**2. Methodology**

**2.1 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.

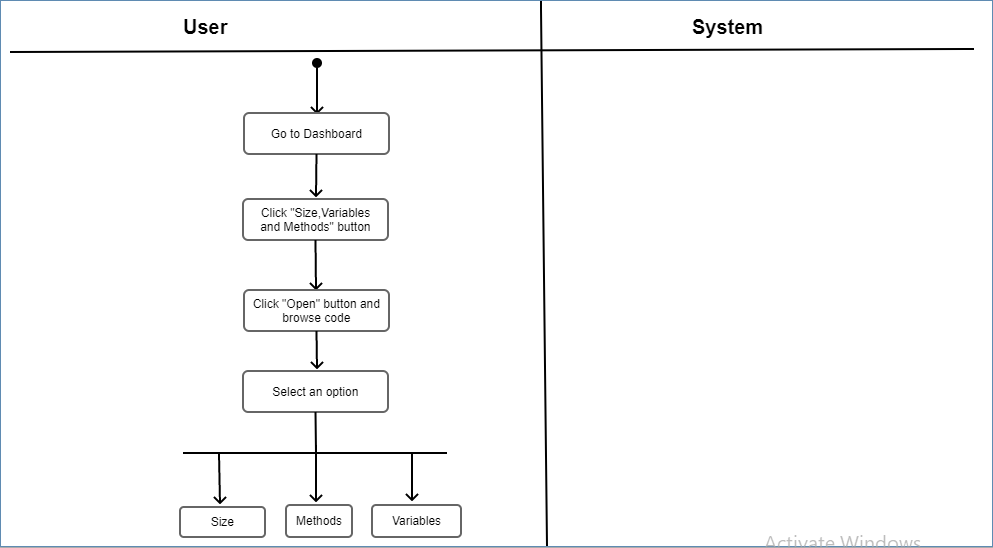
After gathering requirements we started feasibility study. After that we gained basic understanding of the client's problems.

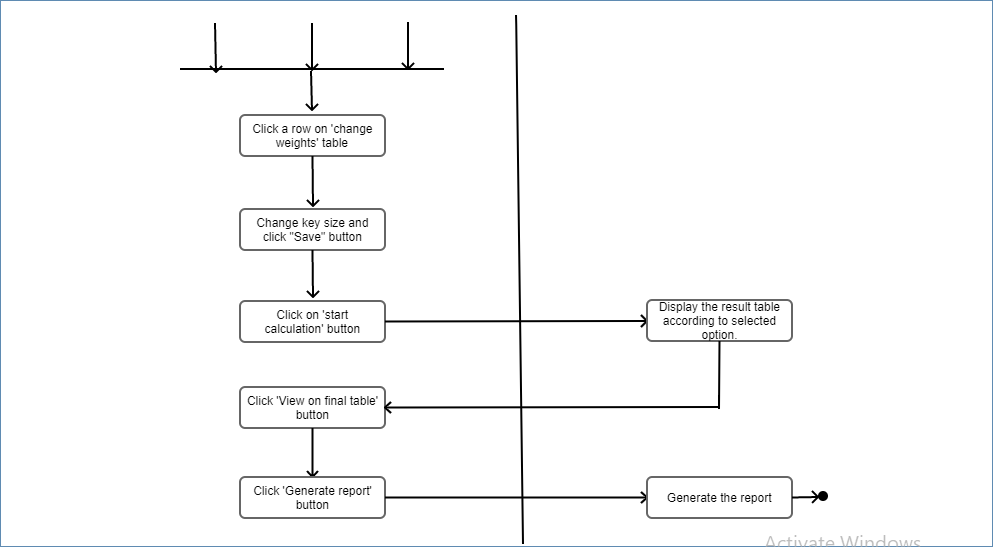
**2.2 Implementation**

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.

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

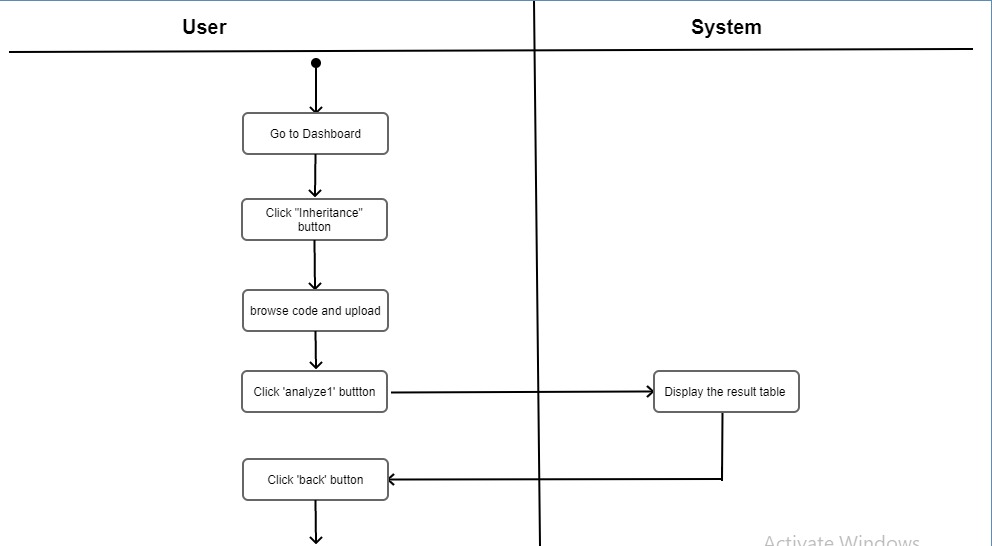
**Activity Diagram**

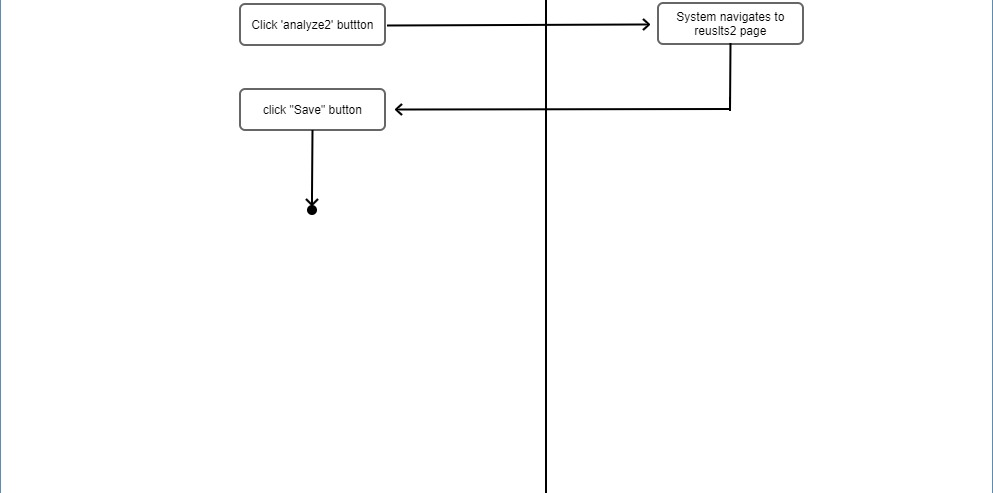




1. **Measure complexity of a program statement due to inheritance.**

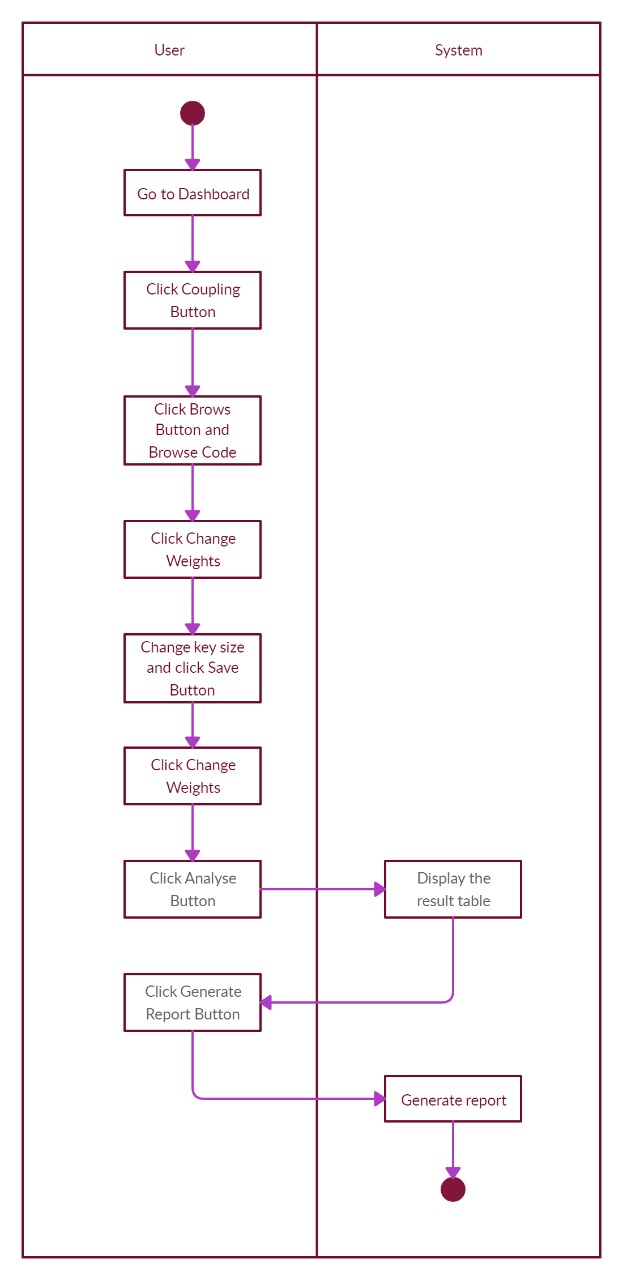
**Activity Diagram**





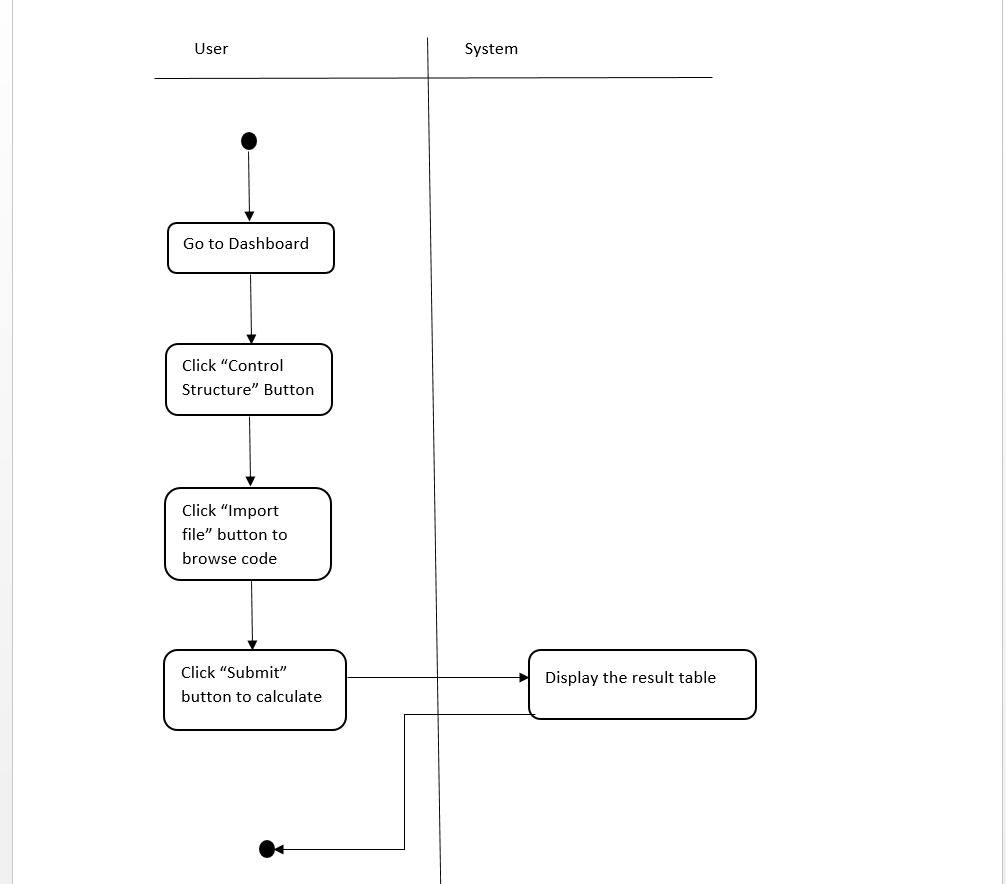
1. **Measure complexity of a program statement due to coupling.**

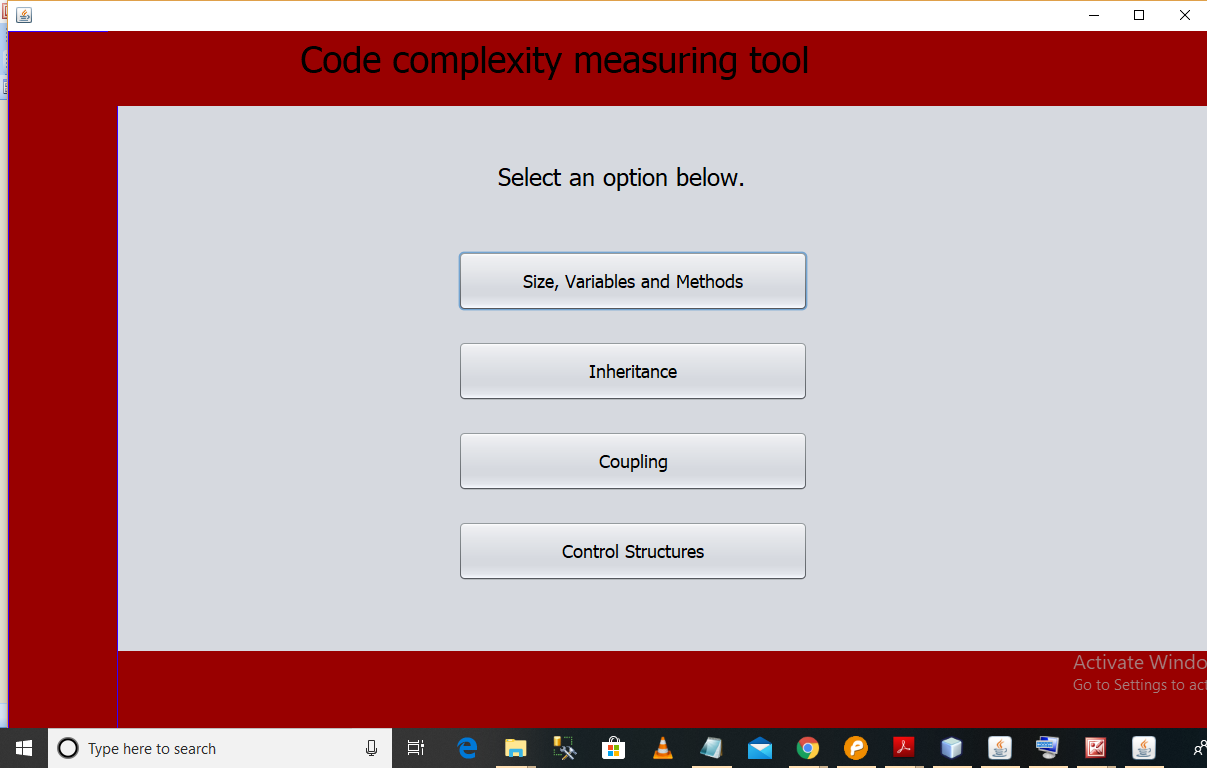
**Activity Diagram**



1. **Measure complexity of a program statement due to control structures**.

**Activity Diagram**

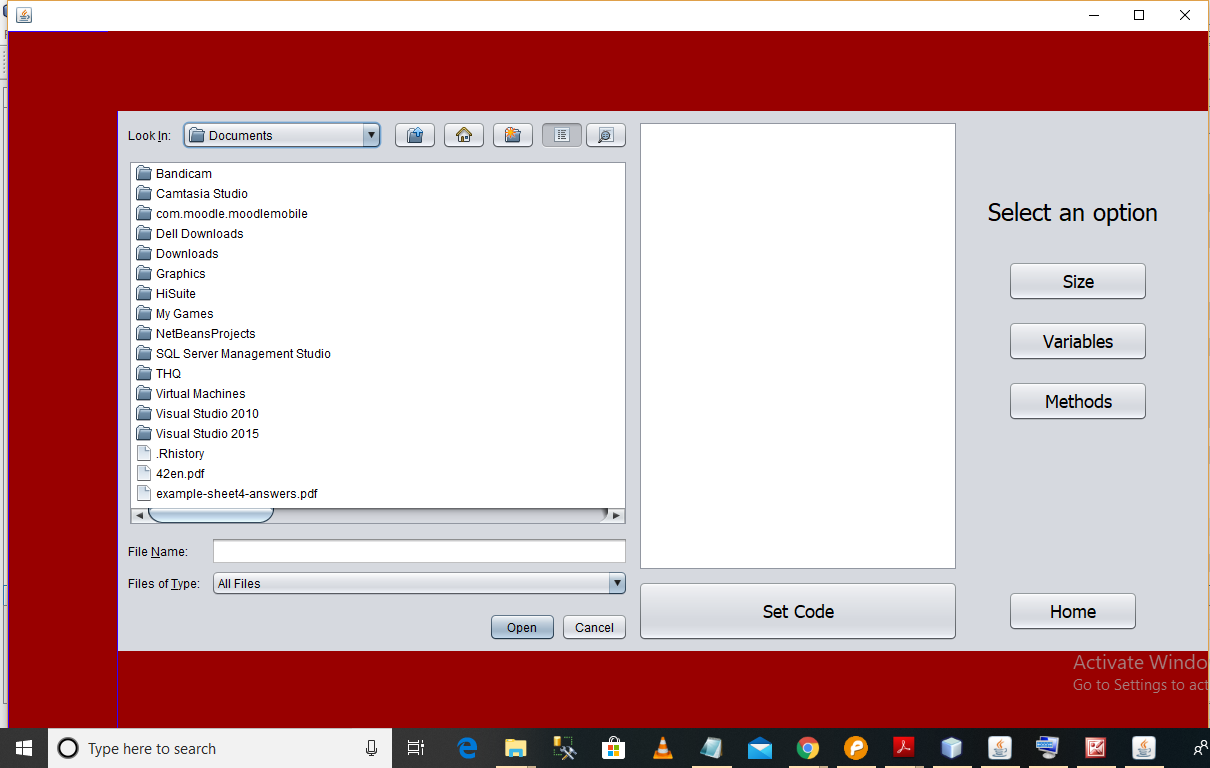


1. This is the dashboard.

2. There are four options available here.

3. User can select one option and click a button according to the option.

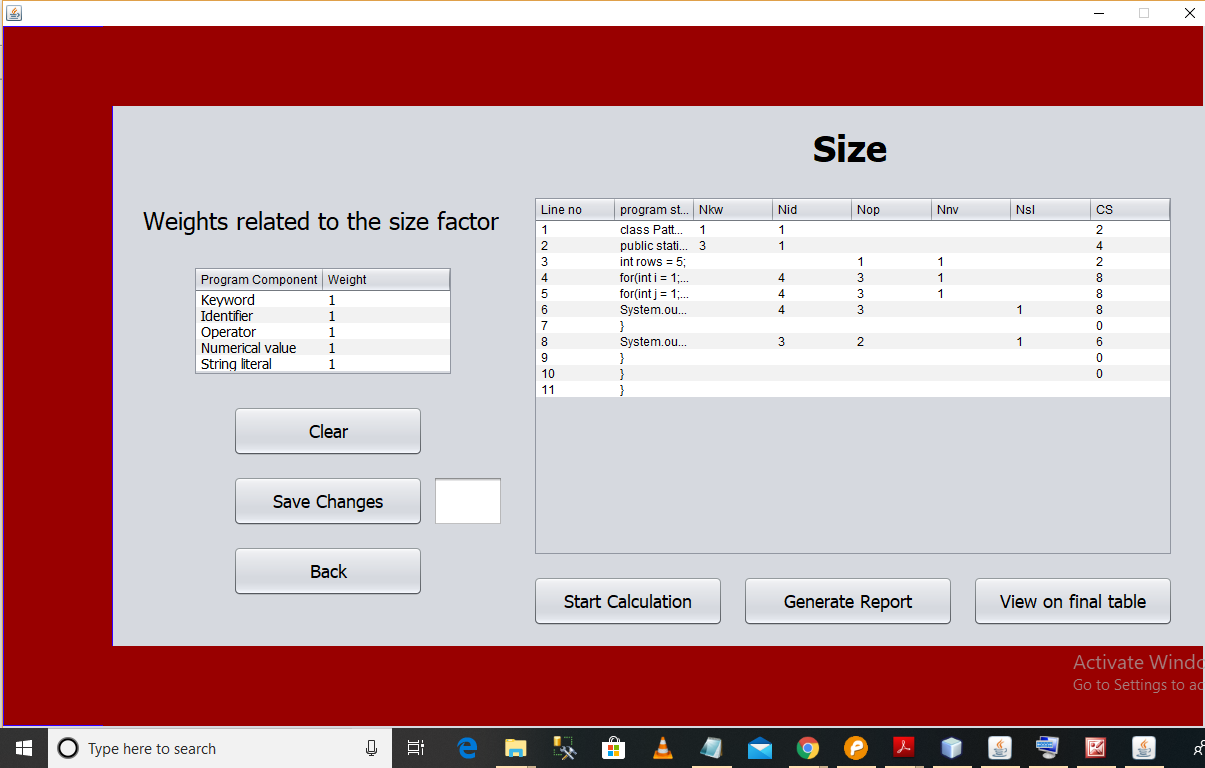
4. After clicking on a button coming forward steps are same for all interfaces. Let’s consider size, variables and methods option.

1. In here we can upload a single file include a code. It is also can be a zip file.

2. We used file chooser for this event. We can browse a file and click ”open” button. Then it will be displayed in the given text field.

3. After that we can select an option.

4. Let’s consider about size option.



1. First we can change the key weights by clicking a row and click ‘save changes’ button.

2. After that we can click start calculation button. Then the filled size table will be displayed according to the key words.

3. If we want get a report here we can get a report by ‘Generate report’ button.

4. Finally we can move onto the final table by clicking ‘View on final table’ button.



1. This is the final table.

2. In here we can view all the calculations below and generate final report.

**2.1** **Complexity of program due to size.**

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

**2.2 Complexity of program due to variables**.

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

**2.3 Complexity of program due to methods.**

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

**2.4** **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)

**2.5 Complexity program due to control structure.**

Ccs = (Wtcs \* NC) + Ccspps