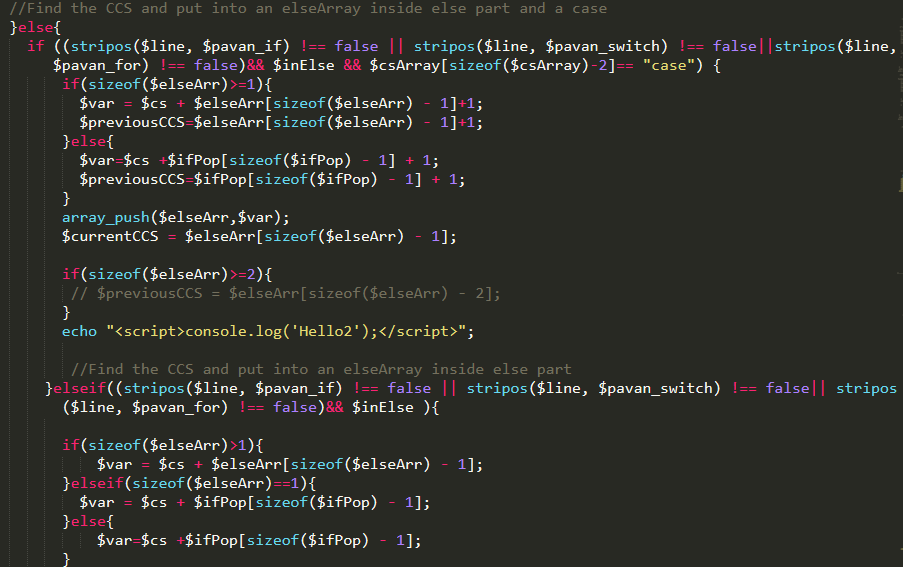


Figure:5.4.1(24) source code of Control Structures

# **6.Results and Discussion**

When we were developing our web-based code complexity measuring tool our first task was How to upload the zip and unzip folder (with the multiple files). As the methodology section we upload the codes, text files to the database and retrieve it where we wanted. It was on of our major finding in our project.

In size, variable, and method, firstly we wanted to start to analyze the complexity of a program statement due to its keywords, identifiers, operators, numerical values, and string literals. Then plan a design for the project. Secondly, we can add rich interactive components and designing user interactive navigation pages. And finally, measure the code complexity according to the computational methods as methodology. The outcomes after running the code are summarized in Figure 6.1, Figure 6.2, Figure 6.3,

In inheritance method, the systems based on object-oriented programming there is an important feature called Inheritance. Through this function, it calculates the complexity at method level regarding internal structure of methods and complexity of class hierarchies. We needed to plan the main logic and started the implementation. We used objected oriented PHP for this. Also design the interfaces (Inheritance complexity interface, Final Report). Then we can start the coding for the above-mentioned interfaces. After that implement solution for the backend. Finally, checking for bugs and fixing the functionality of the inheritance. The outcomes of the inheritance method are in figure 6.4.

In coupling method, firstly we wanted to make the detailed design of the project. After we could start designing the interfaces. Secondly, the developer can start the implementation of the above-mentioned interfaces and windows. After implementing the interfaces, we could add some rich interactive elements to the designing and the navigation pages. In the end, add more calculation methods and functions for computing the program as methodology. The outcomes of the coupling method are in figure 6.5.

In control structure method, Firstly the we could measure the conditional control structure such as an “if” or “else-if”, measure the iterative control structure such as “for”, “while” or “do-while”, and measure the “switch” and “case” statement in a “switch-case” control structure of the code. Finally, implement a complete solution for other components and checking the bugs and fixing the functionality. The outcomes of this part is in figure 6.6

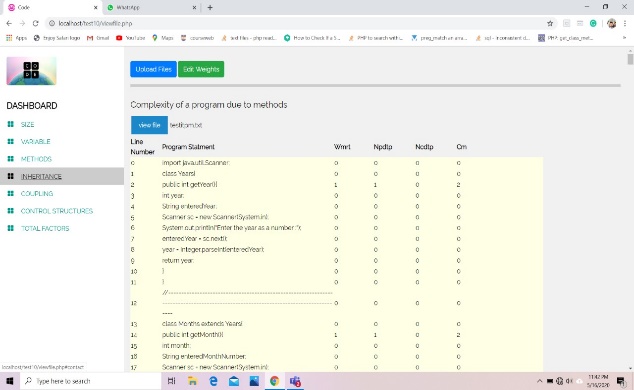


Figure 6.1: methods calculating and displaying interface

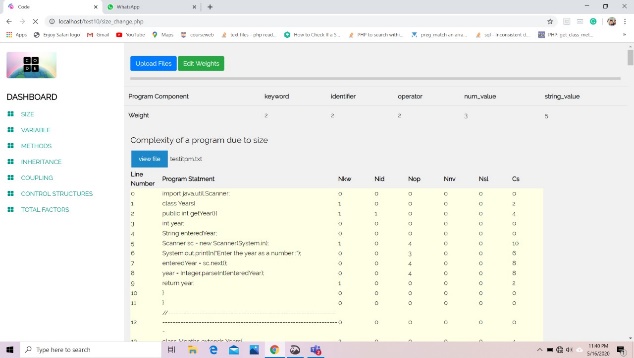


Figure 6.2: Size calculating and displaying interface

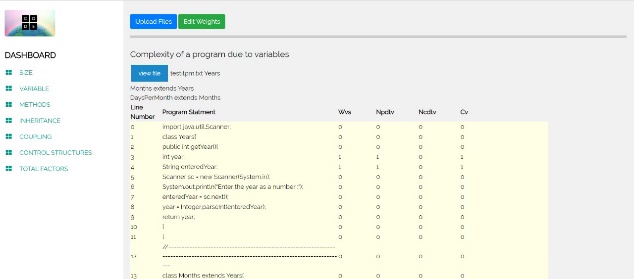


Figure 6.3: Variables calculating and displaying interface

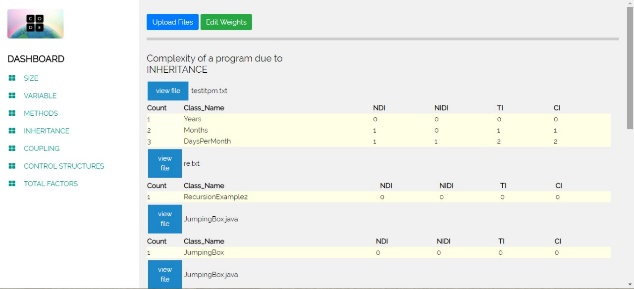


Figure 6.4: Inheritance calculating and displaying interface

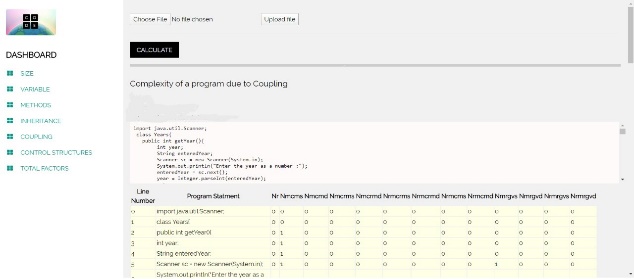


Figure 6.5: Coupling calculating and displaying interface

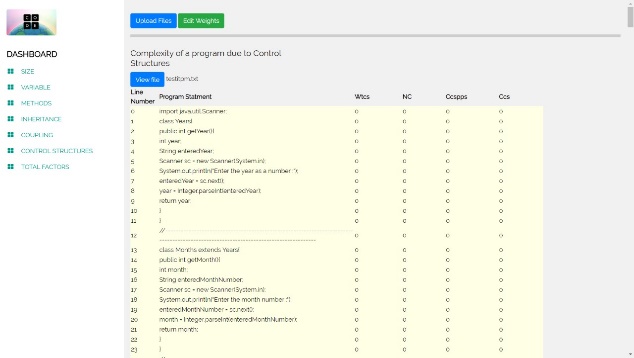


Figure 6.6: Control structures calculating and displaying interface

The problem of this code analysis tool is every text file support to the upload function. As an example, if you choose the text part of the book, the upload function gets it into the calculation function without error. Even if it is, the upload functions and calculate functions are correctly computed. And also, the outputs are properly displayed.

In this implicated code complexity measuring tool is easy to use and its target audience is basically programmers and developers in IT field.

The main advantage of this tool is it's programmed to measure the complexity C++ and Java at the moment but in the future developers can add more programming languages easily because of its easy architecture.

Also, anyone can access this web-based application easily through internet.

**7.Conclusion**

These days software developing company using many software metrics to estimate the complexity od software systems to find software maintenance. Software industry uses software metrics to measure the complexity of software systems for software cost estimation, software development control, software assurance, software testing, and software maintenance. Software complexity metrics for taking nonphysical software attributes, like readability, into considerations. The applications of Readability Metrics are good in indicating the additional efforts required for less readable software systems and help in keeping the software systems maintainable. The numerous metrics and the complicated formulas in the family usually make it tedious to apply Readability Metrics to large scale software systems.

Readability Metrics indicate the readability of software systems and help in keeping the source code readable and maintainable. Allowing a consistent examination of every possible outcome when running the resulting software. Using software version control repository information, we coarsely separate changes made to address bugs from other changes.

The research proposed a constructive way to assess and measure aspects of project complexity. The descriptions of those contributors from the perspective of complexity theory and complexity management were obtained to describe project complexity. The positive impacts and reduce negative impacts of project complexity to manage projects in an effective way.

**8.References**

* <https://www.geeksforgeeks.org/control-structures-in-programming-languages/>
* <https://www.geeksforgeeks.org/coupling-in-java/>
* <https://www.cloudflare.com/learning/serverless/glossary/client-side-vs-server-side/>

**Group Members**

|  |  |
| --- | --- |
| **Name** | **Registration ID** |
| L.H.M.D.S.B.Lansakara | IT18152524 |
| B.M.D.E. Bannaka | IT18074796 |
| K.G.A.H Abeyrathne | IT18119190 |
| H.M.A.S Herath | IT18118582 |