Thilakarathna U.G.C. IT18121384, Rathnasekara M.M.P.V. IT18161502, Fernando P.I.S.P. IT18118728, and

Bandara A.M.R.Y. IT18171570

Code Complexity Calculating Tool

*Abstract*— The technological advancements have influenced the society to take a leap towards success. Every technological reform is a small step towards advancement and progress of mankind. Development in information have also been impacting upon educational institutes. The utilizes computers and internet. But the available technology is not used in a way that can achieve more or in a way which will increase the productivity. This system will support the environment strategic goals and directions. But this discussion is planned to furnish driving force in exploration and help excite further interest. In to this research is about a tool designed and developed to detect the complexity of both java and C++ programs by linking weights to different characters of classes in the different or same file, due to complexity of calculating final value according to the Size, Methods and Variables, Inheritance, Coupling, and Control Structures. Eventually, the total complexity of the program is calculated.

The important of of measure the coupling and cohesion of the modules in the codebase as well in order to get a true system-level, software complexity grater predictability knowing the level of complexity of the code being maintained makes it easier to know how much maintain a program will need.

Most traditional measures used to quantify software complexity. They are simple, essay to count, and very easy to understand. They do not, however, take into account the intelligence content and the layout of the code. There are many benefits of software complexity. There are improve code quality, Reduce maintenance cost, Heighten productivity, Increase robustness, Meet architecture standards.

There are many various complexity tools. There were many attempts to measures software complexity but they all failed in some way.

**For example:**

* **Line of Code.**
* **Cyclomatic complexity.**
* **Assignment, Branches and Condition Metric.**

**Code Development tools consists of mainly programming languages, Debugging tools, Version Management tools compiling tools and Integrated development.** <https://www.researchgate.net/figure/Main-features-of-the-developed-tool_fig4_322506435>, <https://www.tldp.org/HOWTO/Scientific-Computing-with-GNU-Linux/devtools.html>

# INTRODUCTION

F

IRST we thank for lecturer to give us this topic of project. It is a new side of the projects. We do projects in last semesters. But this project is very important for us. Also in this project about deeply get things in this. Actually different projects in YouTube, internet etc. But in this type projects directly cannot see in the internet. Also, How to build this project? , How to run this project? , What advantages of project? etc information get by internet indirectly. It is an accustomed measurement tool to test their ASCII document. Then they will reduce complexity and maintenance costs within the long run. It is a controversy within the software development lifecycle.

In this topic, we build a Desktop Application. It will support many of the activities in the Application. We use java language and create interfaces to use NetBeans IDE. We includes size, variables, methods, inheritance, coupling and control structured in this project. Our system will give many solutions to the problems that they are currently facing.

# Complexity of a program statement due to size.

The user is able to measure his submitted code according to the size component through the web base code measuring tool which the complexity is focused on Java.

# Complexity of a program statement due to Variable.

Only the lines which consist of declared or defined variables are considered under this factor. Scope, primitive data type and composite data type variables are considered when computing.

# Complexity of a program statement due to Method.

Only the lines which include method signatures are considered under the methods factor.

# Tasks of the developer under the size, variable, and method.

Firstly, the developer wants to acquisition the software. After that start to analyze the complexity of a program statement due to its keywords, identifiers, operators, numerical values, and string literals.

Actually people should care about our project. There are five reasons we has measure software complexity. In IT industry, you can’t manage what you can’t measure. Also knowing how an organization’s application portfolio is provides insight into how to manage it best. The problem is the issue that comprise software complexity. The multiple system interfaces and complex requirements, the complexity of the systems sometimes grows beyond control, rendering application and portfolios too costly to maintain ad risky to enhance.

## Complexity Metrics

Fortunately, there have been many methods developed for measuring software complexity. Most break down software complexity according to the following metrics

## Cyclomatic Complexity

Programs with more conditional logic are more difficult to understand, therefore measuring the level of cyclomatic complexity unveils how much need to be managed. For example If,do,select etc. In the first case complexity metrics will look bad. In second, the complexity metrics will look good, but the result will be deceptive. It is important, therefore, to measure the coupling and cohesion of the modules in the code-base as well in order to get a true system-levels, software complexity measure.

## Halstead Volume

We are measuring how much “information” is in the source code and needs to be learned.

## Maintainability Index

Maintainability index is more of an empiric measurement, having been developed over a period of years by consultants working with Hewlett-Packard and its software teams.

Organizations that have this information can capitalize in a number of ways.

1. Greater predictability

2. Software Risk Mitigation

3. Reduced Cost

4. Extended Value

5. Decision Support

Complexity arise when the dependencies among the elements become important. The behavior of many complex systems emerges from the activities of lower-level components. Typically, this emergence is the result of a very powerful organizing force that can overcome a variety of changes to the lower-level components.

There are four proposed tools to use. There are Sonarqube, Selenium, Targetprocess and Git. In this desktop application we can use Sonarqube, Targetprocess and GitHub.

# Related Work/Literature Review

We use more tools to develop our project importantly among that we use Winium, SonarQube, Targetprocess and GitHub

## SonarQube

We used SonarQube to analyze source codes, measuring quality and providing reports for our project. It combines static and dynamic analysis tools. It enables quality to be measured continuously over time. SonarQube is evaluate and inspect ,from minor styling details to critical design errors, Everything that affects our code base, thereby enabling to access and track code analysis data ranging from styling errors, potential bugs, and code defects to design inefficiencies, code duplication, lack of test coverage, and excess complexity. The Sonar platform analyzes source code from different aspects and hence it drills down to our code layer by layer, moving from the module level down to the class level. At each level, revealing problematic areas in the source that require inspection or improvement.

## Winium

Winium is a Selenium based tool for testing and automating desktop applications on the Windows desktop. It is the tool that is built on Selenium, and interact with Windows applications. It is free and open source.

We use this to test our desktop application

## Targetprocess

Targetprocessis a commercial agile project management tool that allows following a Scrum, Kanban, or customized approach. It provides an intuitive and rich visual interface to manage our project in a collaborative way. We use this to manage our project.

## GitHub

GitHub is a version control system, we make constant changes to the code, keep these revisions straight, storing the modifications in a central repository. This allows us to easily collaborate, as they can download a new version of the software, make changes, and upload the newest revision. Our group members can see these new changes, download them, and contribute. It stores file changes more efficiently. We use this to integrate our project.

# Proposed System

When develop a desktop application since it is an application that runs locally on a computer device and hence rather than a web application, it is not necessarily required of any network connections.

Our primary target become to enforce this device which helps for both Java and C++ programming languages on account that many groups inside the corporate world tend to use these because of few reasons mainly. Such as due to the fact Java is platform-independent it could flow from one pc to another, in order that it's far more tremendous to the software developing fields as the identical program may be capable of run on unique systems. So as in Java, Portability characteristic of C++ also useful in programming in order that codes may be run on distinctive operating systems.

In this code complexity measuring device, we were proposed to measure the complexity of user described software statements due to the following factors.

## Main Functions

Complexity is one of the most frequently appearing term in systems literature.

In this system can calculate complexity due to four main factors. That’s are;

1. Calculate Program Complexity Due to Size, Variable and Method Factors.

2. Calculate Program Complexity Due to the Inheritance Factor.

3. Calculate Program Complexity Due to the Coupling Factor.

4. Calculate Program Complexity Due to the Control Structure Factor.

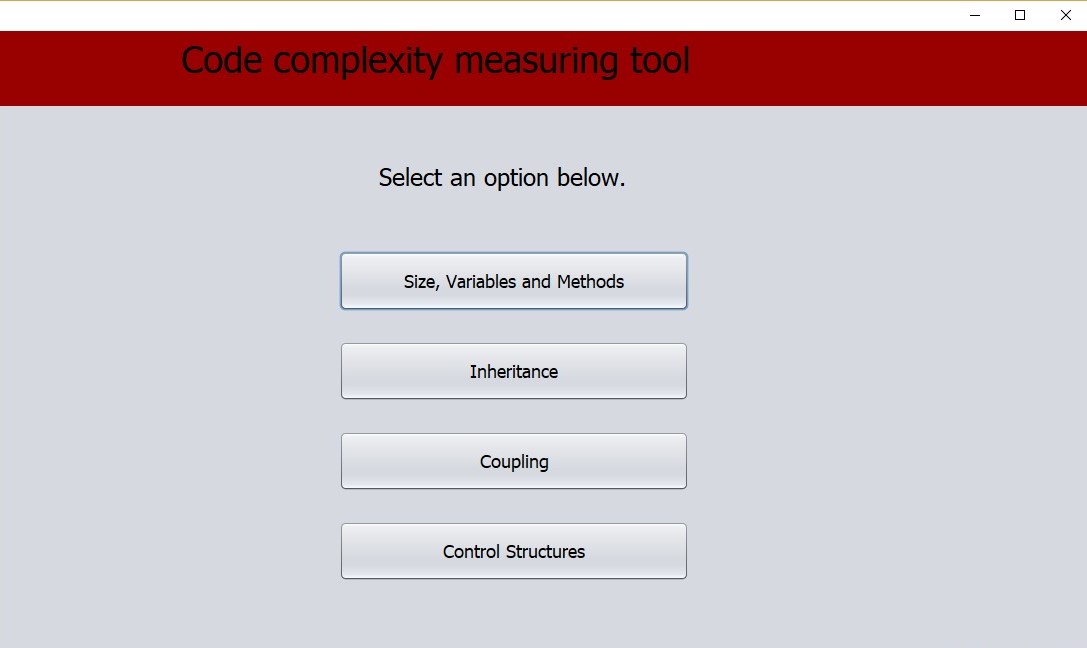


Figure 1: Interface of Dashboard

Basically above Size, Variable and Method function contains three sub methods as;

1. Program Complexity Due to Size.

2. Program Complexity Due to Variable.

3. Program Complexity Due to Method.

### Calculate Program Complexity Due to Size, Variable and Method Factors.

In this case it basically contains three sub methods as;

#### Program Complexity Due to Size.

It is the ‘code length’ which characterizes the components of the equality check framework. Typically, a code is characterized by utilizing its length and line section weight in the statement

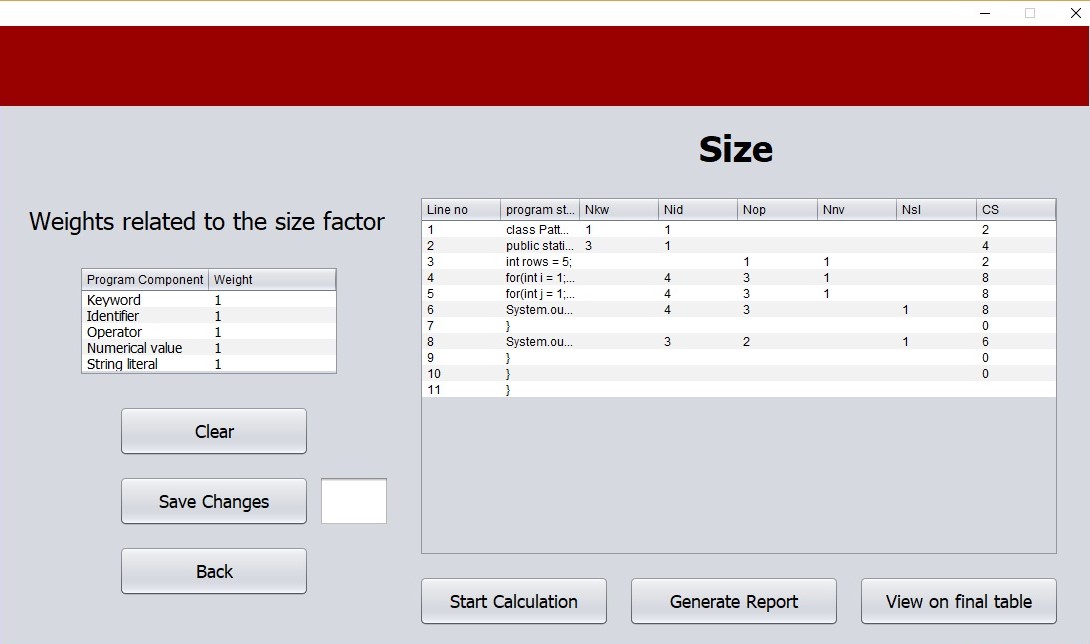


Figure 2: Interface of Size Dashboard

There are five program components which are considered under the size factor.

### Program Complexity Due to Variable

In programming, a variable is a really worth which can change, contingent upon conditions or on statistics exceeded to the application.

Regularly a application comprises of steering s that coach the PC and data that the application utilizes when it's far running. Also they used to store records to be referenced and managed in a PC program.

They moreover furnish a way of naming information with an attractive name, so our projects can be seen all the more plainly via the reader and ourselves. There are mainly two types of variables. That are:

• Local Variables.

• Global Variables.

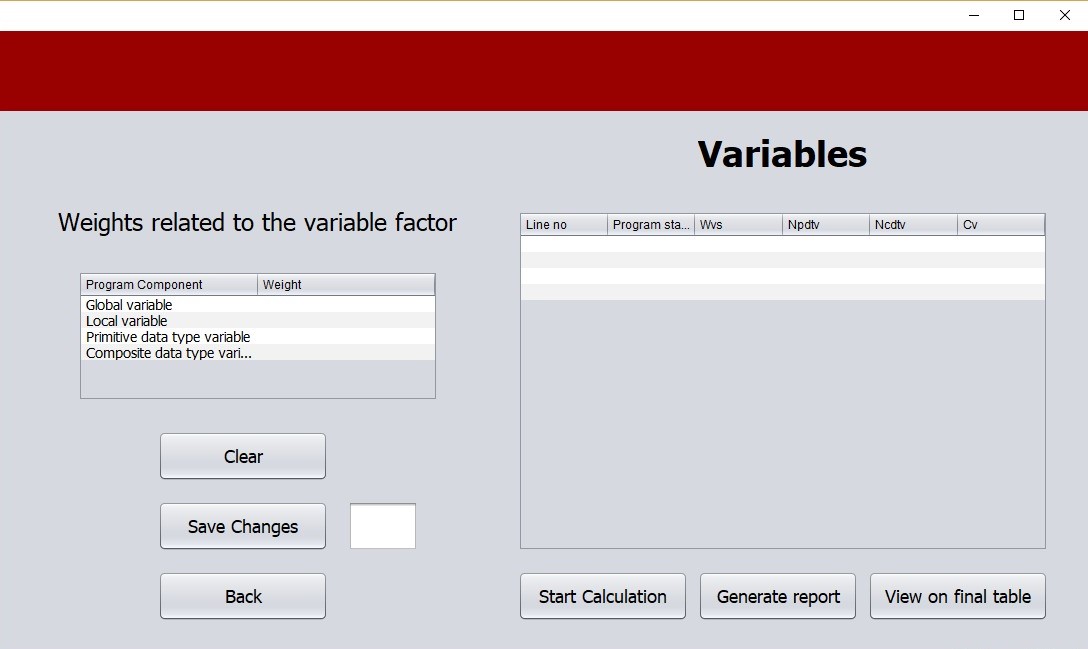


Figure 3: Interface of Variable Dashboard

### Program Complexity Due to Method.

Method is a technique or capacity related with a class. As a component of a class, a technique characterizes a specific conduct of a class example. An object method can just approach the information known by that object which keeps up the trustworthiness of information between sets of articles in a program.

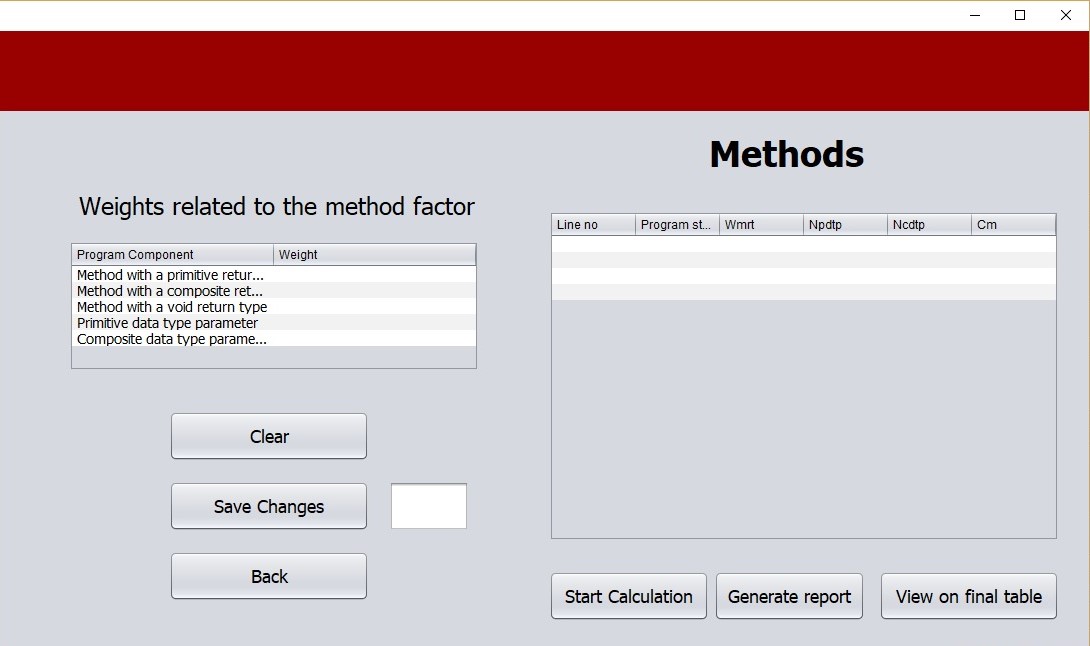


Figure 4: Interface of Method Dashboard

### Calculate Program Complexity Due to the Inheritance Factor.

Inheritance in Java is an instrument where one object secures all the properties and practices of a parent object.

The key point in Java is that we can make new classes that are based after existing classes. At the point when we acquire from a current class, we can reuse techniques and fields of the parent class. Additionally, we can include new strategies and fields in your present class too.

There have few terms used in inheritance as;

• Class.

• Subclass.

• Superclass.

• Reusability.

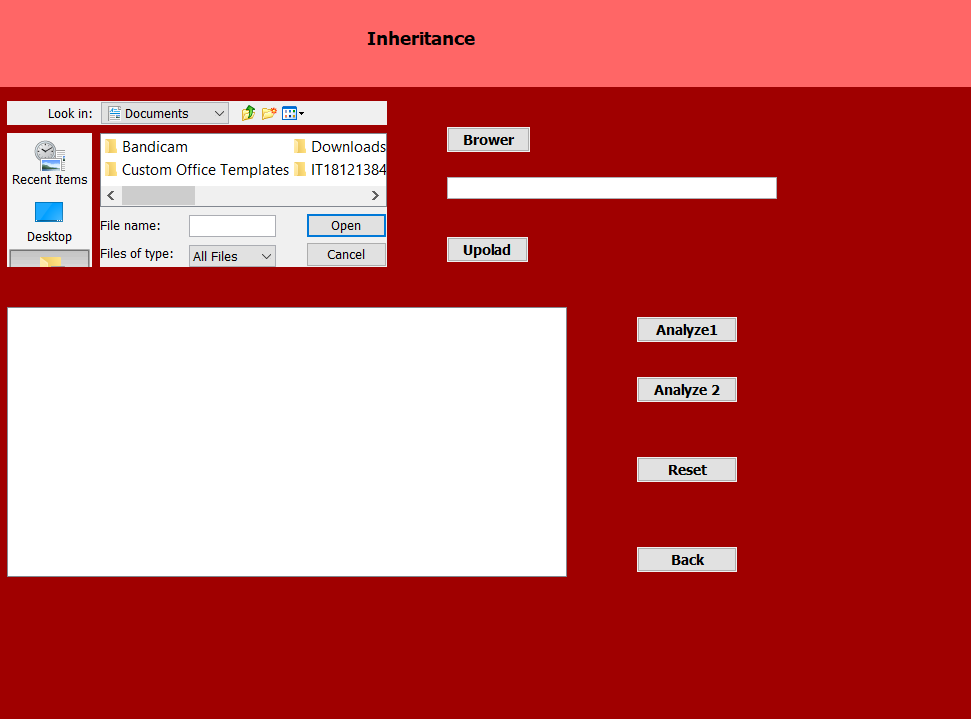
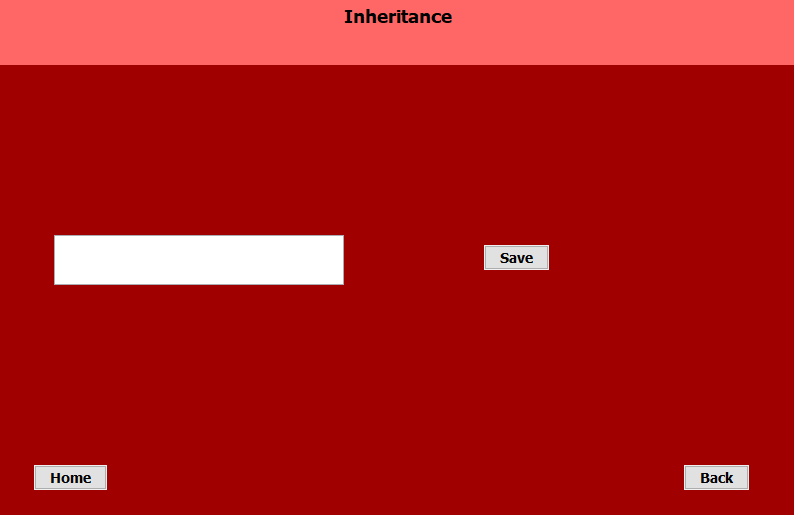


Figure 5: Interface of Inheritance Dashboard

The inheritance complexity of a program statement which belongs to a class is same as the weight assigned for that class due to its inheritance pattern.

Then user can browse the file that containing code to calculate the inheritance by clicking browse button and after upload it.



Fugure 6: Interface of Inheritance File Browse

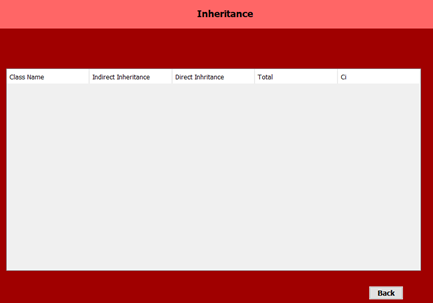


Figure 7: Interface of Inheritance Output

After upload successful, when clicking analyze button it returns the result of code complexity due to the inheritance.

### Calculate Program Complexity Due to the Coupling Factor

In software engineering, 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.

In programming, coupling plays a main role due to the fact that it is able to test in a higher way. Since the code won’t depending on different objects, it makes us ease to write down the unit tests and so on.

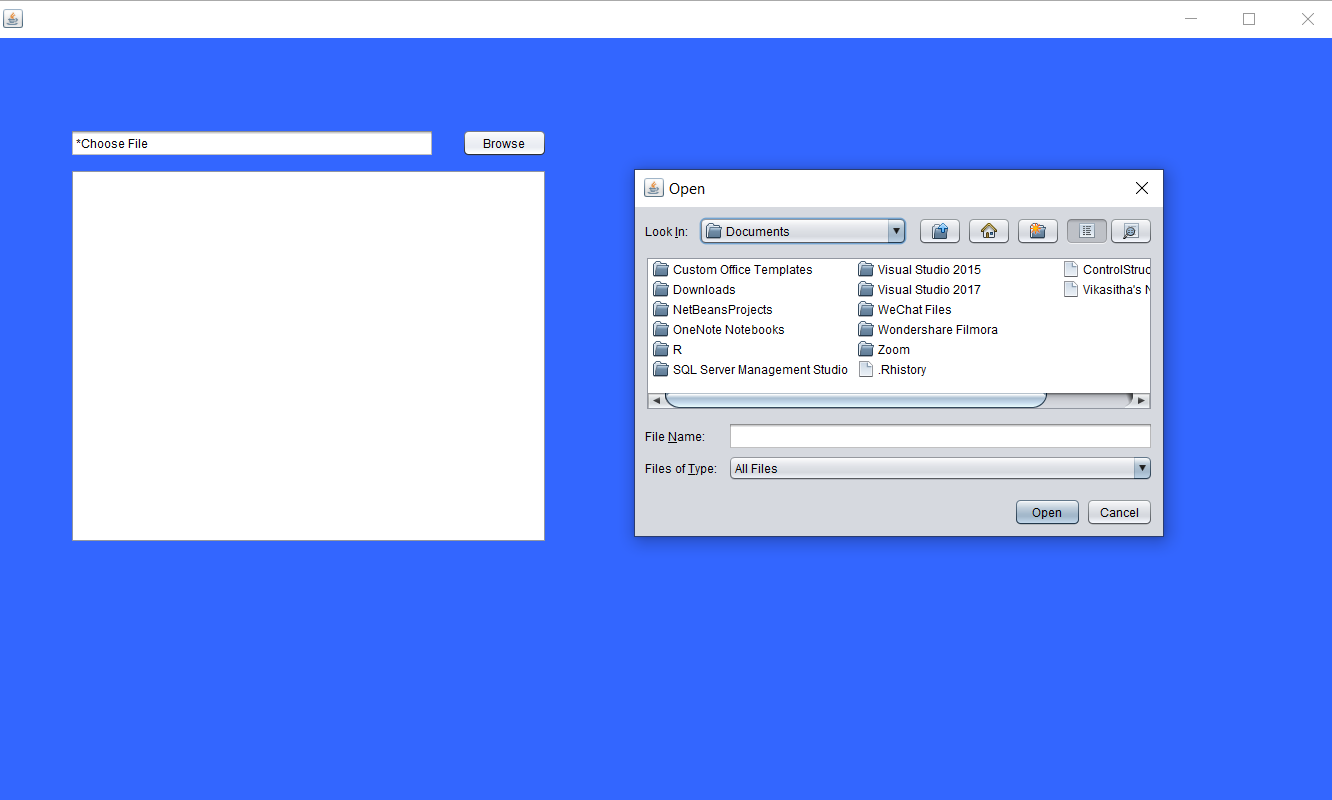


Figure 8: Interface of Coupling File Browse Dashboard

By the above way user can browse and upload the file to system and can save it. Then can calculate complexity due to coupling.

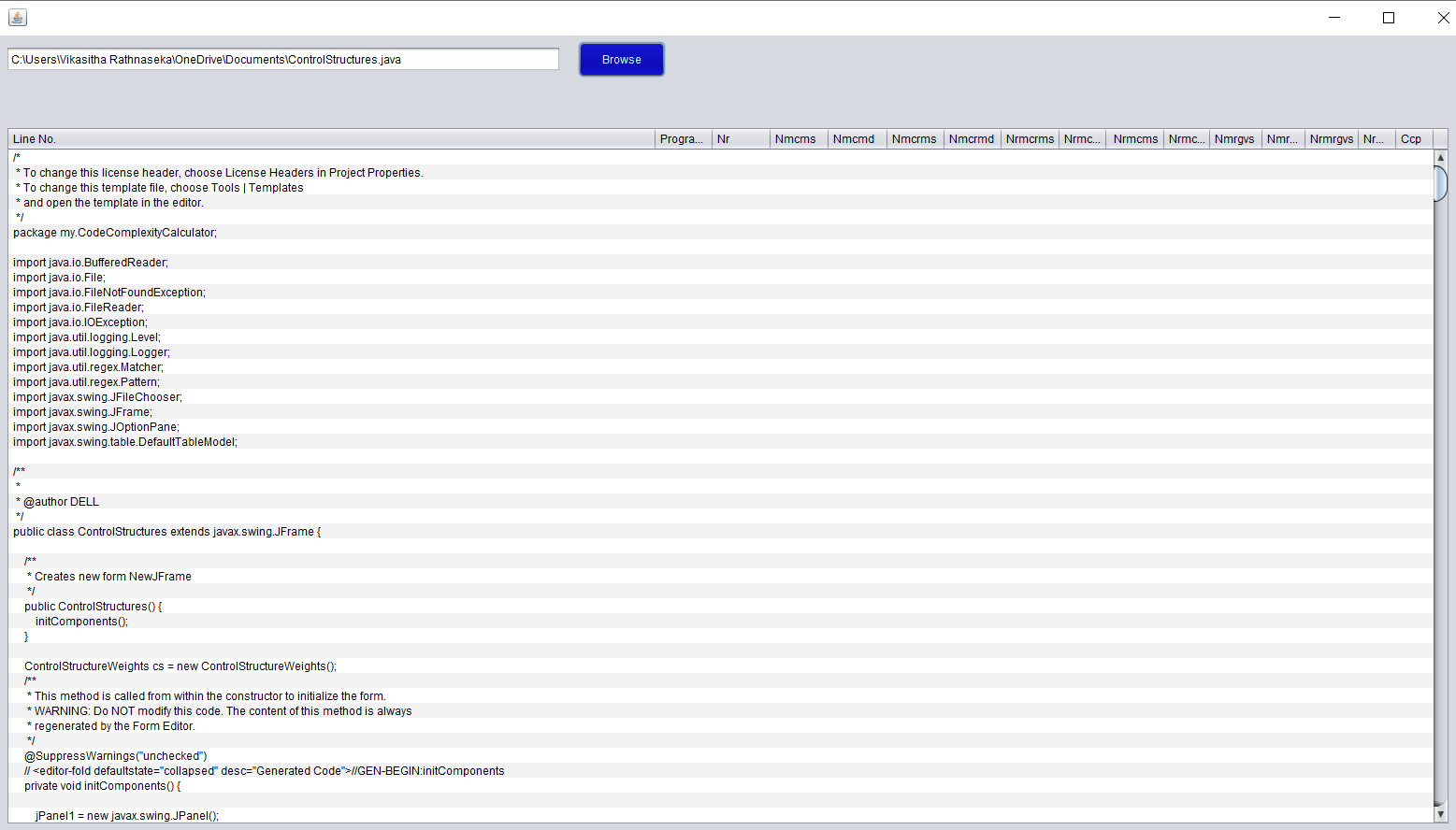


Figure 9: Interface of Coupling Result Dashboard

### Calculate Program Complexity Due to the Control Structure Factor.

A control structure resembles a square of programming that examinations factors and picks a bearing wherein to go on given parameters.

In coupling, there are three main fundamental control structures.

a) Sequence control structure

b) Decision control structure.

c) Loop control structure

For control structures which includes ‘for’, ‘while’, or ‘do-while loops which are known as iterative control structures.

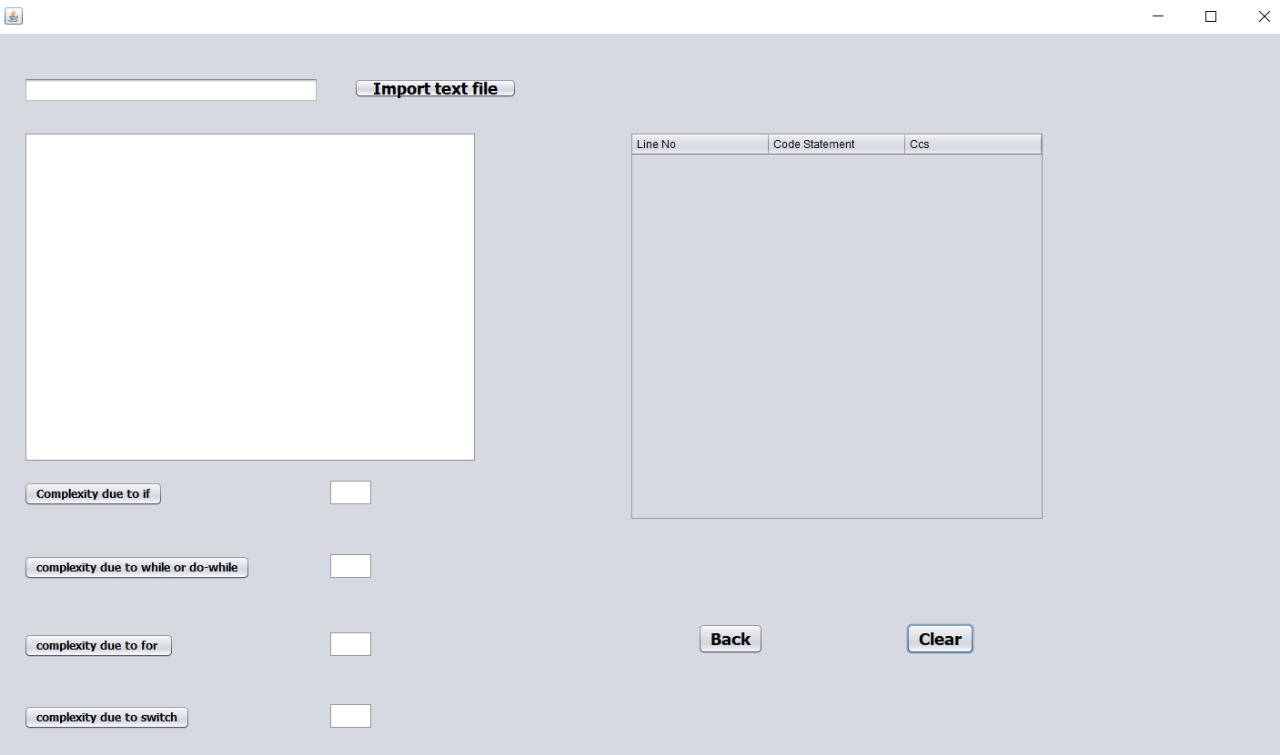


Figure 10: Interface of Control Structure Dashboard

In calculating complexity due to control structure can upload file by browse and it analyze by the system. After it prints the result.

As the above the entire complexity of a application additionally be measured with the aid of including all the factors which are;

 Size.

 Variable.

 Method.

 Inheritance.

 Coupling.

 Control Structure

After developing the desktop application, the user is able to check the complexity of the desired program statement.

## Front-end Flow.

### Select a File.

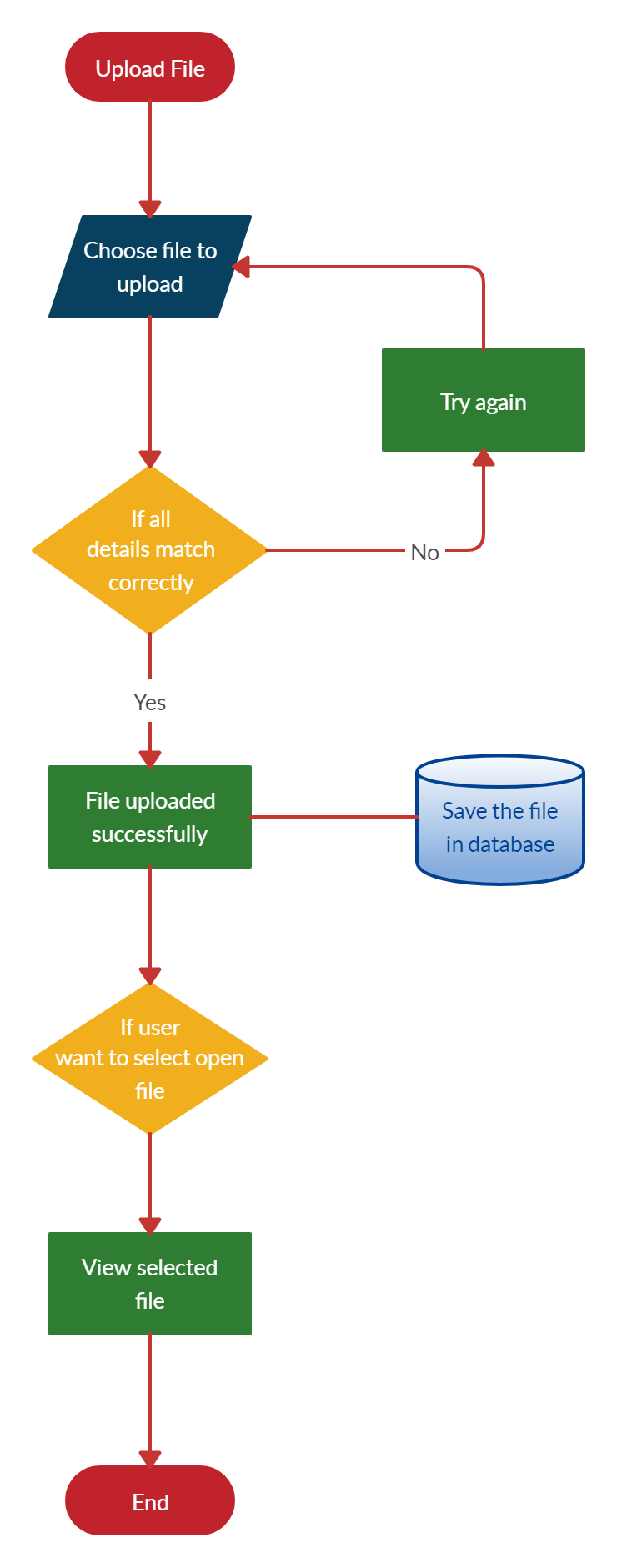


Figure 11: File Select Flow Chart

When selecting file and upload it. If it occur any issue, it display error and want to re-upload. When upload complete it has been save to the database also can select it via database and can retrieve file from database to Program Complexity Calculator.

### Modification of Allocated Weights.

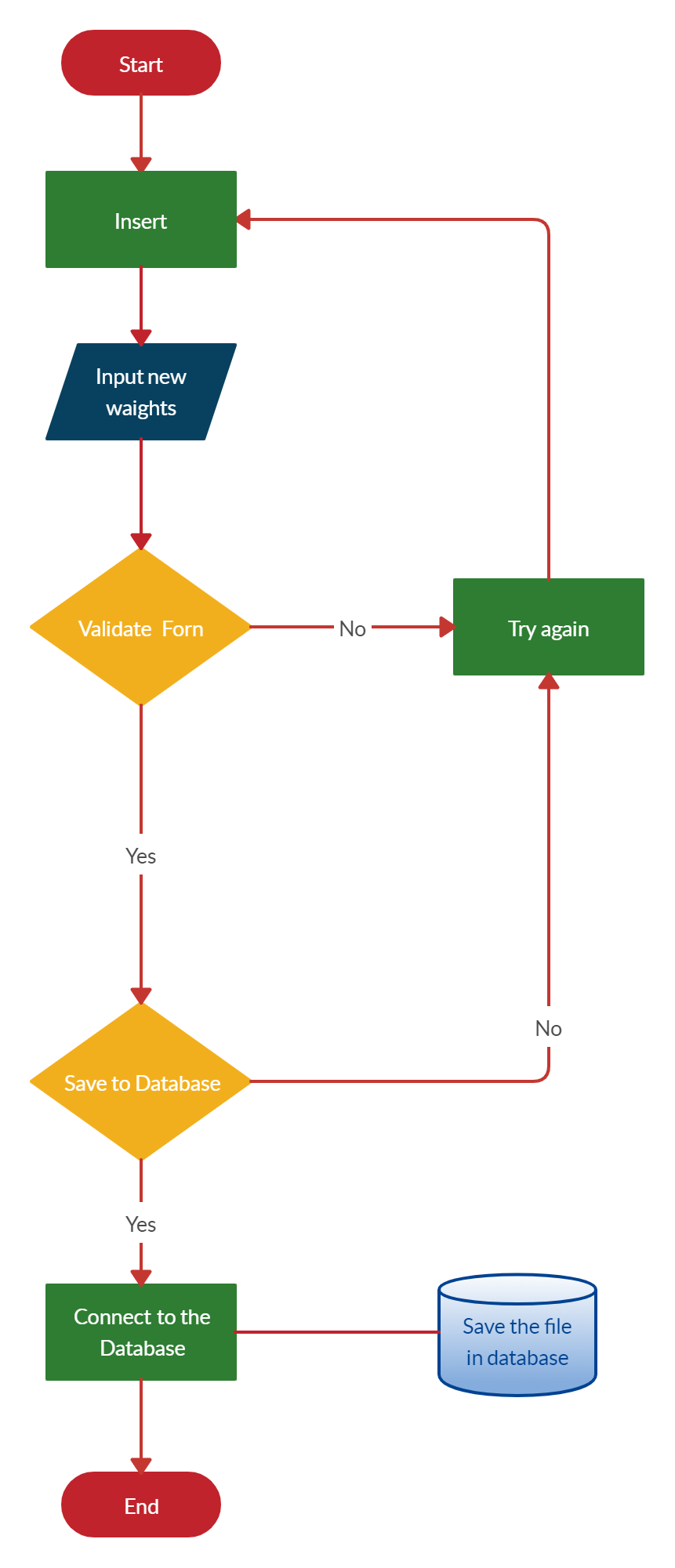


Figure 12: Modify Weights Flow Chart

This system can modify their values of any weights. In this can select particular weights and give new values by client and save it. So then it insert to database automatically by only clicking save button.

# Methodology.

## Requirements Gathering and Analyzing.

The project is a code complexity measuring tool for the CEO of CDE IT solutions company. For this project, we gathered information from the CEO. Mainly he needs to reduce the maintenance cost of the soft wares developed by the company. In here we focused about six functions. After that we categorized these six functions into four main functions. They are:

1. Measure complexity of a program statement due to size, variables, and methods.

2. Measure complexity of a program statement due to inheritance.

3. Measure complexity of a program statement due to coupling.

4. Measure complexity of a program statement due to control structures.

After gathering requirements we started feasibility study. After that we gained basic understanding of the client's problems.

## Implementation.

For this project, we created a desktop application. We used Java for programming language. Java is a general purpose language that included class based, object oriented and abstract features.

### Measure complexity of a program statement due to size, variables, and methods. C:\Users\Ishara Fernando\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot (92).pngScreenshot (93)

Figure 13: Diagram of Size, Variable and Methods

### Measure complexity of a program statement due to inheritance.

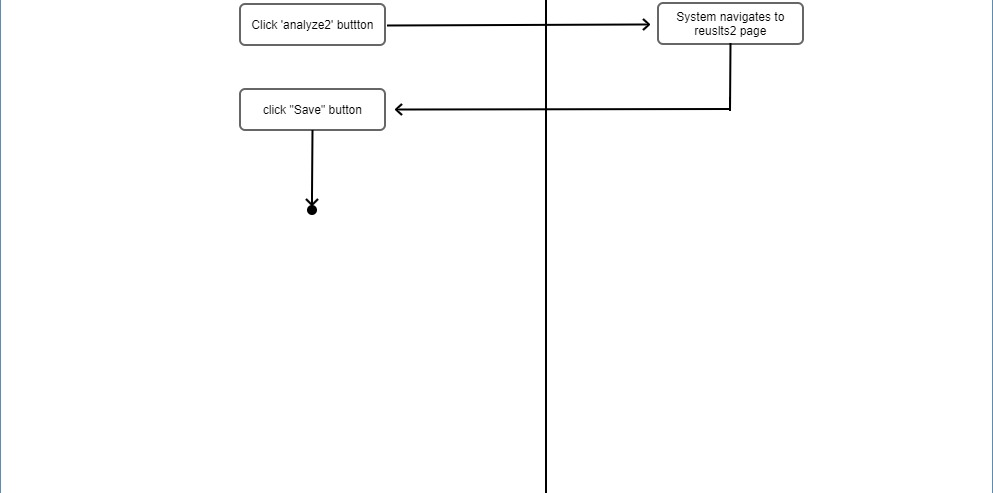
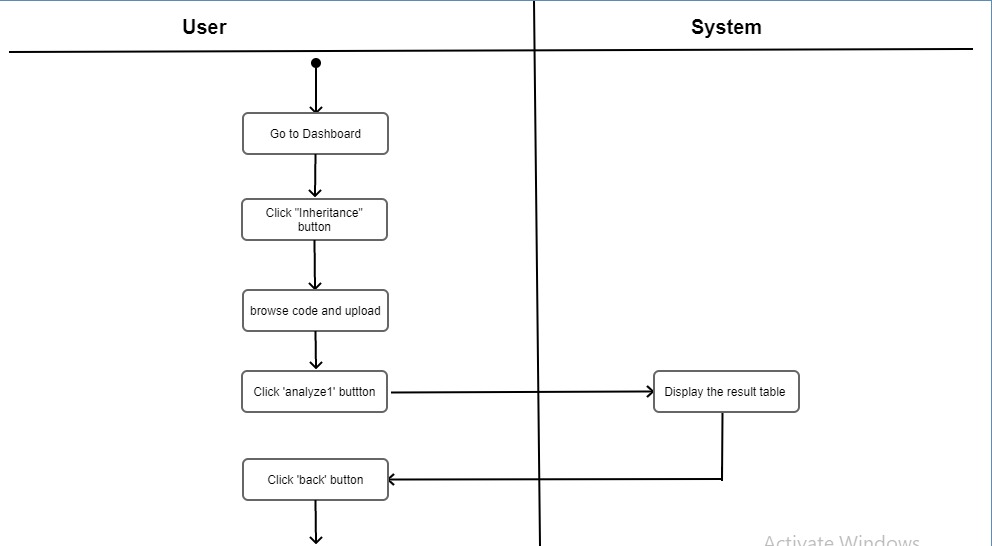
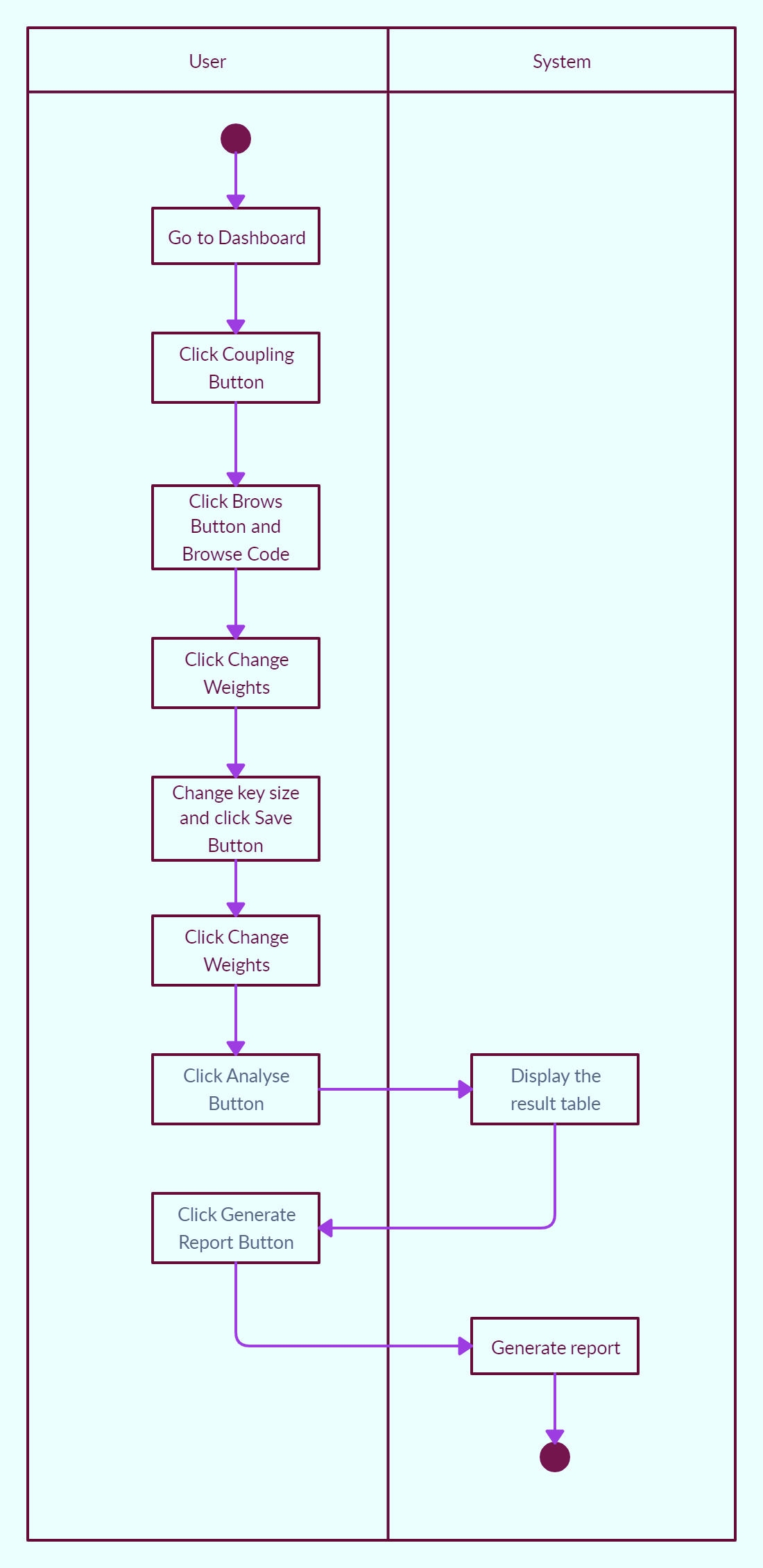


Figure 14: Diagram of Inheritance

### Measure complexity of a program statement due to coupling.



*Complexity of program due to methods.*

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

*Complexity of program due to size.*

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

*Complexity of program due to variables.*

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

Figure 15: Diagram of Coupling

### Measure complexity of a program statement due to control structures.

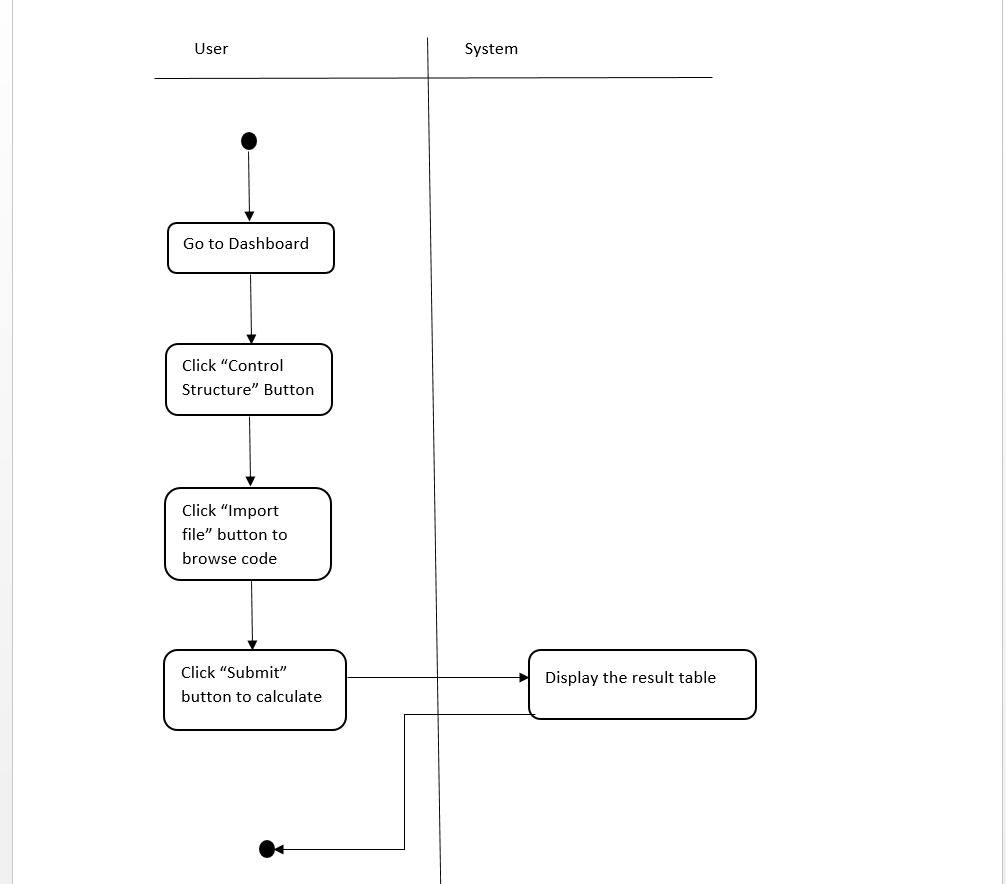


Figure 16: Diagram of Control Structures

*Complexity of program due to coupling..*

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)

# Conclusion

We have a group project in ITPM module in this semester. In this journal we describe about our project of code complexity measuring tool. According to that we plan to develop a desktop application. We write this journal article to describe the Developed complexity measuring tool/system.

We use Winium, SonarQube, Targetprocess and GitHub as a tools to develop our system.

In this system we can calculate Inheritance, Coupling, Control structures, Size, variables, and methods separately. The proposed system part in this journal describe the steps that a user should follow.

The Results and Discussion part describe the specific results of the proposed system.

Our system code complexity measuring tool develop to calculate code complexity for only java code

We research to develop this to for other languages. And we develop a desktop application for this project. But can develop a web based application to using java language.

References.

1. *Control structures in programming languages*. (2020, January 16). GeeksforGeeks. <https://www.geeksforgeeks.org/control-structures-in-programming-languages/>
2. *Attention required!* (n.d.). Cloudflare - The Web Performance & Security Company | Cloudflare[. https://www.cloudflare.com/learning/serverless/glossary/client-side-vs-server-side/](file:///C:\Users\Vikasitha%20Rathnaseka\Desktop\. https:\www.cloudflare.com\learning\serverless\glossary\client-side-vs-server-side\)
3. *Coupling in Java*. (2020, February 5). GeeksforGeeks. <https://www.geeksforgeeks.org/coupling-in-java/>
4. *NetBeans IDE Java quick start tutorial*. (n.d.). Welcome to NetBeans. [https://netbeans.org/kb/docs/java/quickstart.html#build](https://netbeans.org/kb/docs/java/quickstart.html%23build)
5. Vaidya, N. (2020, May 7). *Advanced Java tutorial | Learn advanced Java concepts with examples | Edureka*. Edureka. <https://www.edureka.co/blog/advanced-java-tutorial>

*Complexity of program due to control structure.*

Ccs = (Wtcs \* NC) + Ccspps

1. *Java programming tutorial*. (2018). Leading Global University in Singapore | NTU Singapore | NTU. <https://www.ntu.edu.sg/home/ehchua/programming/java/J4a_GUI.html>
2. *IEEE format for a project report*. (n.d.). Bizfluent. <https://bizfluent.com/info-7872885-ieee-format-project-report.html>