**DESIGN AND IMPLEMENTATION OF COMPUTER-AIDED SYSTEM THAT SOLVES ALGEBRAIC EQUATIONS**

**BY**

**ENYIDI, PRECIOUS KAOSISOCHUKWU**

**U14/NAS/CSC/054**

**Being a B.Sc project report submitted in the partial fulfillment of the requirement for the award of a Bachelor’s degree in Computer Science of the Godfrey Okoye University.**

**Supervisor: Dr. George Okereke**

**Department of Mathematics / Computer Science**

**Godfrey Okoye University, Enugu.**

**JULY, 2018**

**CERTIFICATION**

We hereby declare that the work presented herein was done by me, and not by a third party. Should I be convicted of having cheated in this work, I shall accept the verdict of the University.

**ENYIDI, PRECIOUS KAOSISOCHUKWU, U14/NAS/CSC/054**

**APPROVAL PAGE**

This is to certify that this project work titled the **DESIGN AND IMPLEMENTATION OF A COMPUTER AIDED SYSTEM THAT SOLVES ALGERBRAIC EQUATIONS** is authentic and the research work used for the project has been approved by the supervisor of the project and the head of department, Computer Science, Godfrey Okoye University, Enugu.

………………………………. …………………………..

**Dr. George Okereke Date**

**(Project Supervisor)**

……………………………... ……………………………

**Dr. (Mrs) Monica N. Agu. Date**

**(HOD, Computer Science**)

**DEDICATION**

This project is dedicated to my loving parents Engr. Enyidi Sylvester Obiora and Mrs. Enyidi Margaret Anayo whom I adore and cherish for their for their priceless love, care and understanding. Also to my beloved Bran and Victory who have been the best of siblings, and to my ever loving cousin, Aneke Stella. My Love continues.

**ACKNOWLEDGEMENT**

I remain grateful to my supervisor Dr. George Okereke who took out a portion from his time and his convenience to give me the supervision needed to make this work professional; you will not lose your reward sir.

My appreciation also goes to my Head of Department, Dr. (Mrs) Monica N. Agu for instilling in us the manner and method of handling out academic duties as well as working hard enough to make sure that we as students are given the right treatment. Also to all my departmental lecturers Mrs. Njideka Onos, Mr. Benson Ikenna, Mr. Oguike and Dr. Ndidiamaka Ozofor who were outstanding in their effort to lecture and teach me properly. God Bless you all.

I also wish to acknowledge the invaluable relevance of my former Head of Department, Late Dr. Godwin Anene Madu Ikekeonwu of Computer Sciences. I know you are resting somewhere better right now. While you lived, your personality, discipline and drive were inspirational to me. Even as you are gone, I still remember. Thank you for everything sir; Sleep on.

To my real friends Odinaka, Somto, Maxwell, Jessica, Onyinye, Cynthia, Stephanie, Kizito, Kelechukwu, Leobest, Philip, David, Israel, Ifeanyi, Ifunanya, Ebube, Ugochukwu, Eric, Gladys, Grace, Princess, Ella, Chosen, Etimbuk, Frank, Munachi, Gideon, Daniel, Blessing, Benedict, Jacob, Chioma, Vincent, Louise and Francisca. You all are astoundingly exceptional, remain blessed. I love you all.

Finally, to the class of 2018, Godfrey Okoye University, I remain grateful, it was really a wonderful experience being among you.

**ABSTRACT**

Traditionally, the concept of teaching mathematics has always been a teacher – student relationship; in which the teacher explains the concept of the topic to the student and illustrates it with some examples. The student is then left to understand the topic on his or her own using the tools given by the teacher. A problem often results when the student needs a guide while practicing and the teacher is not available. In that case, learning becomes slow and hindered. As we are in a digital age, where computers have been built to emulate most services usually offered by a human, it is believed that the computer can also stand in the gap for the teacher in his / her absence. With respect to algebra, the objective of this project is the design and implementation of a computer aided system that algebraic equations with limitations to simultaneous equations, quadratic equations and cubic equations (involving real numbers only). The system is designed using Java, CSS (Cascading Style Sheet) and MathTex as the programming languages. The methodology used is the Object-Oriented Analysis and Design method. It is expected that this software would be able to stand in the gap in the absence of the teacher and help students solve algebraic equations on their own, using their own examples and at their own pace and also help teachers in getting versatile knowledge of a algebraic equations by testing them with their own variables. It could also help teachers understand the most optimal methods for solving an algebraic equation to avoid errors in the process of teaching and learning.

**TABLE OF CONTENTS**

Title page i

Certification ii

Approval Page iii

Dedication iv

Acknowledgement v

Abstract vi

Table of contents vii

List of Figures ix

List of Tables x

## CHAPTER ONE: INTRODUCTION

* 1. Background of the study 1
  2. Statement of the problem 2
  3. Objective of the study 3
  4. Significance of the study 3

1.4 Scope of the study 4

**CHAPTER TWO: REVIEW OF RELATED LITERATURE**

2.0 Introduction 5

2.1 Theoretical Background 7

2.2 Review of Related Literature 9

2.3 Summary 24

## CHAPTER THREE: SYSTEM ANALYSIS AND DESIGN

* 1. Introduction 26
  2. Analysis of Existing System 28
  3. Design of the Proposed System 29

3.2.1 Input Design 31

3.2.2 Output Design 31

* + 1. Database Design 32

3.2.4 System Architecture 33

## CHAPTER FOUR: SYSTEM IMPLEMENTATION

4.0 Introduction 35

4.1 Choice of Programming Environment 35

4.2 Implementation Architecture 37

4.3 Software Testing 37

4.4 Documentation 40

4.4.1 User Manual 40

4.4.2 Source Code listing 41

**CHAPTER FIVE: SUMMARY AND RECOMMENDATION**

5.0 Summary 42

5.1 Recommendation 43

**REFERENCES** 44

**APPENDIX** 46

**List of Figures**

Fig 1: Graph of Quadratic Equation showing Real and Distinct roots 17

Fig 2: Graph of Quadratic Equations showing Imaginary roots 18

Fig 3: Graph of a Cubic Equation 24

Fig 4: Software Design Methodology using Object Oriented Analysis and Design 28

Fig 5: Use Case Diagram showing the design of the proposed system 30

Fig 6: Input Design sample using Quadratic Equations 31

Fig 7: Output Design sample using Quadratic Equations 32

Fig 8: System Architecture 34

Fig 9: Eclipse Integrated Development Environment Interface 36

Fig 10: Implementation Architecture 37

Fig 11: Screenshot showing the Homepage 38

Fig 12: Screenshot showing Insertion of Variables 39

Fig 13: Screenshot showing Choice Method 39

Fig 14: Screenshot showing Output 40

**List of Tables**

Table 1: Database Structure for Simultaneous Equations Design 33

**CHAPTER I**

**INTRODUCTION**

1. **Background of the study**

Algebra is a field of mathematics that together with number theory, geometry and analysis, is the study of mathematical symbols and the rules for manipulating those symbols. It is a unifying thread of almost all mathematics. As a result, it includes everything from elementary equations, to the study of abstractions such as groups, rings and fields. Algebra is divided into two main parts; elementary and abstract or modern algebra. Elementary algebra encompasses some of the basic concepts of algebra, and is often used to build one’s understanding of arithmetic (dealing with specified numbers) by introducing quantities without fixed values (called variables). Elementary algebra is mostly concerned with structures within the realm of real and complex numbers. Abstract or modern algebra is the study of algebraic structures such as groups, rings, fields, modules, vector spaces, lattices etc.

Algebraic equations are needed in many aspects of life such as engineering, industry, medicine etc. As a result, we end up solving algebraic equations almost every day as we have to make decisions about specific quantities such as the amount of food to last a week, amount of materials needed for construction of a block in a site, amount of money needed to follow up a project from start to finish etc. As solving algebraic equations manually can be tiring or time-consuming; which is a consequence of the bulky steps one has to pass through that increases as the complexity of the equation increases, this project illustrates the construction of a desktop application that simulates and solves systems of algebraic equations and shows the user the algorithm followed by the computer in solving such equations.

**1.1 Statement of the Problem**

The following problems are observed in the manual solution of systems of algebraic equations;

* Time Conservation: Manually solving an algebraic equation from start to finish can be time consuming especially in cases where the equation is as complex as a quadratic or cubic equation or an exponential equation with long procedures.
* Cost: One who wants to solve an algebraic equation prefers manual solution using a calculator, a pen, a piece(s) of paper as well as four-figure tables in order to get his / her facts right. Thus, a project analyst in an industry would need a stand-by calculator, stacks of paper as well as a writing pen, all of which are costly to constantly supply and exhausts space.
* Accuracy: Man always has the tendency to make errors as a result of extensive approximation. Example, an average individual tends to solve mathematical problems with values not more than 3-4 decimal places. This can cause significant errors when used in the long-run.

**1.2 Aims and Objectives**

The aim of this project is to develop a computerized solution for a system of equations, the specific objectives are to;

* Reduce the time and energy exhumed in the process of solving algebraic equations manually.
* Help students solve algebraic equations on their own without the constant presence of a school teacher or the constant usage of physical textbooks, as well as alleviate the stress of having to carry too many learning materials while going for studies.
* Minimize the cost of analytic materials; in a data analyst’s office one workstation or desktop could carry as many mathematical problem solving applications as possible, which reduces the cost of buying, writing and solving materials such as the calculator, papers, pen etc.
* Aid teachers and examination bodies in the preparation of questions and the construction of error-free marking schemes.

**1.3 Significance of the study**

The beneficiaries of this project are;

* Science students involved in mathematics
* Mathematics teachers

Every science student needs an in-depth understanding of mathematics for any significant goal is to be achieved in his / her study. This project would provide a reliable means of sourcing for help in mathematical problems involving algebraic equations. It would help the student to solve algebraic equations with high degree of accuracy, sighting norms and exceptions, as well as rules to be followed.

Mathematics teachers would benefit widely because they no longer have to rely on the limited examples textbooks offer them, but they can try as many problems as possible to expand their understanding of the algebraic equation to be solved and hence increase their efficiency while teaching.

**1.4 Scope of the Project**

This project covers three main types of algebraic equations, namely;

* Simultaneous Equations which could involve;
  + Two Linear equations
  + One Linear and one quadratic equation
  + One quadratic and one cubic equation
  + One Linear and one cubic equation
* Quadratic Equations
* Cubic Equations

It is limited to real numbers, meaning that complex numbers, trigonometric functions and exponential functions are not considered in this context.

**CHAPTER II**

**LITERATURE REVIEW**

**2.0 INTRODUCTION**

In Mathematics, an equation is a statement of equality between two expressions [1], usually containing variables. Solving the equation consists of determining which values of the variables make the equality true. Variables are also called unknowns and the values of the unknowns that satisfy the equality are called the solutions of an equation. The most common types of equations are algebraic equations, in which two sides are algebraic expressions. Each side of an algebraic equation will contain one or more terms. [2]

Examples include; and etc. There are vast types of algebraic equations as far as the field of algebra is concerned. But we are most conversant with the following types;

* Linear Equations: An equation in which each term is either a constant or the product of a constant and a single variable [3]. In its simplest form, its structure looks like this;

*ax + b = 0*;

Linear equations can come in different sizes; in most cases the difference between them is often the number of the unknowns. Hence, we have

* + Linear Equation in one variable;
    - 7x + 7 = 15
    - 8x + 9 = 22
  + Linear Equation in two variables;
    - 7x + 2y = 15;
    - 2y + 3z = 5;
  + Linear Equation in three variables;
    - 2x + 3y + 4z = 20;
    - x + y – z = 9;

If you noticed, the highest power of the unknowns is 1 [3].

* Quadratic Equations: A quadratic equation is any equation having the form

Where x represents an unknown, and the values a, b, and c represents known numbers such that a is not equal to 0. If a = 0, then the equation is linear. The numbers a, b and c are the co-efficients of the equation and may be distinguished by calling them the quadratic coefficient, linear coefficient and the constant or free term. [2].

* Cubic Equations: A cubic equation is one which can be represented in the form

Where a is a non-zero number. (If a = 0, we have a quadratic equation). A cubic equation has three roots, which may be equal or not equal [4].

**2.1 THEORETICAL BACKGROUND**

The tools and technologies used in this project are;

* Java programming language,
* Cascading Style Sheets (CSS).

The Integrated development environment used here is the Eclipse Enterprise Edition IDE. The program developed is a collection of Java classes i.e Files written and compiled in Java programming language.

* **Java Programming Language:** Java programming language is a general-purpose computer-programming language that is concurrent, class-based and object oriented [5]. It is specifically designed to have few implementation dependencies as possible. It is intended to allow application developers “write once, run anywhere” (WORA), this means that compiled java code can run all platforms that support java without any need for recompilations. In other words, the applications can run all platforms as long as they support java, irrespective of the computer architecture. To do this, the every computer architecture that supports java has a java virtual machine (an emulation of a computer system that allows programs written in java or programs written in another language but compiled into java format to run on any platform, irrespective of the architecture) [6]. As at 2016, Java was one of the most popular programming languages in use, with an estimated 9 million developers, and it is mostly used for most client-server applications [7]. Most of its syntax (grammar that makes up the language), is derived from C and C++ programming languages.

In this application, it is the core language used. The Graphical User Interface (GUI) that allows the user to see and interact with the program is written in Java.To import the Java Library into the IDE (Integrated development environment), we use this syntax [5];

**Import //library;**

The basic JavaFx syntax looks like this [5];

**Public class** newClass {

**Public void** start (**final** Stage stg) {

//your code

}

**Public static void** main(String[] args) {

Application.*launch*(args);

//The code above launches the program

}

}

* **Cascading Style Sheets (CSS):** The cascading style sheet is the core scripting language used for adding styles and colors as well as animation to the program [8]. It became very popular with website designers because it was used to add styles, colors and animation to html (Hyper-text markup language) tags and texts.

However, it is implemented in the Java Class like this [8];

Java Selector = new Java Selector();

Selector.setStyle(“-fx-property: value;”

+ “-fx-property:value;”);

Example:

Menu mbc = **new** Menu("Help and Tutorials");

mbc.setStyle("-fx-font-family: Times;"

+ "-fx-font-size: 15;"

+ "-fx-font-weight: BOLD;");

* **Structured Query Language (SQL):** Structured Query Language is a standard computer language for relational database management and data manipulation. It is used to query, insert, retrieve update and modify data. In this project, it will be used to insert into and retrieve user data from the database management system[9]. The basic SQL syntax is given below [10];

**SELECT \* FROM (Database\_table);**

**2.2 Review of Related Literature**

In this project, algebraic equations treated are;

* Simultaneous Linear Equations
* Quadratic Equations
* Cubic Equations

1. **Simultaneous Equations:** A typical simple simultaneous equation looks like this [11]:

Where **x**  and **y** are variables and **z** is a constant. In this type of equation, the solutions of **x** and **y** must be uniform for both equations and must satisfy both equations.

The methods used in simultaneous linear equations for this project are as follows [11]: Elimination Method

* + Substitution Method
  + Graphical Method

**Elimination Method** involves one or two unknowns being removed (made to be non-existent) either by addition or subtraction being performed on the two equations [11]. For instance, in the pair of equations given below like;

Solving the equation above requires us to make a decision on which variable to eliminate first. In this case, if we chose **y**, we make **x** the subject of the formula by eliminating **y**. To do this, we have to make the co-efficients of y uniform by multiplying equation(1) by 2 and multiplying equation(2) by 1. For which we get;

Then we eliminate to get;

If,

-x = -2, then x = 2;

Having known that x = 2, we insert this into equation\_1 to get;

2 + y = 5

y = 5 – 2

y = 3;

Hence, x = 2 and y = 3.

**Substitution Method**: In this case, we make any of the variables (**x** or **y**) the subject of the formula in any of the equations (many prefer equation1) then transfer the right-hand-side of that expression to represent the subject variable in the second equation. For instance,

Make **x** the subject of the formula to get;

Use the RHS (Right-hand-side) which is **10 – y** and use it to represent x in equation(II) to get;

Then we simplify to get;

-y = -3

y = 3;

Having got the value of **y**, we insert the value into equation(I) to get;

x = 5 – 3

x = 2

**The Graphical Method** makes use of graphs to find co-ordinates that match each other by plotting the graphs of each linear equation [12]. The aim is to find that point where the two graphs intersect; that point is the solution of the simultaneous equation. For instance, look at this equation;

Using graphical method, first we have to

* + Create tables for both equations and assume values for **x**,
  + Then use the assumed values to find the values of **y.**
  + After which we would plot the graphs
  + Then find the intersecting points of the graphs [12]

Putting it to practice, we have;

For Equation[I], create a table

For values of x ranging from -2 to +6, we calculate y = 5 – x for all values of x.

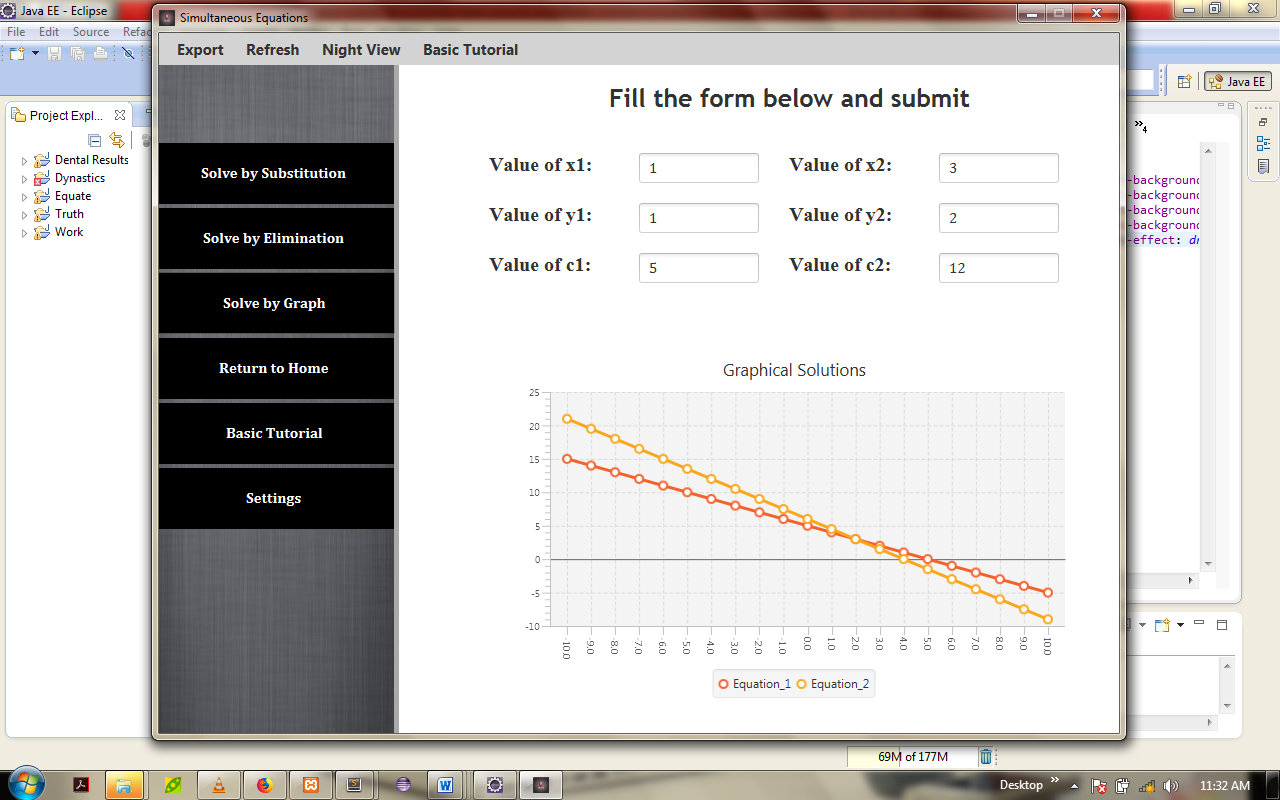
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| y | 7 | 6 | 6 | 4 | 3 | 2 | 1 | 0 | -1 |

For Equation[II], create a table

For values x ranging from -2 to +6, we calculate y =

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| y | 9 | 6 | 6 | 4 | 3 | 2 | 1 | 0 | -1 |

A Graphical representation of this would be



From the graph above, we get to see that the intersection occurs at;

* **y** = 3 and
* **x** = 2

Hence, the solution to the equation is **y** = 3 and **x** = 4.

1. **Quadratic Equations:** A quadratic equation is any equation having the form

Where *x*  is the unknown and *a*, *b* and *c* represent known numbers such that a is not equal to 0 (zero). Otherwise it becomes a linear equation [2].

The roots of a quadratic equation are of four types;

* + Real: greater than or equal to 0.
  + Imaginary: roots less than 0.
  + Rational: the root is the square of a rational number.
  + Irrational: is not the square of a rational number, hence has to expressed in surd form [2].

To be able to detect the type of root a quadratic equation would have, we usually start by finding the discriminant, which is given by the formula [2];

For instance, if we have a quadratic equation like;

Where,

***a = 2, b = 3 and c = -5.***

The Discriminant would be

***=***

The discriminant is 7 which is a real number and is greater than 0, hence the roots of this equation are real.

In this project there are three main methods of solving quadratic equations involving real numbers [13]. They are;

* + **Quadratic Formula:** The quadratic formula is [13];

Example, if we have a quadratic equation as described above;

Solving using the quadratic formula would give;

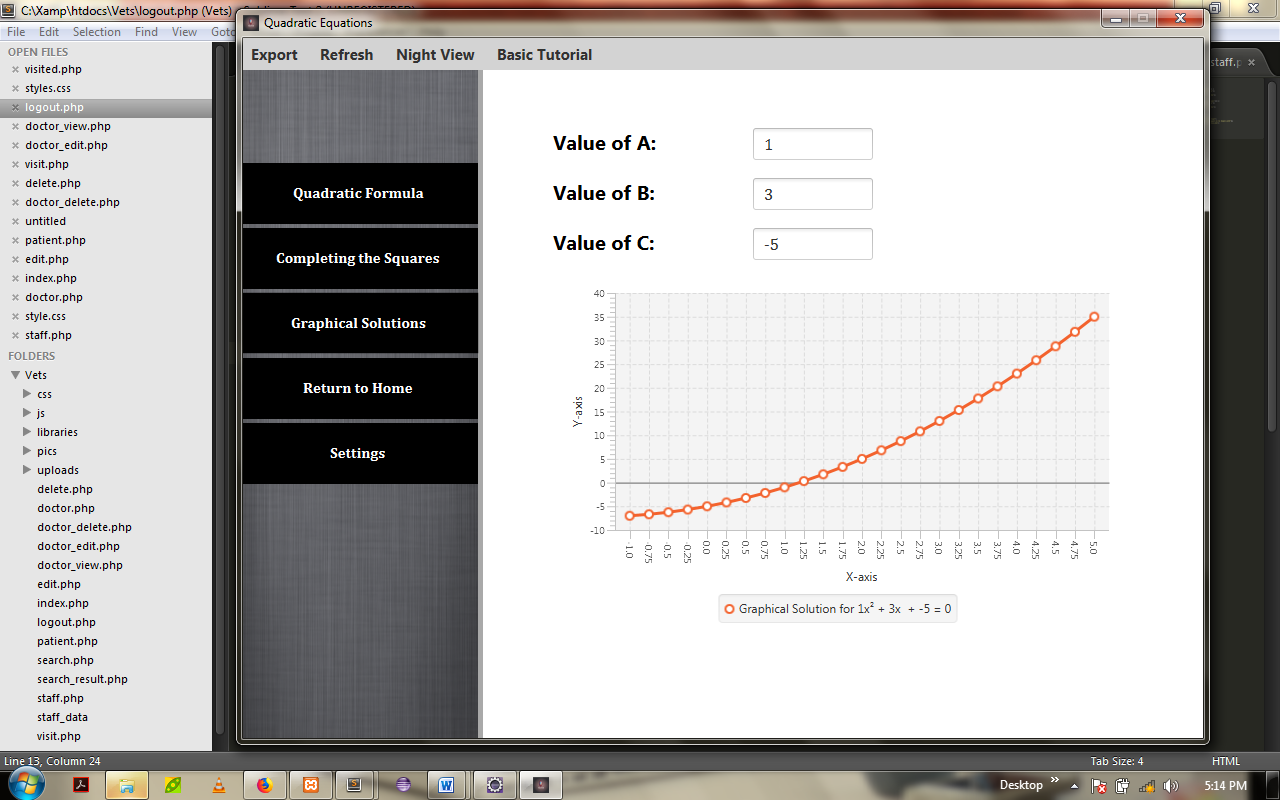
Simplify further to get,

* + **Completing the Square Method:** The method of completing the square is a much more detailed method than the quadratic formula. Chronology has it that it was used to derive the current quadratic formula from the equation[11];

If we have an equation like this;

The steps to solve it goes this way[12];

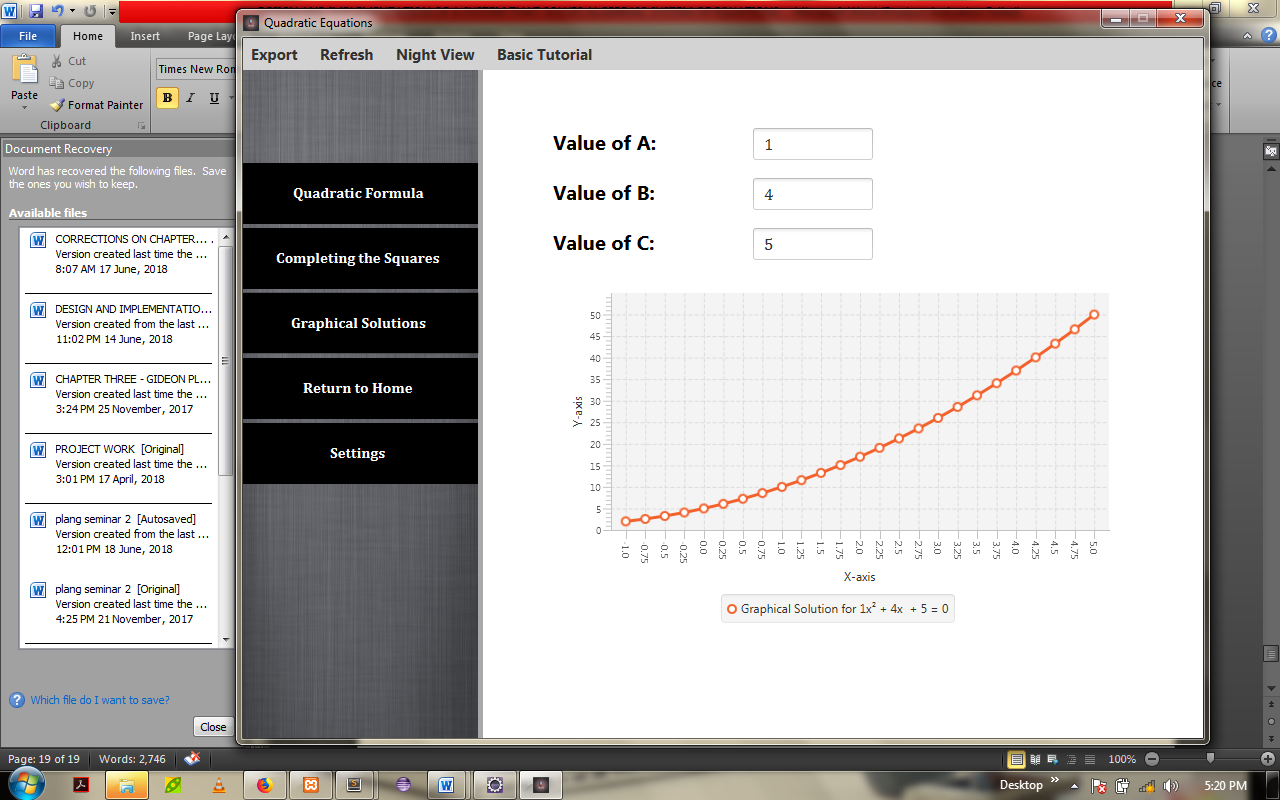
* + - **Step 1:** Move the constant to the Right Hand Side;
    - **Step 2:** Divide through by the co-efficient of ***x2*** (which is 1).
    - **Step 3:** Add half the square of the co-efficient of x to both sides and factorize;
    - **Step 4:** Simplify further to get;
  + **Graphical Method:** The graphical method uses the intersection of co-ordinates in a graph to determine the roots or the solutions to a quadratic equation[12]. The graphical method is often the most perfect way of solving quadratic equations in the sense that it is often faultless, easy to plot and often more detailed than the other methods[12]. The graphical method gives us curves with about 3 distinct characteristics depending on the nature of the roots of the equation.
    - **Real and Distinct Roots;**



**Fig 1: Graph of Quadratic Equation showing Real and Distinct roots**

Here, the curve cuts through the x-axis; meaning that the roots are real and distinct. Examples of equations like this include;

* + - **Imaginary Roots;**



**Fig 2: Graph of Quadratic equation showing imaginary root**

Here, the graph doesn’t cut through the x-axis, instead, its slightly above it (in some cases, far above it). Here, the roots of the equation are imaginary. Examples of quadratic equations like this include;

1. **Cubic Equations:** A cubic equation’s unknown has its highest power as 3[14]. It often has one real number answer and two imaginary answers. In this project however, we deal mostly with the real number answers (Although imaginary numbers could come in occasionally).

The general mode of cubic equations is in this format[14];

Where ***d*** is the constant.

The methods used for cubic equations in this project include;

* + **Quadratic Formula:** Reducing the cubic equation to a quadratic one, provided that the constant (*d*) is zero[15].
  + **Cubic Formula:** Using the general formula for solving cubic equations. Usually arrives at one answer[15].
  + **Graphical Method:** The graphical method is simply representing the cubic function on a graph. The solution to the equation lies at the points where the curve intersects with the x-axis. Here, the solution is devoid of assumptions, and is complete (Has three distinct answers)[15].

These methods will be explained below;

**Quadratic Formula:** This formula reduces the cubic equation to a quadratic one by factorizing out the cubic power of *x*[15]. For example, to solve an equation such as the one below;

We find out that the constant there is zero. So, reducing it by factoring out *x* yields a quadratic equation. So, we go-ahead to get;

We solve the quadratic equation in the bracket using the quadratic formula which is;

Insert the values of the quadratic into this formula to get;

Simplify further to get;

Because a cubic equation usually has three possible answers, we have found, two. Therefore, we make the third answer zero. Thus, the solution to the above cubic equation is;

**Cubic Formula:** The cubic formula is the general formula for solving cubic equations. It usually results to just one real number answer, which is why it is referred to as “solving the depressed cubic”, and the presence or absence of the constant is not noticed [16]. The cubic formula is given thus;

Where,

And

Hence, if we have a cubic equation such as;

Where

We find the value of A to be;

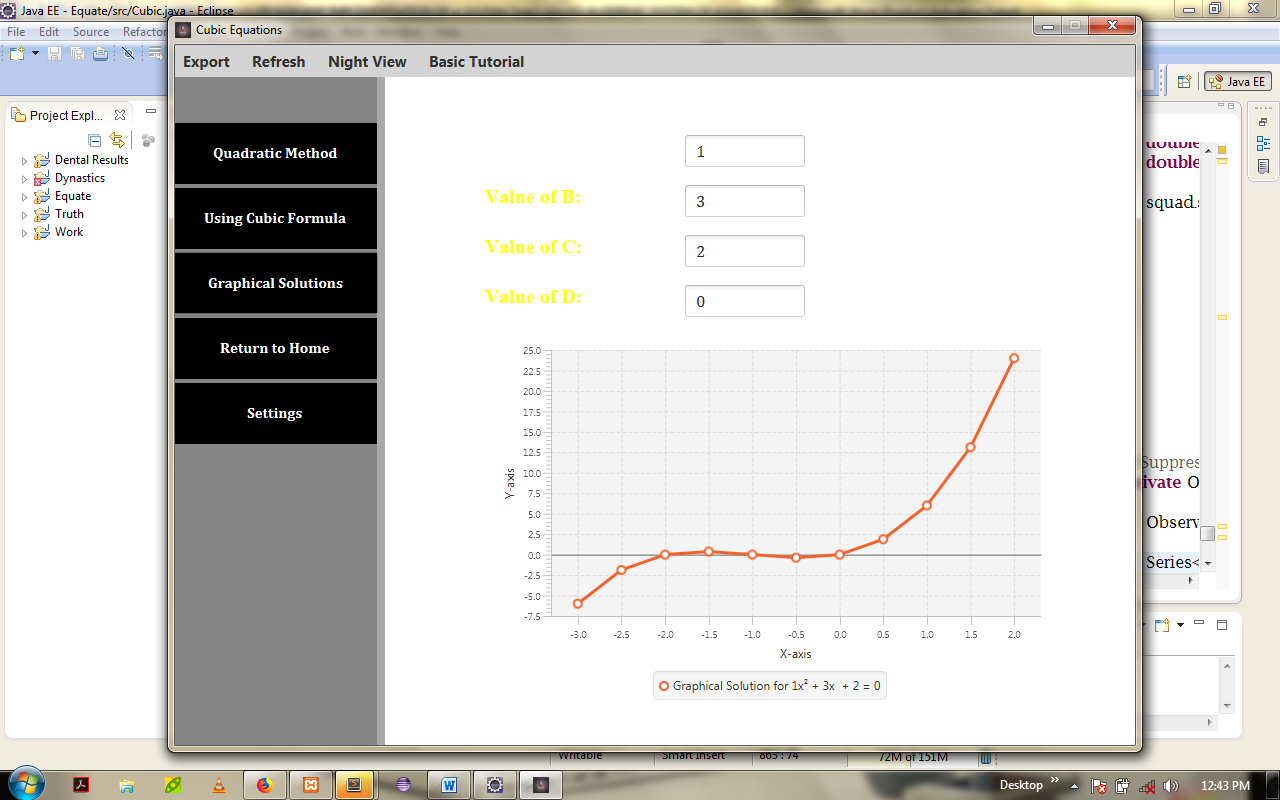
We find the value of B to be;

Having found the values of **A** and **B**, we find the value of ***x*** to be;

Hence, we have a single answer which satisfies the equation.

**Graphical Method:** This method makes use of graphs to find the solutions of the cubic equation. It’s usually a curve that is meant to cut the x-axis in three places (when constant is zero) or less, the points of intersection gives us the solution to the equation [16]. An example is a cubic equation given below as;

Solving with the graphical method, we have;



**Fig 3: Graph of a Cubic Equation**

From our diagram above, we find out that the graph intersects the x-axis in three points, namely ***0,-1,-2*.** So, we take these values to be the solutions of the cubic equation.

**2.3 Summary**

This project is designed to be a desktop application implemented using Java and MathTex libraries as well as Cascading Style Sheets.

The algebraic equations dealt with in this context are;

* simultaneous,
* quadratic and
* cubic equations.

The simultaneous equations are solved using the following methods;

* Substitution method
* Elimination method
* Graphical method

The quadratic equations are solved using the following methods;

* Completing the square method
* Quadratic Formula and
* Graphical method

The cubic equations are solved using the following methods;

* Cubic Formula
* Quadratic Formula and
* Graphical Method

**CHAPTER THREE**

**ANALYSIS AND DESIGN OF THE SYSTEM**

**3.0 INTRODUCTION**

This chapter discusses the current procedure used in solving algebraic equations, then proposes the model the project uses to remedy the errors and difficulties caused by the existing system. It also shows the system architecture of the proposed system and the methodology the proposed system intends to use in solving current problems.

The methodology used in development of this project is the Object Oriented Analysis and Design Method (OOADM). The object oriented model is the construction of objects that contain stored values of the instance variables found within an object. Object oriented values are solely objects [17].

The phases in object-oriented software development include [18]:

1. Object Oriented Analysis: In this stage, a problem is formulated, user requirements are identified, then a model is built based upon real world objects. The analysis produces models on how the desired system should function and how it should be developed.
2. Object Oriented Design: This has two main stages namely:
   1. System Design: The complete architecture of the desired system is designed. System analysis emphasizes on the objects comprising the system rather than the processes in the system. The system is conceived as a set of interacting subsystems that in turn is composed of a hierarchy of interacting objects, grouped into classes.
   2. Object Design: In this phase, a model is developed based on the models developed in the system analysis phase and the architecture designed in the system design phase. All the classes required for the software are identified. It is here that we have to define the associations between the identified classes and the hierarchies involved. The designer also designs the internal details of the classes and their associations. (Data structure of attributes and algorithms for the operations).
3. Object Oriented Implementation: In this stage, the design model developed in the object design is translated into code in an appropriate programming language or software tool. The databases are created and the specific hardware requirements are ascertained. Once the coding is completed, it is tested using specialized techniques to identify and remove the errors in the code.

The diagram below visualizes the methodology used in the development of this software.

**Object Oriented Analysis**

**Object Oriented Design**

**Object Oriented Implementation**

**Testing**

**Evolution**

**Fig 4: Software Design methodology using Object Oriented Analysis and Design**

**3.1 Analysis of the Existing System**

In the existing system;

* The student or the teacher reads about algebraic equations and tries to understand examples given in written textbooks.
* The student or teacher uses the knowledge of understanding from textbooks to try solving other examples given in the textbook.
* The student or teacher checks the back of textbooks for correct answers to the questions given in the textbook.

After analysis, the following problems were found

* The student / teacher is not allowed to test as multiple values as possible, instead, he is limited to the content the textbook can offer.
* Manual solution to algebraic equations can often give errors which may not be noticed until much later.
* Sometimes the textbooks do not offer the correct answers due to errors in publication and printing, and such wrong answers could be misleading to both the teacher and the student.

The analysis of the existing system suggests that the need for a computer based solution is imminent. This system has to fulfill the following requirements;

* Allow the user to learn at his own pace, and with his / her own variables.
* Show the user the step by step method of solving each equation.
* Test as many values as possible so as to improve the understanding of the algebraic equation; including the most suitable values for a method and cases where a method cannot be used.
* Act as a check for the user to know the level of understanding of the algebraic equation in question.

**3.2 Design of the Proposed System**

The proposed system allows the user to learn at his own pace, with his own examples. In this way, the user chooses the examples he or she wishes to use to learn the algebraic equation by querying the software and having the answers displayed as well as the process involved in getting the answer.

1. The user selects a particular type of algebraic equation.
2. The user enters values to be solved and selects the method to be used.
3. The software gathers the values entered; checks for compatibility and solves the equation using the chosen method.
4. The procedure followed and the answers are displayed for the user to see.

The use case diagram below describes the process;

**Consumer**

**Registered Consumer**

**New Consumer**

**Authentication**

**Fig 5: Use case Diagram showing the design of the proposed system**

**3.2.1 Input Design**

The input design used here is the form. The contents of the form will be determined by the type of algebraic equation. Example, the form for quadratic equations would only contain 3 entries for **a**, **b** and **c** in the equation as described below;

**Fill the Form Below and select a method to solve**

**Value of A:**

**Value of B:**

**Value of C:**

**QUADRATIC METHOD**

**COMPLETING THE SQUARE**

**GRAPHICAL METHOD**

**Fig 6: Input Design sample using Quadratic Equation**

The design above is for quadratic equations, which usually have the values **a**, **b** and **c** (constant) as the coefficients of the equation. To begin, the user enters the values for **a**, **b** and **c** for the equations and selects the method of his/her choice from the list of available methods.

**3.2.2 Output Design**

When the user is done with entries and selects the method of his/her choice, the application gathers the values and solves the equation using that method. The results are displayed in a scroll pane below the form for the user to read through. The example below is for a quadratic equation solved using the quadratic formula method.

**x**

**z**

**y**

**Fill the Form Below and select a method to solve the quadratic equation**

**Value of A:**

**Value of B:**

**Value of C:**

**QUADRATIC FORMULA**

**COMPLETING THE SQUARE**

**GRAPHICAL METHOD**

**Using the Quadratic formula……..**

**Solved Equation and Displayed Input goes here…..**

**(Solved Equation and displayed output)**

**Fig 7: Output Design sample using Quadratic Equation**

**3.2.3 Database Design**

The database architecture is relational (i.e it makes use of tables). Every user of the software is entitled to an account (he or she must register or login to use the software). Each table in the database is assigned to a particular type of equation. Thus, simultaneous equation and quadratic equations have different tables. Hence, when the user enters some values for a particular equation and selects a method of solving it, the software collects the values, method used, the user’s id and the solution derived and sends it to the database. The structure below describes the database design for simultaneous equations.

**Table 1: Database Structure for Simultaneous Equations Design (standard database design)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Type** | **Attribute** | **Null** | **Default** | **Extra** |
| **1** | **Id** | **Int(100)** |  | **No** | **None** | **AUTO\_INCREMENT** |
| **2** | **Var\_x1** | **Varchar(200)** |  | **No** | **None** |  |
| **3** | **Var\_x2** | **Varchar(200)** |  | **No** | **None** |  |
| **4** | **Var\_y1** | **Varchar(200)** |  | **No** | **None** |  |
| **5** | **Var\_y2** | **Varchar(200)** |  | **No** | **None** |  |
| **6** | **C\_1** | **Varchar(200)** |  | **No** | **None** |  |
| **7** | **C\_2** | **Varchar(200)** |  | **No** | **None** |  |
| **9** | **Sol\_x** | **Varchar(200)** |  | **No** | **None** |  |
| **10** | **Sol\_y** | **Varchar(200)** |  | **No** | **None** |  |
| **11** | **Date\_of\_entry** | **Timestamp(6)** | **On\_update CURRENT\_TIMESTAMP** | **No** | **CURRENT\_TIMESTAMP** | **ON UPDATE CURRENT\_TIMESTAMP** |
| **12** | **Method** | **Varchar(200)** |  | **No** | **None** |  |
| **12** | **User\_id** | **Varchar(200)** |  | **No** | **None** |  |

**3.2.4 System Architecture**

The architecture of this system is designed to take minimal memory space from the computer system. The only memory space used for the software is that taken during the equation solving process. Because it has no database design, there is little memory space taken out for the storage of values or procedures and there is only one user, no administrator, no need for internet or server connectivity. Hence, the cost of maintaining this software is highly reduced.

**CHOICE OF EQUATIONS**

**DATA ENTRIES**

**CHOICE OF FORMULA / METHOD**

**OUTPUT**

**KNOWLEDGE BASE**

**EQUATION SOLVED**

**Fig 8: System Architecture**

**CHAPTER 4**

**System Implementation**

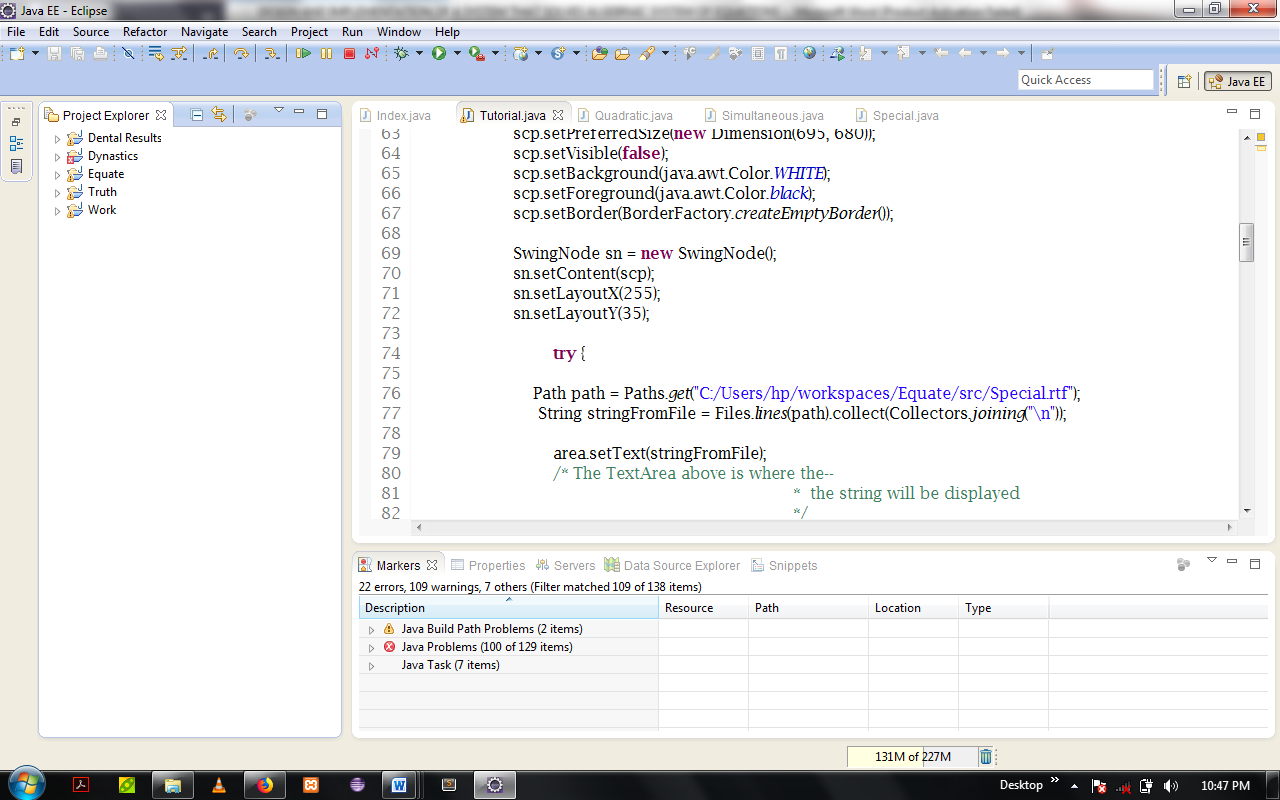
* 1. **Introduction**

System implementation deals with the testing and debugging of the implemented design of the software in process. Here, the choice of environment used is shown, the architecture used for the implementation is explained and the software is tested at each level of construction to test for efficiency and discover possible technical defects. The conversion of the software and documentation is also done at this level.

**4.1 Choice of Development Environment**

The development environment for software consists of the following components;

1. Operating System: this is the low-level software that supports a computer's basic functions, such as scheduling tasks and controlling peripherals. The operating system used as the platform for this project is the windows operating system (specification: Windows 7). Hence, it can only run as a windows desktop application.
2. Integrated Development Environment: An integrated development environment (IDE) is a software suite that consolidates the basic tools developers need to write and test software. Typically, an IDE software contains a code editor, a compiler or interpreter and a debugger that the developer accesses through a single graphical user interface (GUI). An IDE may be a standalone application, or it may be included as part of one or more existing and compatible applications. The IDE used for implementation of this project is the Eclipse IDE.



**Fig 9: Eclipse Integrated Development Environment Interface**

1. Programming Language: A programming language is a vocabulary and set of grammatical rules for instructing a computer or computing device to perform specific tasks. The programming language used for the implementation of this project is the Java programming language which is combined with additional libraries like CSS (Cascading Style sheet) and MathTex Java plugin.

**4.2 Implementation Architecture**

**FRONT-END**

**LOGIN PAGE (JAVA, CSS, SQL)**

**HOMEPAGE (JAVA, CSS)**

**USER CHOICE OF EQUATIONS (JAVA, CSS)**

**VALUES ENTERED FOR EQUATION (JAVA, CSS)**

**EQUATION SOLVED (JAVA, CSS, MathTex)**

**OUTPUT DISPLAYED (JAVA, CSS, MathTex)**

**SQL**

**(database)**

**Knowledge Base: Algorithm (JAVAFX, MathTex, JLatexMath)**

**BACKEND**

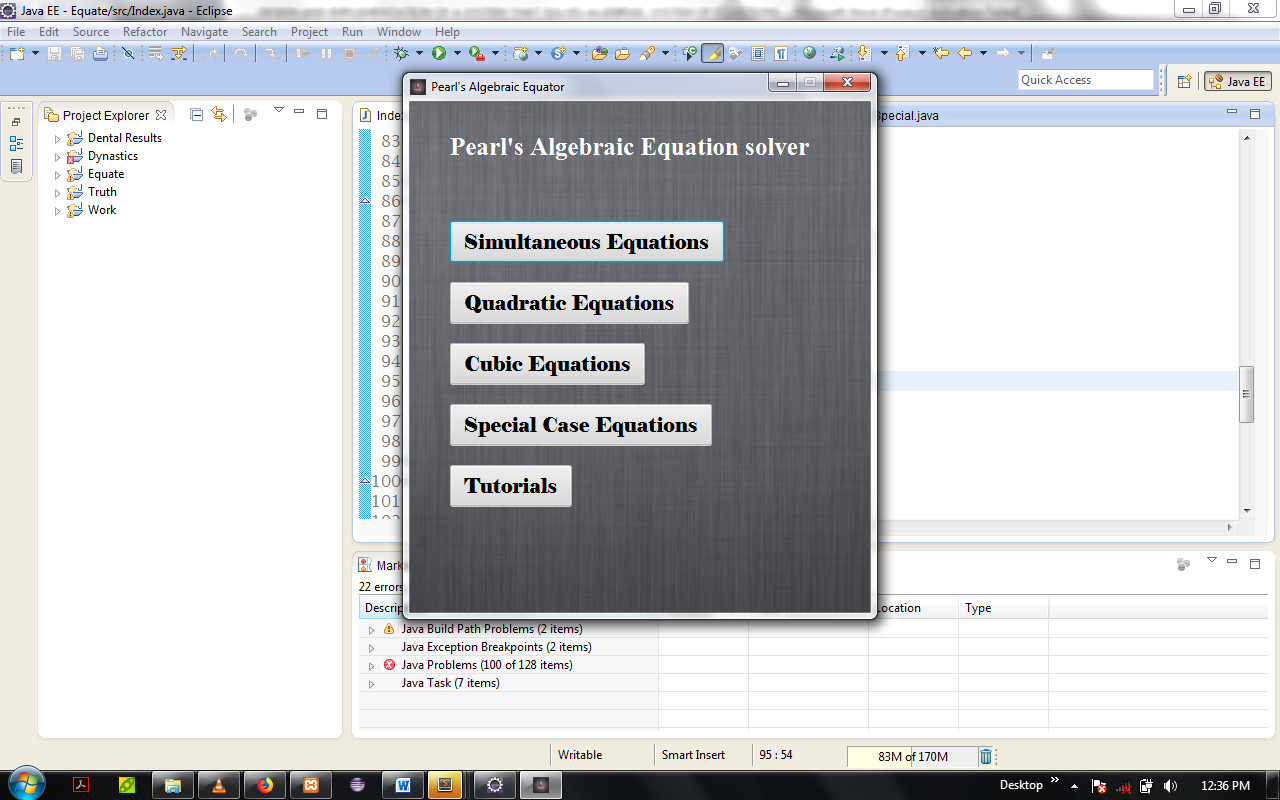
**Fig 10: Implementation Architecture**

**4.3 Software Testing**

Here, the researcher tests the project at the different stages of implementation to check for efficiency and detect possible flaws or defects.

The following aspects would be discussed in software testing;

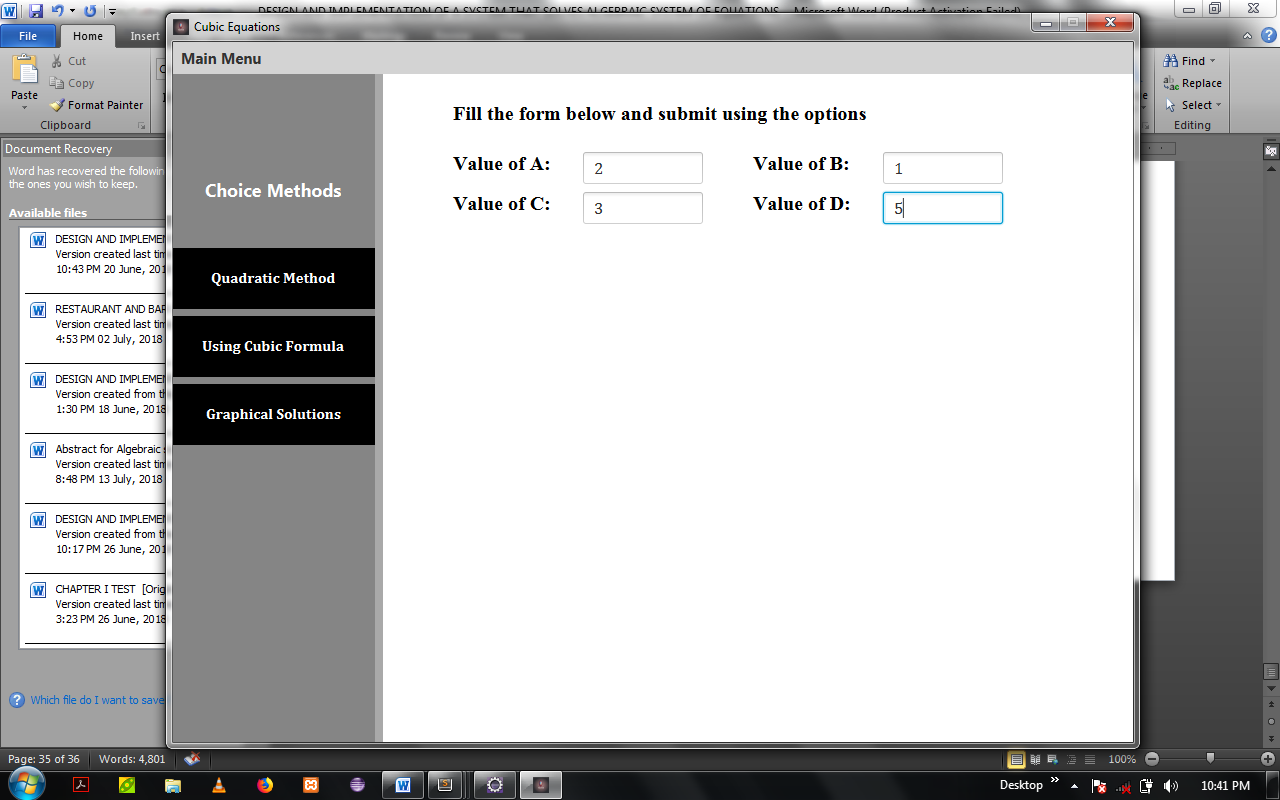
**1. The Homepage:** The Homepage displays the welcome banner to the user and the list of equations it has the capability to solve. The diagram below displays the user interface design for the homepage:



**Fig 11: Screenshot showing Homepage**

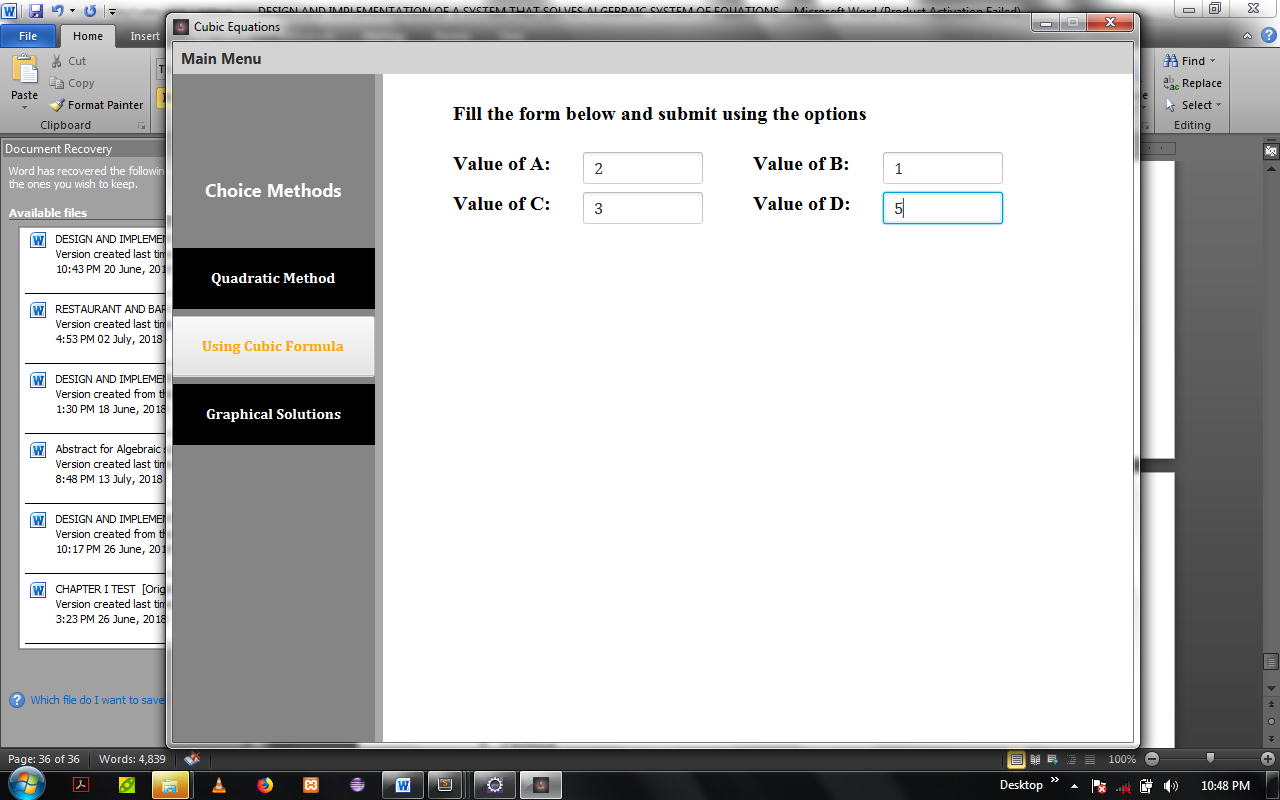
**2. The Choice of Equations:** In this section, the user selects a type of algebraic equation he or she would wish to work on. In this case, we would use Cubic equations to illustrate it.

**3. Insertion of variables:** The user is instructed to fill a form that contains the values for the cubic equation. In this case, we know that a cubic equation has coefficients a,b,c and d.



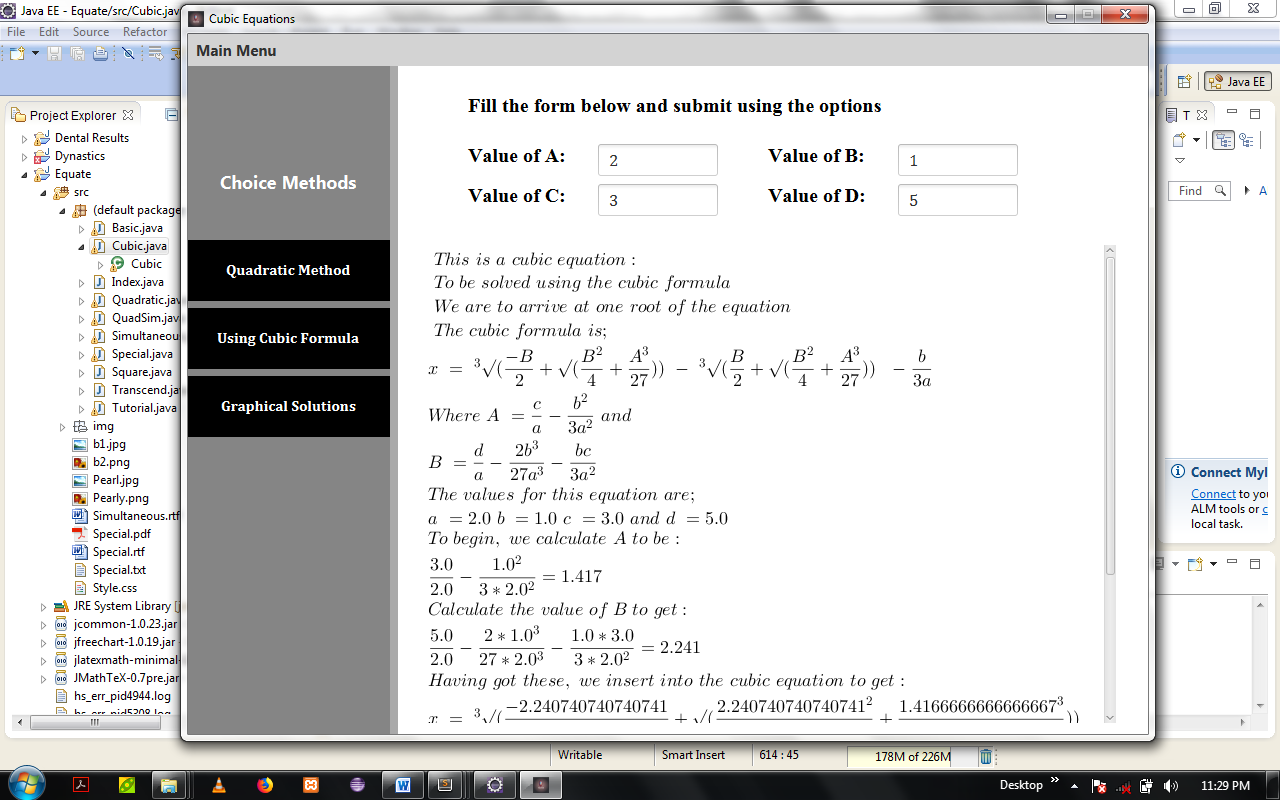
**Fig 12: Screenshot showing Insertion of Variables**

**4. Choice of Method:** The user is prompted to select a method of choice for solving the equation entered in the form. When chosen the application picks the data in the form and begins to solve it. In this case, we’ve chosen the cubic formula.



**Fig 13: Screenshot showing the choice of Method**

**5. Output:** Here, the solution to the equation is displayed as well as the procedures taken to solve it.



**Fig 14: Screenshot showing output**

**4.4 Documentation**

**4.4.1 User Manual**

To begin, follow these steps;

1. Make sure you are using a Microsoft windows operating system that supports Java.

2. Double click on the application icon and wait for the application to load.

3. When the application loads, select any algebraic equation of your choice and start solving questions.

**4.4.2 Source Code Listing**

1. The source code listing for the Homepage (or Index page) is in Appendix A.

2. The source code listing for Simultaneous Equations is in Appendix B.

3. The source code listing for Quadratic Equations is in Appendix C.

4. The source code listing for Cubic Equations is in Appendix D.

**CHAPTER FIVE**

**CONCLUSION AND RECOMMENDATION**

**5.1 Summary**

In conclusion, the software is built for students and is meant to solve algebraic equations explicitly and show the student the step-by-step method of solving the equation. It is also meant to allow the student learn at his or her own pace, using his / her variables. The software also offers database support such that the user is able to view past work done and see the results got at each time without re-entering the values again. The software is limited to the following algebraic equations:

1. Simultaneous Equations
2. Quadratic Equations
3. Cubic Equations

The programming languages used in the construction of the project include:

1. Java Programming Language
2. Cascading Style Sheets (CSS) for styling
3. MathTex (For Mathematical Formulas)
4. Structured Query Language (for Database)

The software design methodology used in the design of this project is the Object-Oriented Analysis and Design Methodology which uses Analysis, Design (System and Object), and Implementation as the steps in designing Object Oriented Software.

**5.2 Recommendation**

This software is recommended for secondary school students as well as mathematics teachers. It improves the performance of students in mathematics course involving algebraic equations because it helps the user understand the concept of algebraic equations by exposing one to as many problems as possible, all of which are engineered by the user.

**References**

**[1]** G. James, C. James, “Mathematics Dictionary”, Van Nonstrand, 1968, 3 ed. p131.

**[2]** H. Michiel, “Algebraic Equation”, Encyclopedia of Mathematics, Springer, 2001,

ed.

**[3]** enotes.com (2018). “What is a Linear Equation”, [Online], Available:

http://www.enotes.com/homework-help/what-linear-equation-471761

**[4]** brilliant.org (2018). “Cubic Equations”, 2. [Online], Available:

https://brilliant.org/wiki/cubic-equations/

**[5]** G. James, J. Bill, S. Guy, B. Gilad, B. Alex. The Java Language Specification,

2014 Java SE 8 ed. PDF.

**[6]** D, Eck (2006] “Introduction to Java Programming using Java”, 2006, 7ed. Section

1.3.

**[7]** B2C (2017). “Top 20 Programming Lanugages in 2017”, [Online], Available:

https://www.businesss2community.com/tech-gadgets/top-20-popular-programming-languages-2017-01791470

**[8]** Oracle (2018), “Getting Started With JavaFX”, [Online], Available:

https://docs.oracle.com/javafx/2/get\_started/css.htm.

**[9]** Techopedia (2018), “Structured Query Language”, [Online], Available:

https://www.techopedia.com/definition/1245/structured-query-language-sql.

**[10]** W3Schools (2018), “SQL Syntax”, [Online], Available:

https://www.w3schools.com/sql/sql\_syntax.asp.

**[11]** D. Adu, “Comprehensive Mathematics for senior secondary schools”, A. Johnson

Publishers Ltd, 2004, p67 – 70.

**[12]** Mathematics Association of Nigeria, “MAN Mathematics for Senior Secondary

Schools 3”, University Press Publishers PLC, 2012, 3 ed, p160 – 171.

**[13]** W. Allyn, “Basic Technical Mathematics with Calculus, Seventh Edition”.

Addison Wesley Longman Inc, 2000.

**[14]** R. Barnett, S. Philip, “Schaum’s Outline of Theory and Problems of

Elementary Algebra”. The McGraw-Hill Companies, 2004, Chapter 13 p. 291.

**[15]** K. Victor “A history of Mathematics”, Boston: Addison Wesley, 2004, p.220.

**[16]** R. Nickalls, “Viete, Descartes and the cubic equation”. *Mathematical Gazette*,

vol 90, pp. 203 – 208, July 2006.

**[17]** Techopedia (2018), “Object Oriented Model”. [Online], Available:

https://www.techopedia.com/definition/28584/object-oriented-modeling-oom.

**[18]** Tutorialspoint (2018), “Object Oriented System”, [Online], Available:

https://www.tutorialspoint.come/object\_oriented\_analysis\_design/ooad\_object\_` oriented\_system.htm

**APPENDIX A: Source Code for Homepage (Login page and Index Page)**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import javafx.application.Application;**

**import javafx.geometry.Insets;**

**import javafx.scene.Group;**

**import javafx.scene.Scene;**

**import javafx.scene.control.Button;**

**import javafx.scene.control.Label;**

**import javafx.scene.control.TextField;**

**import javafx.scene.layout.VBox;**

**import javafx.stage.Stage;**

**public class Login extends Application {**

**public void start(final Stage stg) {**

**Group gp = new Group();**

**Scene sc = new Scene(gp, 400, 300);**

**stg.setTitle("Pearl Algebraic Math Equator");**

**Label log = new Label("Login first to begin");**

**log.setPrefWidth(350);**

**log.setStyle("-fx-text-fill: black;"**

**+ "-fx-font-family: Times New Roman;"**

**+ "-fx-font-size: 16px;"**

**+ "-fx-alignment: center;"**

**+ "-fx-font-weight: bold;");**

**VBox gd = new VBox();**

**gd.setStyle("-fx-background-color: beige;"**

**+ "-fx-border-radius: 5px;");**

**gd.setPrefWidth(400);**

**gd.setPrefHeight(400);**

**gd.setLayoutX(0);**

**gd.setLayoutY(0);**

**gd.setSpacing(12);**

**gd.setPadding(new Insets(20,20,20,20));**

**TextField tf1 = new TextField();**

**tf1.setPromptText("Username Goes Here");**

**tf1.setPrefWidth(100);**

**tf1.setPrefHeight(35);**

**tf1.setStyle("-fx-alignment: center;"**

**+ "-fx-font-family: Calisto;"**

**+ "-fx-font-size: 15px;");**

**TextField tf2 = new TextField();**

**tf2.setPromptText("Password Goes Here");**

**tf2.setPrefWidth(100);**

**tf2.setPrefHeight(35);**

**tf2.setStyle("-fx-alignment: center;"**

**+ "-fx-font-family: Calisto;"**

**+ "-fx-font-size: 15px;");**

**Button logs = new Button("Log me in");**

**logs.setPrefHeight(35);**

**logs.setPrefWidth(360);**

**logs.setStyle("-fx-background-color: lightgray;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-text-align: center;"**

**+ "-fx-font-family: Times;"**

**+ "-fx-font-weight: bold;"**

**+ "-fx-font-size: 15px;");**

**Button logged = new Button("I need to register");**

**logged.setPrefWidth(360);**

**logged.setPrefHeight(35);**

**logged.setStyle("-fx-background-color: lightgray;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-text-align: center;"**

**+ "-fx-font-family: Times;"**

**+ "-fx-font-weight: bold;"**

**+ "-fx-font-size: 15px;");**

**gd.getChildren().addAll(log, tf1, tf2, logs, logged);**

**gp.getChildren().addAll(gd);**

**stg.setScene(sc);**

**stg.show();**

**}**

**public static void main(String[] args) throws Exception {**

**getConnection();**

**Application.launch(args);**

**}**

**public static Connection getConnection() throws Exception {**

**try {**

**String driver = "com.mysql.cj.jdbc.Driver";**

**String url = "jdbc:mysql://localhost:3306/JavaMath";**

**String username = "root";**

**String password = "";**

**Class.forName(driver);**

**Connection conn = DriverManager.getConnection(url, username, password);**

**System.out.println("Connected");**

**return conn;**

**}**

**catch (Exception e){System.out.println(e);};**

**return null;**

**}**

**}**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import javafx.application.Application;**

**import javafx.event.ActionEvent;**

**import javafx.event.EventHandler;**

**import javafx.geometry.Insets;**

**import javafx.scene.Scene;**

**import javafx.scene.control.Button;**

**import javafx.scene.control.Label;**

**import javafx.scene.image.Image;**

**import javafx.scene.layout.GridPane;**

**import javafx.scene.text.Font;**

**import javafx.stage.Stage;**

**public class Indexes {**

**private Image ico;**

**public Indexes() {**

**ico = new Image(Indexes.class.getResource("img/Advert.png").toExternalForm());**

**}**

**public void start(final Stage stg) {**

**GridPane gp = new GridPane();**

**gp.setPadding(new Insets(10, 10, 10, 10));**

**gp.setVgap(1.5);**

**gp.setHgap(3);**

**String img = Indexes.class.getResource("b2.png").toExternalForm();**

**Scene sc = new Scene(gp, 450, 500);**

**gp.setStyle("-fx-background-image: url("+ img +"); "**

**+ "-fx-background-position: center center; "**

**+ "-fx-background-repeat: stretch; "**

**+ "-fx-background-size: 980 550");**

**stg.setScene(sc);**

**stg.setResizable(false);**

**stg.getIcons().add(ico);**

**stg.setTitle("Pearl's Algebraic Equator");**

**Font ft = new Font("Elephant", 20);**

**Font fit = new Font("Times New Roman", 25);**

**Label lbl = new Label ("Pearl's Algebraic Equation solver");**

**GridPane.setConstraints(lbl, 10, 10 );**

**lbl.setFont(fit);**

**lbl.setStyle("-fx-text-fill: white;"**

**+ "-fx-font-weight: bold;");**

**lbl.setPrefSize(400, 20);**

**Button a = new Button("Simultaneous Equations");**

**GridPane.setConstraints(a, 10, 40);**

**a.setFont(ft);**

**a.setStyle("-fx-text-fill: black;");**

**a.setOnAction(new EventHandler<ActionEvent>() {**

**public void handle (ActionEvent e) {**

**Simultaneous sm = new Simultaneous();**

**sm.start(stg);**

**}**

**});**

**Button b = new Button("Quadratic Equations");**

**GridPane.setConstraints(b, 10, 50);**

**b.setFont(ft);**

**b.setStyle("-fx-text-fill: black;");**

**b.setOnAction(new EventHandler<ActionEvent>(){**

**public void handle(ActionEvent ed) {**

**Quadratic quad = new Quadratic();**

**quad.start(stg);**

**}**

**});**

**Button bub = new Button("Cubic Equations");**

**GridPane.setConstraints(bub, 10, 60);**

**bub.setFont(ft);**

**bub.setStyle("-fx-text-fill: black;");**

**bub.setOnAction(new EventHandler<ActionEvent>(){**

**public void handle(ActionEvent ed) {**

**Cubic quad = new Cubic();**

**quad.start(stg);**

**}**

**});**

**Button p = new Button("Special Case Equations");**

**GridPane.setConstraints(p, 10, 70);**

**p.setFont(ft);**

**p.setStyle("-fx-text-fill: black;");**

**p.setOnAction(new EventHandler<ActionEvent>() {**

**public void handle(ActionEvent hg) {**

**Special sp = new Special();**

**sp.start(stg);**

**}**

**});**

**Button q = new Button("Database Search");**

**GridPane.setConstraints(q, 10, 80);**

**q.setFont(ft);**

**q.setStyle("-fx-text-fill: black;");**

**q.setOnAction(new EventHandler<ActionEvent>() {**

**public void handle(ActionEvent hg) {**

**Javadb sp = new Javadb();**

**sp.start(stg);**

**}**

**});**

**Button sct = new Button("Tutorials");**

**GridPane.setConstraints(sct, 10, 80);**

**sct.setFont(ft);**

**sct.setStyle("-fx-text-fill: black;");**

**sct.setOnAction(new EventHandler<ActionEvent>() {**

**public void handle(ActionEvent ae) {**

**Tutorial tl = new Tutorial();**

**tl.start(stg);**

**}**

**});**

**gp.getChildren().addAll(lbl, a, b, bub, p, q, sct);**

**stg.show();**

**}**

**}**

**APPENDIX B: Source code listing for Simultaneous Equations**

**import java.awt.Dimension;**

**import java.awt.FlowLayout;**

**import java.awt.GridBagConstraints;**

**import java.awt.image.BufferedImage;**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import java.sql.PreparedStatement;**

**import java.text.DecimalFormat;**

**import javax.swing.BorderFactory;**

**import javax.swing.JLabel;**

**import javax.swing.JPanel;**

**import javax.swing.JTextPane;**

**import org.scilab.forge.jlatexmath.TeXConstants;**

**import org.scilab.forge.jlatexmath.TeXFormula;**

**import org.scilab.forge.jlatexmath.TeXIcon;**

**import javafx.collections.FXCollections;**

**import javafx.collections.ObservableList;**

**import javafx.embed.swing.SwingNode;**

**import javafx.event.ActionEvent;**

**import javafx.event.EventHandler;**

**import javafx.geometry.Insets;**

**import javafx.scene.Group;**

**import javafx.scene.Scene;**

**import javafx.scene.SubScene;**

**import javafx.scene.chart.CategoryAxis;**

**import javafx.scene.chart.LineChart;**

**import javafx.scene.chart.NumberAxis;**

**import javafx.scene.chart.XYChart;**

**import javafx.scene.chart.XYChart.Series;**

**import javafx.scene.control.Button;**

**import javafx.scene.control.Label;**

**import javafx.scene.control.Menu;**

**import javafx.scene.control.MenuBar;**

**import javafx.scene.control.ScrollPane;**

**import javafx.scene.control.TextField;**

**import javafx.scene.input.MouseEvent;**

**import javafx.scene.layout.HBox;**

**import javafx.scene.layout.StackPane;**

**import javafx.scene.layout.VBox;**

**import javafx.scene.paint.Color;**

**import javafx.scene.text.Font;**

**import javafx.scene.text.FontWeight;**

**import javafx.stage.\*;**

**public class Simultaneous {**

**public static Connection getConnection() throws Exception {**

**try {**

**String driver = "com.mysql.cj.jdbc.Driver";**

**String url = "jdbc:mysql://localhost:3306/JavaMath";**

**String username = "root";**

**String password = "";**

**Class.forName(driver);**

**Connection conn = DriverManager.getConnection(url, username, password);**

**System.out.println("Connected");**

**return conn;**

**}**

**catch (Exception e){System.out.println(e);};**

**return null;**

**}**

**public void start (final Stage stg) {**

**stg.setTitle("Simultaneous Equations");**

**stg.setResizable(false);**

**final Group gp = new Group();**

**Scene sc = new Scene(gp, 960, 700, Color.WHITE);**

**sc.getStylesheets().addAll(Simultaneous.class.getResource("Style.css").toExternalForm());**

**MenuBar mb = new MenuBar();**

**mb.setLayoutY(0);**

**mb.setLayoutX(0);**

**mb.setPrefWidth(960);**

**mb.setPrefHeight(30);**

**mb.setStyle("-fx-background-color: lightgray;"**

**+ "-fx-background-opacity: 1;");**

**Menu mbt = new Menu("Export");**

**mbt.setStyle("-fx-font-family: Times;"**

**+ "-fx-font-size: 15;"**

**+ "-fx-font-weight: BOLD;");**

**Menu mbo = new Menu("Refresh");**

**mbo.setStyle("-fx-font-family: Times;"**

**+ "-fx-font-size: 15;"**

**+ "-fx-font-weight: BOLD;");**

**Menu mbb = new Menu("Night View");**

**mbb.setStyle("-fx-font-family: Times;"**

**+ "-fx-font-size: 15;"**

**+ "-fx-font-weight: BOLD;");**

**Menu mbc = new Menu("Basic Tutorial");**

**mbc.setStyle("-fx-font-family: Times;"**

**+ "-fx-font-size: 15;"**

**+ "-fx-font-weight: BOLD;");**

**mb.getMenus().addAll(mbt, mbo, mbb, mbc);**

**Label info = new Label("Fill the form below and submit");**

**info.setFont(Font.font("Trebuchet MS", FontWeight.BOLD, 25));**

**info.setLayoutX(450);**

**info.setLayoutY(50);**

**Label a\_in = new Label ("Value of x1: ");**

**a\_in.setFont(Font.font("Times New Roman", FontWeight.BOLD, 20));**

**a\_in.setLayoutX(330);**

**a\_in.setLayoutY(120);**

**final TextField a = new TextField();**

**a.setFont(Font.font("Cambria", FontWeight.MEDIUM, 15));**

**a.setLayoutX(480);**

**a.setLayoutY(120);**

**a.setPrefWidth(120);**

**Label a1\_in = new Label ("Value of x2: ");**

**a1\_in.setFont(Font.font("Times New Roman", FontWeight.BOLD, 20));**

**a1\_in.setLayoutX(630);**

**a1\_in.setLayoutY(120);**

**final TextField a1 = new TextField();**

**a1.setFont(Font.font("Cambria", FontWeight.MEDIUM, 15));**

**a1.setLayoutX(780);**

**a1.setLayoutY(120);**

**a1.setPrefWidth(120);**

**Label b\_in = new Label ("Value of y1: ");**

**b\_in.setFont(Font.font("Times New Roman", FontWeight.BOLD, 20));**

**b\_in.setLayoutX(330);**

**b\_in.setLayoutY(170);**

**final TextField b = new TextField();**

**b.setFont(Font.font("Cambria", FontWeight.MEDIUM, 15));**

**b.setLayoutX(480);**

**b.setLayoutY(170);**

**b.setPrefWidth(120);**

**Label b1\_in = new Label ("Value of y2: ");**

**b1\_in.setFont(Font.font("Times New Roman", FontWeight.BOLD, 20));**

**b1\_in.setLayoutX(630);**

**b1\_in.setLayoutY(170);**

**final TextField b1 = new TextField();**

**b1.setFont(Font.font("Cambria", FontWeight.MEDIUM, 15));**

**b1.setLayoutX(780);**

**b1.setLayoutY(170);**

**b1.setPrefWidth(120);**

**Label c\_in = new Label ("Value of c1: ");**

**c\_in.setFont(Font.font("Times New Roman", FontWeight.BOLD, 20));**

**c\_in.setLayoutX(330);**

**c\_in.setLayoutY(220);**

**final TextField c = new TextField();**

**c.setFont(Font.font("Cambria", FontWeight.MEDIUM, 15));**

**c.setLayoutX(480);**

**c.setLayoutY(220);**

**c.setPrefWidth(120);**

**Label c1\_in = new Label ("Value of c2: ");**

**c1\_in.setFont(Font.font("Times New Roman", FontWeight.BOLD, 20));**

**c1\_in.setLayoutX(630);**

**c1\_in.setLayoutY(220);**

**final TextField c1 = new TextField();**

**c1.setFont(Font.font("Cambria", FontWeight.MEDIUM, 15));**

**c1.setLayoutX(780);**

**c1.setLayoutY(220);**

**c1.setPrefWidth(120);**

**VBox vb = new VBox();**

**vb.setSpacing(5);**

**vb.setLayoutX(-10);**

**vb.setLayoutY(25);**

**vb.setPrefHeight(900);**

**vb.setPrefWidth(250);**

**vb.setPadding(new Insets(80, 0, 0, 0));**

**String img = Indexes.class.getResource("b2.png").toExternalForm();**

**vb.setStyle("-fx-background-image: url( ' " + img + " '); "**

**+ "-fx-background-position: center center; "**

**+ "-fx-background-repeat: stretch; "**

**+ "-fx-background-size: 250 900;"**

**+ "-fx-border-width: 5px;"**

**+ "-fx-border-color: darkgray;");**

**final Button sub = new Button ("Solve by Substitution");**

**sub.setLayoutX(150);**

**sub.setLayoutY(280);**

**sub.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**sub.setPrefWidth(250);**

**sub.setPrefHeight(60);**

**sub.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0;");**

**sub.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**sub.setStyle("-fx-background-color: white;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**sub.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**sub.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**final Button elim = new Button ("Solve by Elimination");**

**elim.setLayoutX(350);**

**elim.setLayoutY(280);**

**elim.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**elim.setPrefWidth(250);**

**elim.setPrefHeight(60);**

**elim.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**elim.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**elim.setStyle("-fx-background-color: white;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**elim.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**elim.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**final Button graph = new Button ("Solve by Graph");**

**graph.setLayoutX(550);**

**graph.setLayoutY(280);**

**graph.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**graph.setPrefWidth(250);**

**graph.setPrefHeight(60);**

**graph.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**graph.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**graph.setStyle("-fx-background-color: white;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**graph.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**graph.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**final Button gap = new Button ("Return to Home");**

**gap.setLayoutX(550);**

**gap.setLayoutY(280);**

**gap.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**gap.setPrefWidth(250);**

**gap.setPrefHeight(60);**

**gap.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**gap.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**gap.setStyle("-fx-background-color: white;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**gap.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**gap.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0;");**

**}**

**});**

**final Button gay = new Button ("Basic Tutorial");**

**gay.setLayoutX(550);**

**gay.setLayoutY(280);**

**gay.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**gay.setPrefWidth(250);**

**gay.setPrefHeight(60);**

**gay.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**gay.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**gay.setStyle("-fx-background-color: darkgreen;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**gay.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**gay.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**gay.setOnAction(new EventHandler<ActionEvent>(){**

**public void handle(ActionEvent ae) {**

**Basic bs = new Basic();**

**bs.start(stg);**

**}**

**});**

**final Button gee = new Button ("Settings");**

**gee.setLayoutX(550);**

**gee.setLayoutY(280);**

**gee.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**gee.setPrefWidth(250);**

**gee.setPrefHeight(60);**

**gee.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**gee.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**gee.setStyle("-fx-background-color: darkgreen;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**gee.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**gee.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**gap.setOnAction(new EventHandler<ActionEvent>(){**

**public void handle(ActionEvent av) {**

**Indexes ind = new Indexes();**

**ind.start(stg);**

**}**

**});**

**vb.getChildren().addAll(sub, elim, graph, gap, gay, gee);**

**sub.setOnAction(new EventHandler<ActionEvent>() {**

**@Override**

**public void handle(ActionEvent ex) {**

**// TODO Auto-generated method stub**

**double i = Double.parseDouble(a.getText());**

**double j = Double.parseDouble(a1.getText());**

**double k = Double.parseDouble(b.getText());**

**double l = Double.parseDouble(b1.getText());**

**double m = Double.parseDouble(c.getText());**

**double n = Double.parseDouble(c1.getText());**

**double o = m/i;**

**double p = j \* o;**

**double od = k/i;**

**double q = j \* od;**

**double s = l - q;**

**double t = n - p;**

**double u = t / s;**

**double v = k \* u;**

**double w = m - v;**

**double z = w / i;**

**DecimalFormat df = new DecimalFormat("#.###");**

**String s1 = "\\ The \\ Equations \\ are: " ;**

**String s2 = "\\ {" + "} \\";**

**String s3 = "\\ { " + i + "x + " + k + "y = " + m + "} \\";**

**String s4 = "\\ { " + j + "x + " + l + "y = " + n + "} \\";**

**String s5 = "\\ {" + "} \\";**

**String s55 = "\\ Using \\ Substitution, ";**

**String s6 = "\\ Make \\ x \\ the \\ subject \\ of \\ the \\ formula \\ in \\ equation(1) " ;**

**String s7 = "\\ {" + "} \\";**

**String s8 = "\\ And \\ we \\ have: " ;**

**String s9 = "\\ " + "x = " + "\\dfrac { " + m + " - " + k + "y " + "} {" + i + "}" + " \\" ;**

**String s10 = "\\ {" + "} \\";**

**String s11 = "\\ Simplify \\ the \\ expression \\ to \\ get: " + " \\";**

**String s12 = "\\ { " + "x = " + o + " - " + od + "y --- " + "\\ Equation \\ (3)" + "} \\";**

**String s13 = "\\ {" + "} \\";**

**String s14 = "\\ Insert \\ equation \\ (3) \\ into \\ equation \\ (2) " ;**

**String s15 = "\\ {" + "} \\";**

**String s16 = "\\ { " + j + " \* " + " ( " + o + " - " + od + "y ) + " + l + "y = " + n + "} \\";**

**String s17 = "\\ {" + "} \\";**

**String s18 = "\\ { " + p + " - " + q + "y + " + l + "y = " + n + "} \\";**

**String s19 = "\\ Collect \\ Like \\ terms \\ to \\ get: " + " \\";**

**String s20 = "\\ {" + "} \\";**

**String s21 = "\\ { " + " - " + q + "y + " + l + "y = " + n + " - " + p + "} \\";**

**String s215 = "\\ {" + "} \\";**

**String s22 = "\\ { " + s + "y = " + t + "} \\";**

**String s23 = "\\ {" + "} \\";**

**String s24 = "\\ Divide \\ both \\ sides \\ by \\ " + s + "\\ to \\ get \\ y: " + " \\";**

**String s245 = "\\ {" + "} \\";**

**String s25 = "\\ y \\ = " + "\\dfrac { " + t + "} {" + s + "}" + " \\";**

**String s255 = "\\ {" + "} \\";**

**String s26 = "\\ y \\ = " + u + " \\";**

**String s27 = "\\ {" + "} \\";**

**String s28 = "\\ Substitute \\ the \\ value \\ of \\ y \\ into \\ equation \\ (1) " + " \\";**

**String s285 = "\\ {" + "} \\";**

**String s29 = "\\ { " + i + "x + " + k + " \* " + u + "= " + m + "} \\";**

**String s295 = "\\ {" + "} \\";**

**String s30 = "\\ { " + i + "x + " + v + " = " + m + "} \\";**

**String s31 = "\\ {" + "} \\";**

**String s32 = "\\ Collect \\ like \\ terms \\ to \\ get: " + " \\";**

**String s325 = "\\ {" + "} \\";**

**String s33 = "\\ { " + i + "x = " + m + " - " + v + "} \\";**

**String s335 = "\\ {" + "} \\";**

**String s34 = "\\ { "+ i + "x = " + w + "} \\";**

**String s345 = "\\ {" + "} \\";**

**String s35 = "\\ Divide \\ both \\ sides \\ by \\ the \\ co-efficient \\ of \\ x \\ to \\ get: " + " \\";**

**String s355 = "\\ {" + "} \\";**

**String s36 = "\\ x = " + "\\dfrac { " + w + "} {" + i + "}" + " \\";**

**String s365 = "\\ {" + "} \\";**

**String s37 = "\\ and \\ x = " + z + " \\";**

**String s375 = "\\ {" + "} \\";**

**String s38 = "\\ Thus, \\ the \\ value \\ of \\ x \\ is \\ = " + df.format(z) + " \\";**

**String s385 = "\\ {" + "} \\";**

**String s39 = "\\ and \\ y \\ is \\ = " + df.format(u) + " \\";**

**TeXFormula ef = new TeXFormula(s1 + s2 + s3 + s2 + s4 + s5 + s55 + s5 + s6 + s7 + s8 + s7 + s9**

**+ s10 + s11 + s10 + s12 + s13 + s14 + s15 + s16 + s17 + s18 + s17 + s19**

**+ s20 + s21 + s215 + s22 + s23 + s24 + s245 + s25 + s255 + s26**

**+ s27 + s28 + s285 + s29 + s295 + s30 + s31 + s32 + s325**

**+ s33 + s335 + s34 + s345 + s35 + s355 + s36 + s365 + s37**

**+ s375 + s38 + s385 + s39);**

**TeXIcon ti = ef.createTeXIcon(TeXConstants.DELIM\_SQUARE\_BRACKET, 18);**

**BufferedImage bi = new BufferedImage(ti.getIconWidth(), ti.getIconHeight(), BufferedImage.TYPE\_4BYTE\_ABGR);**

**ti.setInsets(new java.awt.Insets (1, 4, 0, 0));**

**ti.paintIcon(new JLabel(), bi.getGraphics(), 0, 0);**

**ti.setIconWidth(300, TeXConstants.ALIGN\_LEFT);**

**ti.setIconHeight(500, TeXConstants.ALIGN\_LEFT);**

**ti.setForeground(java.awt.Color.BLACK);**

**// Employ Swing for operations in FX---- the node that carries it all!**

**SwingNode sn = new SwingNode();**

**// Swing TextPane; for storing the Strings**

**JTextPane ht = new JTextPane();**

**ht.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 0, java.awt.Color.WHITE));**

**ht.insertIcon(ti);**

**ht.setPreferredSize(new Dimension(550, 920));**

**ht.setForeground(java.awt.Color.BLUE);**

**ht.setBackground(java.awt.Color.WHITE);**

**ht.setVisible(true);**

**ht.setMargin(new java.awt.Insets(0,0,0,0));**

**ht.setFocusable(false);**

**ht.setEditable(false);**

**// JPanel to hold the TextPane**

**JPanel jp = new JPanel();**

**jp.setLayout(new FlowLayout());**

**GridBagConstraints hi = new GridBagConstraints();**

**hi.fill = GridBagConstraints.FIRST\_LINE\_START;**

**jp.add(ht, hi);**

**jp.setBackground(java.awt.Color.WHITE);**

**// Add all SWING components to the SwingNode**

**sn.setContent(jp);**

**sn.setLayoutX(50);**

**sn.setLayoutY(300);**

**// Add the SwingNode to the ScrollPane**

**ScrollPane scp = new ScrollPane(sn);**

**scp.setContent(sn);**

**scp.setFitToWidth(true);**

**scp.setPrefHeight(350);**

**scp.setPrefWidth(570);**

**scp.setLayoutX(350);**

**scp.setLayoutY(320);**

**scp.setVisible(true);**

**// Add the ScrollPane to the Scene**

**gp.getChildren().add(scp);**

**try {**

**Connection con = getConnection();**

**PreparedStatement posted = con.prepareStatement("INSERT INTO simultaneous (var\_x1, var\_x2, var\_y1, var\_y2, c\_1, c\_2, sol\_x, sol\_y, method\_used) "**

**+ "VALUES ('"+i+"', '"+j+"', + '"+k+"', + '"+l+"', '"+m+"', '"+n+"', '"+u+"', '"+z+"', 'Substitition')");**

**posted.executeUpdate();**

**System.out.println("Successful");**

**}**

**catch (Exception e){System.out.println(e);}**

**finally{**

**System.out.println("Successful");**

**}**

**}**

**});**

**elim.setOnAction(new EventHandler <ActionEvent>(){**

**public void handle (ActionEvent eh) {**

**double i = Double.parseDouble(a.getText());**

**double j = Double.parseDouble(a1.getText());**

**double k = Double.parseDouble(b.getText());**

**double l = Double.parseDouble(b1.getText());**

**double m = Double.parseDouble(c.getText());**

**double n = Double.parseDouble(c1.getText());**

**double o = i \* j;**

**double p = k \* j;**

**double q = m \* j;**

**double r = l \* i;**

**double s = n \* i;**

**double t = p - r;**

**double u = q - s;**

**double v = u / t;**

**double w = k \* v;**

**double z = m - w;**

**double oz = z / i;**

**// Format for displaying decimal numbers**

**DecimalFormat fm = new DecimalFormat("#.###");**

**// The step-by-step solutions broken down**

**String first = "\\ The \\ Equations \\ are: " + " \\" ;**

**String es = "\\ { " + " } \\";**

**String sec = "\\ {" + i + "x + " + k + "y " + " = " + m + "} \\";**

**String third = "\\ { " + j + "x + " + l + "y " + " = " + n + "} \\";**

**String fourth = "\\ " + "Eliminate \\ x \\ to \\ get \\ y " + " \\";**

**String fifth = "\\ " + "And \\ we \\ have: " + " \\" + " \\";**

**String five = "\\ " + "Multiply \\ equation(1) \\ by \\ the \\ co-efficient \\ of \\";**

**String six = "\\ " + "x \\ in \\ equation(2) \\ and \\ vice \\ versa \\";**

**String sixth = "\\ { " + j + " \* " + "( " + i + "x + " + k + "y = " + m + ")" + "} \\";**

**String seven = "\\ { " + i + " \* " + "( " + j + "x + " + l + "y = " + n + ")" + "} \\";**

**String eight = "\\ " + "Open \\ the \\ brackets \\ to \\ get " + " \\";**

**String nine = "\\ { " + o + "x + " + p + "y = " + q + "-------- (I)" + "} \\";**

**String ten = "\\ { " + o + "x + " + r + "y = " + s + "---------- (II)" + "} \\";**

**String eleven = "\\ " + "Subtract \\ equation \\ (II) \\ from \\ equation \\ (I) \\ to \\ get: " + " \\";**

**String twelve = "\\ { " + "0 + " + t + "y = " + u + "} \\";**

**String teen = "\\ " + "Divide \\ both \\ sides \\ by \\ the \\ co-efficient \\ of \\ y \\ to \\ get: " +" \\";**

**String teen1 = "\\ " + "\\dfrac { " + t + "} {" + t + "}" + " = ";**

**String teen2 = "\\ " + "\\dfrac { " + u + "} {" + t + "}" + " \\";**

**String teen3 = "\\ " + "Therefore, \\ y \\ = " + v + " \\";**

**String teen4 = "\\ " + "Substitute \\ the \\ value \\ of \\ y \\ into \\ equation(I) \\ to \\ get: " + " \\";**

**String teen5 = "\\ { " + i + "x + " + k + " \* " + v + " = " + m + "} \\";**

**String teen6 = "\\ { " + i + "x + " + w + " = " + m + "} \\";**

**String teen7 = "\\ " + "Subtract \\ " + w + "\\ from \\ both \\ sides \\ to get: " + " \\";**

**String teen8 = "\\ { " + i + "x = " + m + " - " + w + "} \\";**

**String teen9 = "\\ { " + i + "x = " + z +"} \\";**

**String teen10 = "\\ " + "Divide \\ both \\ sides \\ by \\ the \\ co-efficient \\ of \\ x \\ to \\ get: " + " \\";**

**String teen11 = "\\ " + "\\dfrac { " + i + "} {" + i + "}" + " = ";**

**String teen12 = "\\ " + "\\dfrac { " + z + "} {" + i + "}" + " \\";**

**String teen13 = "\\ " + "And \\ x = " + oz + " \\";**

**String teen14 = "\\ " + "Thus, \\ x = " + fm.format(oz) + "\\ and \\ y = " + fm.format(v);**

**// Convert all Strings to one TeXFormula String**

**TeXFormula ef = new TeXFormula(first + es +sec + es + third + es + fourth + fifth + es + five + es + six + es + sixth + es + seven + es + eight +**

**es + nine + es + ten + es + eleven + es + twelve + es + teen + es + teen1 + teen2 + es + teen3 + es + teen4 + es + teen5 + es + teen6 + es + teen7 +**

**es + teen8 + es + teen9 + es + teen10 + es + teen11 + teen12 + es + teen13 + es + teen14);**

**// Convert String to Icon**

**TeXIcon ti = ef.createTeXIcon(TeXConstants.DELIM\_SQUARE\_BRACKET, 18);**

**BufferedImage b = new BufferedImage(ti.getIconWidth(), ti.getIconHeight(), BufferedImage.TYPE\_4BYTE\_ABGR);**

**ti.setInsets(new java.awt.Insets (1, 4, 0, 0));**

**ti.paintIcon(new JLabel(), b.getGraphics(), 0, 0);**

**ti.setIconWidth(300, TeXConstants.ALIGN\_LEFT);**

**ti.setForeground(java.awt.Color.BLACK);**

**// Import SwingNode to hold Swing Code**

**SwingNode sn = new SwingNode();**

**// Swing TextPane**

**JTextPane ht = new JTextPane();**

**ht.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 0, java.awt.Color.WHITE));**

**ht.insertIcon(ti);**

**ht.setPreferredSize(new Dimension(550, 890));**

**ht.setForeground(java.awt.Color.BLUE);**

**ht.setBackground(java.awt.Color.WHITE);**

**ht.setVisible(true);**

**ht.setMargin(new java.awt.Insets(0,0,0,0));**

**ht.setFocusable(false);**

**ht.setEditable(false);**

**// JPanel --- Swing Component**

**JPanel jp = new JPanel();**

**jp.setLayout(new FlowLayout());**

**GridBagConstraints hi = new GridBagConstraints();**

**hi.fill = GridBagConstraints.FIRST\_LINE\_START;**

**jp.add(ht, hi);**

**jp.setBackground(java.awt.Color.WHITE);**

**// SwingNode to hold JPanel**

**sn.setContent(jp);**

**sn.setLayoutX(50);**

**sn.setLayoutY(300);**

**// ScrollPane to Hold SwingNode**

**ScrollPane scp = new ScrollPane(sn);**

**scp.setContent(sn);**

**scp.setFitToWidth(true);**

**scp.setPrefHeight(350);**

**scp.setPrefWidth(570);**

**scp.setLayoutX(350);**

**scp.setLayoutY(320);**

**scp.setVisible(true);**

**// Add all to Layout**

**gp.getChildren().add(scp);**

**try {**

**Connection con = getConnection();**

**PreparedStatement posted = con.prepareStatement("INSERT INTO simultaneous (var\_x1, var\_x2, var\_y1, var\_y2, c\_1, c\_2, sol\_x, sol\_y, method\_used) "**

**+ "VALUES ('"+i+"', '"+j+"', + '"+k+"', + '"+l+"', '"+m+"', '"+n+"', '"+oz+"', '"+v+"', 'Elimination')");**

**posted.executeUpdate();**

**System.out.println("Successful");**

**}**

**catch (Exception e){System.out.println(e);}**

**finally{**

**System.out.println("Successful");**

**}**

**}**

**});**

**graph.setOnAction(new EventHandler<ActionEvent>(){**

**@SuppressWarnings({ "unchecked", "rawtypes" })**

**public ObservableList<XYChart.Series<String, Double>> chat() {**

**double ga = Double.parseDouble(a.getText());**

**double gb = Double.parseDouble(a1.getText());**

**double gc = Double.parseDouble(b.getText());**

**double gd = Double.parseDouble(b1.getText());**

**double ge = Double.parseDouble(c.getText());**

**double gf = Double.parseDouble(c1.getText());**

**ObservableList<XYChart.Series<String, Double>> chart = FXCollections.observableArrayList();**

**Series<String, Double> ace = new Series<String, Double>();**

**Series<String, Double> brace = new Series<String, Double>();**

**ace.setName("Equation\_1");**

**brace.setName("Equation\_2");**

**for (double i = -10; i <= 10; i++) {**

**double aceValue = (ge - (ga \* i)) / gc;**

**double braceValue = (gf - (gb \* i)) / gd;**

**ace.getData().add(new XYChart.Data(Double.toString(i), aceValue));**

**brace.getData().add(new XYChart.Data(Double.toString(i), braceValue));**

**}**

**chart.addAll(ace, brace);**

**return chart;**

**}**

**@SuppressWarnings("unchecked")**

**public void handle(ActionEvent gra) {**

**CategoryAxis xAxis = new CategoryAxis();**

**NumberAxis yAxis = new NumberAxis();**

**@SuppressWarnings({ "rawtypes" })**

**LineChart lineChart = new LineChart(xAxis, yAxis);**

**lineChart.setData(chat());**

**lineChart.setTitle("Graphical Solutions");**

**lineChart.setStyle("-fx-font-family: ALGERIAN");**

**lineChart.setStyle("-fx-background-color: white;"**

**+ "-fx-text-fill: blue;");**

**lineChart.setAnimated(true);**

**StackPane root = new StackPane();**

**root.getChildren().add(lineChart);**

**SubScene scf = new SubScene(root, 570, 350);**

**scf.setLayoutX(350);**

**scf.setLayoutY(320);**

**gp.getChildren().add(scf);**

**}**

**});**

**gp.getChildren().addAll(info, vb, a\_in, a, a1\_in, a1, b\_in, b, b1\_in, b1, c\_in, c, c1\_in, c1, mb);**

**stg.setScene(sc);**

**stg.show();**

**}**

**}**

**APPENDIX C: Source Code Listing for Quadratic Equations**

**import java.awt.Dimension;**

**import java.awt.FlowLayout;**

**import java.awt.GridBagConstraints;**

**import java.awt.image.BufferedImage;**

**import java.text.DecimalFormat;**

**import javax.swing.BorderFactory;**

**import javax.swing.JLabel;**

**import javax.swing.JPanel;**

**import javax.swing.JTextPane;**

**import org.scilab.forge.jlatexmath.TeXConstants;**

**import org.scilab.forge.jlatexmath.TeXFormula;**

**import org.scilab.forge.jlatexmath.TeXIcon;**

**import javafx.collections.FXCollections;**

**import javafx.collections.ObservableList;**

**import javafx.embed.swing.SwingNode;**

**import javafx.event.ActionEvent;**

**import javafx.event.EventHandler;**

**import javafx.geometry.Insets;**

**import javafx.scene.Group;**

**import javafx.scene.Scene;**

**import javafx.scene.SubScene;**

**import javafx.scene.chart.CategoryAxis;**

**import javafx.scene.chart.LineChart;**

**import javafx.scene.chart.NumberAxis;**

**import javafx.scene.chart.XYChart;**

**import javafx.scene.chart.XYChart.Series;**

**import javafx.scene.control.Button;**

**import javafx.scene.control.Label;**

**import javafx.scene.control.Menu;**

**import javafx.scene.control.MenuBar;**

**import javafx.scene.control.MenuButton;**

**import javafx.scene.control.MenuItem;**

**import javafx.scene.control.ScrollPane;**

**import javafx.scene.control.TextField;**

**import javafx.scene.control.Tooltip;**

**import javafx.scene.input.MouseEvent;**

**import javafx.scene.layout.HBox;**

**import javafx.scene.layout.StackPane;**

**import javafx.scene.layout.VBox;**

**import javafx.scene.paint.Color;**

**import javafx.scene.shape.Shape;**

**import javafx.scene.text.Font;**

**import javafx.scene.text.FontWeight;**

**import javafx.stage.\*;**

**public class Quadratic {**

**public void start (final Stage stg) {**

**stg.setTitle("Quadratic Equations");**

**stg.setResizable(false);**

**final HBox hb = new HBox();**

**hb.setLayoutX(-10);**

**hb.setLayoutY(-10);**

**hb.setPrefHeight(750);**

**hb.setPrefWidth(220);**

**hb.setPadding(new Insets(0, 0, 0, 0));**

**hb.setStyle("-fx-background-color: white;");**

**String img = Indexes.class.getResource("b2.png").toExternalForm();**

**MenuBar mb = new MenuBar();**

**mb.setLayoutY(10);**

**mb.setLayoutX(0);**

**mb.setPrefWidth(980);**

**mb.setPrefHeight(30);**

**mb.setStyle("-fx-background-color: lightgray;"**

**+ "-fx-background-opacity: 1;");**

**Menu mbt = new Menu("Export");**

**mbt.setStyle("-fx-font-family: Times;"**

**+ "-fx-font-size: 15;"**

**+ "-fx-font-weight: BOLD;");**

**Menu mbo = new Menu("Refresh");**

**mbo.setStyle("-fx-font-family: Times;"**

**+ "-fx-font-size: 15;"**

**+ "-fx-font-weight: BOLD;");**

**Menu mbb = new Menu("Night View");**

**mbb.setStyle("-fx-font-family: Times;"**

**+ "-fx-font-size: 15;"**

**+ "-fx-font-weight: BOLD;");**

**Menu mbc = new Menu("Basic Tutorial");**

**mbc.setStyle("-fx-font-family: Times;"**

**+ "-fx-font-size: 15;"**

**+ "-fx-font-weight: BOLD;");**

**mb.getMenus().addAll(mbt, mbo, mbb, mbc);**

**final Group gp = new Group();**

**gp.setStyle("-fx-background-color: rgba(15,15,15,0.5);"**

**+ "-fx-background-opacity: 0.5;");**

**hb.getChildren().addAll(gp);**

**Scene sc = new Scene(hb, 960, 700);**

**sc.getStylesheets().addAll(Simultaneous.class.getResource("Style.css").toExternalForm());**

**VBox vb = new VBox();**

**vb.setSpacing(5);**

**vb.setLayoutX(0);**

**vb.setLayoutY(0);**

**vb.setPrefHeight(900);**

**vb.setPrefWidth(250);**

**vb.setPadding(new Insets(130, 0, 0, 0));**

**vb.setStyle("-fx-background-image: url( ' " + img + " '); "**

**+ "-fx-background-position: center center; "**

**+ "-fx-background-repeat: stretch; "**

**+ "-fx-background-size: 250 900;"**

**+ "-fx-border-width: 5px;"**

**+ "-fx-border-color: darkgray;");**

**final Button sub = new Button ("Quadratic Formula");**

**sub.setLayoutX(150);**

**sub.setLayoutY(280);**

**sub.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**sub.setPrefWidth(350);**

**sub.setPrefHeight(60);**

**sub.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0;");**

**sub.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**sub.setStyle("-fx-background-color: white;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**sub.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**sub.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**final Button elim = new Button ("Completing the Squares");**

**elim.setLayoutX(350);**

**elim.setLayoutY(280);**

**elim.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**elim.setPrefWidth(350);**

**elim.setPrefHeight(60);**

**elim.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**elim.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**elim.setStyle("-fx-background-color: white;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**elim.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**elim.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**final Button graph = new Button ("Graphical Solutions");**

**graph.setLayoutX(550);**

**graph.setLayoutY(280);**

**graph.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**graph.setPrefWidth(350);**

**graph.setPrefHeight(60);**

**graph.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**graph.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**graph.setStyle("-fx-background-color: white;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**graph.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**graph.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**final Button gap = new Button ("Return to Home");**

**gap.setLayoutX(550);**

**gap.setLayoutY(280);**

**gap.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**gap.setPrefWidth(350);**

**gap.setPrefHeight(60);**

**gap.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**gap.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**gap.setStyle("-fx-background-color: white;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**gap.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**gap.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0;");**

**}**

**});**

**final Button gay = new Button ("Settings");**

**gay.setLayoutX(550);**

**gay.setLayoutY(280);**

**gay.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**gay.setPrefWidth(350);**

**gay.setPrefHeight(60);**

**gay.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**gay.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**gay.setStyle("-fx-background-color: white;"**

**+ "-fx-text-fill: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**gay.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**gay.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**vb.getChildren().addAll(sub, elim, graph, gap, gay);**

**gap.setOnAction(new EventHandler<ActionEvent>(){**

**public void handle(ActionEvent av) {**

**Indexes ind = new Indexes();**

**ind.start(stg);**

**}**

**});**

**Label info = new Label ("Fill the form below and submit using the options");**

**info.setFont(Font.font("Times New Roman", FontWeight.BOLD, 20));**

**info.setLayoutX(320);**

**info.setLayoutY(40);**

**info.setStyle("-fx-text-fill: white;");**

**Label f1 = new Label ("Value of A: ");**

**f1.setFont(Font.font("Adobe Garamond Pro Bold", FontWeight.BOLD, 20));**

**f1.setLayoutX(320);**

**f1.setLayoutY(100);**

**f1.setStyle("-fx-text-fill: black;");**

**final TextField f = new TextField();**

**f.setFont(Font.font("Cambria", FontWeight.MEDIUM, 17));**

**f.setLayoutX(520);**

**f.setLayoutY(100);**

**f.setPrefWidth(120);**

**f.setStyle("-fx-border-color: none;");**

**Label g1 = new Label ("Value of B: ");**

**g1.setFont(Font.font("Adobe Garamond Pro Bold", FontWeight.BOLD, 20));**

**g1.setLayoutX(320);**

**g1.setLayoutY(150);**

**g1.setStyle("-fx-text-fill: black;");**

**final TextField g = new TextField();**

**g.setFont(Font.font("Cambria", FontWeight.MEDIUM, 17));**

**g.setLayoutX(520);**

**g.setLayoutY(150);**

**g.setPrefWidth(120);**

**Label h1 = new Label ("Value of C: ");**

**h1.setFont(Font.font("Adobe Garamond Pro Bold", FontWeight.BOLD, 20));**

**h1.setLayoutX(320);**

**h1.setLayoutY(200);**

**h1.setStyle("-fx-text-fill: black;");**

**final TextField h = new TextField();**

**h.setFont(Font.font("Cambria", FontWeight.MEDIUM, 17));**

**h.setLayoutX(520);**

**h.setLayoutY(200);**

**h.setPrefWidth(120);**

**elim.setOnAction(new EventHandler <ActionEvent>() {**

**public void handle(ActionEvent eg) {**

**final Double a = Double.parseDouble(f.getText());**

**final Double b = Double.parseDouble(g.getText());**

**final Double c = Double.parseDouble(h.getText());**

**if ((b\*b - 4 \* a \* c) < 0) {**

**double d = b/a;**

**double e = c/a;**

**double eh = d/2;**

**double i = eh \* eh;**

**double j = -e + i;**

**double ka = Math.abs(j);**

**double kb = Math.sqrt(ka);**

**double ek = Math.abs(eh);**

**DecimalFormat df = new DecimalFormat("#.###");**

**String ai = "\\ " + "The \\ Equation \\ is; ";**

**String aj = "\\ " + a + "x^2 \\ + \\ " + b + "x" + "\\ + \\ " + c + "\\ = \\ 0";**

**String ad = "\\ { " + " } \\";**

**String ak = "\\ " + "\\ By \\ Completing \\ the \\ squares";**

**String al = "\\ " + "\\ Divide \\ through \\ by \\ the \\ co-efficient \\ of \\ x^2 ";**

**String am = "\\ " + "\\dfrac {" + a + "x^2}" + "{" + a + "} \\ + " + "\\dfrac {" + b + "x" + "} {" + a + "} \\ + " + "\\dfrac {" + c + "} {" + a + "} \\ " + "= \\ 0";**

**String an = "\\ " + "Simplify \\ to \\ get \\ ";**

**String ao = "\\ " + "x^2 \\ + \\ " + d + "x \\ + \\ " + e + "\\ = \\ 0";**

**String ap = "\\ " + "Transfer \\ the \\ constant \\ term \\ to \\ the \\ RHS \\ ";**

**String aq = "\\ " + "x^2 \\ + " + d + "x \\ = -" + e + "\\";**

**String ar = "\\ " + "Make \\ the \\ LHS \\ a \\ Perfect \\ square \\ by \\ adding \\ the \\ square \\ ";**

**String as = "\\ " + "Of \\ half \\ the \\ co-efficient \\ of \\ x \\ to \\ both \\ sides \\ ";**

**String at = "\\ " + "x^2 + \\ " + d + "x + \\ " + eh + "^2 \\ = \\ - " + e + "\\ + \\ " + eh + "^2";**

**String au = "\\ " + "x^2 + \\ " + d + "x + \\ " + i + "\\ = \\ -" + e + " + " + i + "\\";**

**String av = "\\ " + "Factorize \\ the \\ LHS \\";**

**String aw = "\\ " + "( \\ x + " + eh + "\\ )^2 \\ = " + j + "\\";**

**String ax = "\\ " + "\\ Take \\ the \\ square \\ root \\ of \\ both \\ sides \\ to \\ get \\";**

**String ay = "\\ " + "x \\ + " + eh + "\\ = \\ +/- \\ ( \\ i \\ \* " + df.format(kb) + "\\ )";**

**String az = "\\ " + "x \\ = \\ " + ek + " + (" + df.format(kb) + "i) \\";**

**String bi = "\\ " + "Or \\ x = \\ " + ek + " - (" + df.format(kb) + "i)";**

**String bj = "\\ " + "The \\ roots \\ of \\ x \\ are \\ Imaginary \\";**

**// Convert all Strings to one TeXFormula String**

**TeXFormula ef = new TeXFormula(ai + ad + aj + ad + ak + ad + al + ad + am + ad + an + ad + ao + ad +**

**ap + ad + aq + ad + ar + ad + as + ad + at + ad + au + ad + av + ad + aw**

**+ ad + ax + ad + ay + ad + az + ad + bi + ad + bj) ;**

**// Convert String to Icon**

**TeXIcon ti = ef.createTeXIcon(TeXConstants.DELIM\_SQUARE\_BRACKET, 18);**

**BufferedImage bo = new BufferedImage(ti.getIconWidth(), ti.getIconHeight(), BufferedImage.TYPE\_4BYTE\_ABGR);**

**ti.setInsets(new java.awt.Insets (1, 3, 0, 0));**

**ti.paintIcon(new JLabel(), bo.getGraphics(), 0, 0);**

**ti.setIconWidth(300, TeXConstants.ALIGN\_LEFT);**

**ti.setForeground(java.awt.Color.BLACK);**

**// Import SwingNode to hold Swing Code**

**SwingNode sn = new SwingNode();**

**// Swing TextPane**

**JTextPane ht = new JTextPane();**

**ht.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 0, java.awt.Color.WHITE));**

**ht.setPreferredSize(new Dimension(550, 570));**

**ht.insertIcon(ti);**

**ht.setForeground(java.awt.Color.BLUE);**

**ht.setBackground(java.awt.Color.WHITE);**

**ht.setVisible(true);**

**ht.setMargin(new java.awt.Insets(0,0,0,0));**

**ht.setFocusable(false);**

**ht.setEditable(false);**

**// JPanel --- Swing Component**

**JPanel jp = new JPanel();**

**jp.setLayout(new FlowLayout());**

**GridBagConstraints hi = new GridBagConstraints();**

**hi.fill = GridBagConstraints.FIRST\_LINE\_START;**

**jp.add(ht, hi);**

**jp.setBackground(java.awt.Color.WHITE);**

**// SwingNode to hold JPanel**

**sn.setContent(jp);**

**sn.setLayoutX(50);**

**sn.setLayoutY(300);**

**// ScrollPane to Hold SwingNode**

**ScrollPane scp = new ScrollPane(sn);**

**scp.setContent(sn);**

**scp.setFitToWidth(true);**

**scp.setPrefHeight(350);**

**scp.setPrefWidth(570);**

**scp.setLayoutX(320);**

**scp.setLayoutY(250);**

**scp.setVisible(true);**

**// Add all to Layout**

**gp.getChildren().add(scp);**

**}**

**else if ((b\*b - 4 \* a \* c) >= 0) {**

**@SuppressWarnings("unused")**

**Double d = b/a;**

**Double e = c/a;**

**Double eh = d/2;**

**Double i = eh \* eh;**

**Double j = -e + i;**

**Double k = Math.sqrt(Math.abs(j));**

**Double l = -eh + k;**

**Double m = -eh -k;**

**DecimalFormat df = new DecimalFormat("#.###");**

**String ai = "\\ " + "The \\ Equation \\ is; ";**

**String aj = "\\ " + a + "x^2 \\ + \\ " + b + "x" + "\\ + \\ " + c + "\\ = \\ 0";**

**String ad = "\\ { " + " } \\";**

**String ak = "\\ " + "\\ By \\ Completing \\ the \\ squares";**

**String al = "\\ " + "\\ Divide \\ through \\ by \\ the \\ co-efficient \\ of \\ x^2 ";**

**String am = "\\ " + "\\dfrac {" + a + "x^2}" + "{" + a + "} \\ + " + "\\dfrac {" + b + "x" + "} {" + a + "} \\ + " + "\\dfrac {" + c + "} {" + a + "} \\ " + "= \\ 0";**

**String an = "\\ " + "Simplify \\ to \\ get \\ ";**

**String ao = "\\ " + "x^2 \\ + \\ " + d + "x \\ + \\ " + e + "\\ = \\ 0";**

**String ap = "\\ " + "Transfer \\ the \\ constant \\ term \\ to \\ the \\ RHS \\ ";**

**String aq = "\\ " + "x^2 \\ + " + d + "x \\ = -" + e + "\\";**

**String ar = "\\ " + "Make \\ the \\ LHS \\ a \\ Perfect \\ square \\ by \\ adding \\ the \\ square \\ ";**

**String as = "\\ " + "Of \\ half \\ the \\ co-efficient \\ of \\ x \\ to \\ both \\ sides \\ ";**

**String at = "\\ " + "x^2 + \\ " + d + "x + \\ " + eh + "^2 \\ = \\ - " + e + "\\ + \\ " + eh + "^2";**

**String au = "\\ " + "x^2 + \\ " + d + "x + \\ " + i + "\\ = \\ -" + e + " + " + i + "\\";**

**String av = "\\ " + "Factorize \\ the \\ LHS \\";**

**String aw = "\\ " + "( \\ x + " + eh + "\\ )^2 \\ = " + j + "\\";**

**String ax = "\\ " + "\\ Take \\ the \\ square \\ root \\ of \\ both \\ sides \\ to \\ get ";**

**String ay = "\\ " + "x \\ + " + eh + " \\ = \\ +/- \\ " + k + "\\";**

**String az = "\\ " + "x \\ = \\ - " + eh + " + \\ " + k + "\\";**

**String bi = "\\ " + "Or \\ x = \\ - " + eh + " - \\ " + k + "\\";**

**String bj = "\\ " + "x \\ = " + df.format(l) + "\\ or \\ " + df.format(m) + "\\";**

**// Convert all Strings to one TeXFormula String**

**TeXFormula ef = new TeXFormula(ai + aj + ad + ak + ad + al + ad + am + ad + an + ad + ao**

**+ ad + ap + ad + aq + ad + ar + ad + as + ad + at + ad + au + ad + av + ad + aw + ad + ax + ad**

**+ ay + ad + az + ad + bi + ad + bj);**

**// Convert String to Icon**

**TeXIcon ti = ef.createTeXIcon(TeXConstants.DELIM\_SQUARE\_BRACKET, 18);**

**BufferedImage bo = new BufferedImage(ti.getIconWidth(), ti.getIconHeight(), BufferedImage.TYPE\_4BYTE\_ABGR);**

**ti.setInsets(new java.awt.Insets (1, 3, 0, 0));**

**ti.paintIcon(new JLabel(), bo.getGraphics(), 0, 0);**

**ti.setIconWidth(300, TeXConstants.ALIGN\_LEFT);**

**ti.setForeground(java.awt.Color.BLACK);**

**// Import SwingNode to hold Swing Code**

**SwingNode sn = new SwingNode();**

**// Swing TextPane**

**JTextPane ht = new JTextPane();**

**ht.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 0, java.awt.Color.WHITE));**

**ht.setPreferredSize(new Dimension(550, 570));**

**ht.insertIcon(ti);**

**ht.setForeground(java.awt.Color.BLUE);**

**ht.setBackground(java.awt.Color.WHITE);**

**ht.setVisible(true);**

**ht.setMargin(new java.awt.Insets(0,0,0,0));**

**ht.setFocusable(false);**

**ht.setEditable(false);**

**// JPanel --- Swing Component**

**JPanel jp = new JPanel();**

**jp.setLayout(new FlowLayout());**

**GridBagConstraints hi = new GridBagConstraints();**

**hi.fill = GridBagConstraints.FIRST\_LINE\_START;**

**jp.add(ht, hi);**

**jp.setBackground(java.awt.Color.WHITE);**

**// SwingNode to hold JPanel**

**sn.setContent(jp);**

**sn.setLayoutX(50);**

**sn.setLayoutY(300);**

**// ScrollPane to Hold SwingNode**

**ScrollPane scp = new ScrollPane(sn);**

**scp.setContent(sn);**

**scp.setFitToWidth(true);**

**scp.setPrefHeight(350);**

**scp.setPrefWidth(570);**

**scp.setLayoutX(320);**

**scp.setLayoutY(250);**

**scp.setVisible(true);**

**// Add all to Layout**

**gp.getChildren().add(scp);**

**}**

**}**

**});**

**sub.setOnAction(new EventHandler <ActionEvent>() {**

**public void handle(ActionEvent eg) {**

**final Double a = Double.parseDouble(f.getText());**

**final Double b = Double.parseDouble(g.getText());**

**final Double c = Double.parseDouble(h.getText());**

**if((b\*b - 4 \* a \* c) < 0 && b <= 0) {**

**Double d = Double.parseDouble(f.getText());**

**Double e = Double.parseDouble(g.getText());**

**Double f = Double.parseDouble(h.getText());**

**Double i = e \* e;**

**Double j = 4 \* d \* f;**

**Double k = 2 \* d;**

**Double l = i - j;**

**Double m = Math.sqrt(Math.abs(l)); /\* Piece of Code for finding square-root \*/**

**Double ed = Math.abs(e);**

**DecimalFormat df = new DecimalFormat("#.###");**

**String ball = "\\ The \\ Formula \\ is: " ;**

**String one = "\\ " + "\\dfrac {-b \u00B1 \u221A(b^2 - 4ac)} {2a}" + " \\";**

**String add= "\\ " + "And \\ the \\ Equation \\ is: ";**

**String two = "\\ { " + d + "x^2 + \\ " + e + "x + \\ " + f + " \\ = \\ 0" + "} \\" ;**

**String twomid = "\\ Here, \\ a \\ = " + d + " \\ b \\ = " + e + "\\ and \\ c \\ = " + f + " \\";**

**String tree = "\\ " + "By \\ our \\ formula, \\ we \\ have ";**

**String four = "\\ " + "\\dfrac { -" + e + '\u00B1' + '\u221A' + "( " + e + "^2 - 4 \* " + d + " \* " + f + ")} {2 \*" + d + "}" + " \\" ;**

**String bw = "\\ " + "Simplify \\ Further \\ to \\ get" + " \\";**

**String five = "\\ " + "\\dfrac {-" + e + '\u00B1' + '\u221A' + "( " + i + " - " + j + ")} {" + k + "}" + " \\";**

**String six = "\\ " + "\\dfrac {-" + e + '\u00B1' + '\u221A' + "( " + l + ")} {" + k + "}" + " \\";**

**String seven = "\\ " + "\\dfrac {-" + e + '\u00B1' + "i \\ \* " + l + "} {" + k + "}" + " \\";**

**String mid = "\\ " + "Solve \\ differently \\ and \\ we \\ have" + " \\";**

**String eight = "\\ x \\ = " + "\\dfrac {" + ed + " + " + df.format(m) + "i" + "} {" + k + "}" + " \\";**

**String nine = "\\ " + " or " + " \\" + "\\ x \\ = " + "\\dfrac {" + ed + " - " + df.format(m) + "i" + "} {" + k + "}" + " \\";**

**String ten = "\\ " + " Your \\ answers \\ are \\ imaginary \\";**

**TeXFormula eef = new TeXFormula(ball + one + add + two + twomid + tree + four +**

**bw + five + six + seven + mid + eight + nine + ten);**

**/\* TeXIcon, to make the compiled string as a picture \*/**

**TeXIcon ti = eef.createTeXIcon(TeXConstants.DELIM\_SQUARE\_BRACKET, 19);**

**/\* Buffered Image to edit the Picture's features \*/**

**BufferedImage bud = new BufferedImage(ti.getIconWidth(), ti.getIconHeight(), BufferedImage.TYPE\_4BYTE\_ABGR);**

**/\* TeXICon code to modify the appearance of the picture, including the position \*/**

**ti.setInsets(new java.awt.Insets (-1, -1, 0, 0));**

**/\* TeXICon to add graphic quality to the picture \*/**

**ti.paintIcon(new JLabel(), bud.getGraphics(), 0, 0);**

**ti.setForeground(java.awt.Color.BLACK);**

**/\* SwingNode used to accommodate swing components in a JavaFX application \*/**

**SwingNode sn = new SwingNode();**

**/\* Swing TextPane to include the TeXICon produced \*/**

**JTextPane ht = new JTextPane(); /\* Initialization \*/**

**ht.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 0, java.awt.Color.DARK\_GRAY)); /\* Border settings between the Pane and the outside \*/**

**ht.insertIcon(ti); /\* The TextPane accommodates the Icon \*/**

**ht.setPreferredSize(new Dimension(550, 550)); /\* The Size of the TextPane \*/**

**ht.setForeground(java.awt.Color.BLUE); /\* Set the Text Color of the Icon \*/**

**ht.setBackground(java.awt.Color.WHITE); /\* Set the Background color of the TextPane \*/**

**ht.setVisible(true); /\* Make this visible, although it is always by default \*/**

**ht.setMargin(new java.awt.Insets(0,0,0,0)); /\* Set Margin between it and the Icon \*/**

**ht.setFocusable(true); /\* Remove the ability to focus \*/**

**ht.setEditable(false); /\* You cannot edit the items in the TextPane \*/**

**JPanel jp = new JPanel(); /\* The Framework for the TextPane \*/**

**jp.setLayout(new FlowLayout()); /\* The Layout of the Framework \*/**

**GridBagConstraints hi = new GridBagConstraints(); /\* The Constraints of the Layout \*/**

**hi.fill = GridBagConstraints.FIRST\_LINE\_START; /\* Positioning using the Constraints \*/**

**jp.add(ht, hi); /\* Apply the Constraints to the Layout and the Framework \*/**

**jp.setBackground(java.awt.Color.WHITE); /\* Background settings for the Framework \*/**

**sn.setContent(jp); /\* Add the Framework to the Node \*/**

**sn.setLayoutX(40); /\* Modify the positions of the Node \*/**

**sn.setLayoutY(300);**

**ScrollPane scp = new ScrollPane(sn); /\* Add the Node to the ScrollPane to make it Scrollable \*/**

**scp.setLayoutX(320); /\* Modify the Position of the ScrollPane \*/**

**scp.setLayoutY(250);**

**scp.setPrefWidth(570); /\* Set the width and height of the ScrollPane \*/**

**scp.setPrefHeight(350);**

**scp.setFitToWidth(true); /\* Make the Width of the Node the width of the Pane \*/**

**scp.setVisible(true); /\* Make the Pane visible \*/**

**scp.setStyle("-fx-background-color: white;");**

**gp.getChildren().add(scp); /\* Add everything you did to the Chief Layout \*/**

**}**

**else if((b\*b - 4 \* a \* c) < 0 && b >= 0) {**

**Double d = Double.parseDouble(f.getText());**

**Double e = Double.parseDouble(g.getText());**

**Double f = Double.parseDouble(h.getText());**

**Double i = e \* e;**

**Double j = 4 \* d \* f;**

**Double k = 2 \* d;**

**Double l = i - j;**

**Double m = Math.sqrt(Math.abs(l)); /\* Piece of Code for finding square-root \*/**

**Double ed = Math.abs(e);**

**DecimalFormat df = new DecimalFormat("#.###");**

**String ball = "\\ The \\ Formula \\ is: " ;**

**String one = "\\ " + "\\dfrac {-b \u00B1 \u221A(b^2 - 4ac)} {2a}" + " \\";**

**String add= "\\ " + "And \\ the \\ Equation \\ is: ";**

**String two = "\\ { " + d + "x^2 + \\ " + e + "x + \\ " + f + " \\ = \\ 0" + "} \\" ;**

**String twomid = "\\ Here, \\ a \\ = " + d + " \\ b \\ = " + e + "\\ and \\ c \\ = " + f + " \\";**

**String tree = "\\ " + "By \\ our \\ formula, \\ we \\ have ";**

**String four = "\\ " + "\\dfrac { -" + e + '\u00B1' + '\u221A' + "( " + e + "^2 - 4 \* " + d + " \* " + f + ")} {2 \*" + d + "}" + " \\" ;**

**String bw = "\\ " + "Simplify \\ Further \\ to \\ get" + " \\";**

**String five = "\\ " + "\\dfrac {-" + e + '\u00B1' + '\u221A' + "( " + i + " - " + j + ")} {" + k + "}" + " \\";**

**String six = "\\ " + "\\dfrac {-" + e + '\u00B1' + '\u221A' + "( " + l + ")} {" + k + "}" + " \\";**

**String seven = "\\ " + "\\dfrac {-" + e + '\u00B1' + "i \\ \* " + l + "} {" + k + "}" + " \\";**

**String mid = "\\ " + "Solve \\ differently \\ and \\ we \\ have" + " \\";**

**String eight = "\\ x \\ = " + "\\dfrac { \\ -" + ed + " + " + df.format(m) + "i" + "} {" + k + "}" + " \\";**

**String nine = "\\ " + " or " + " \\" + "\\ x \\ = " + "\\dfrac { \\ -" + ed + " - " + df.format(m) + "i" + "} {" + k + "}" + " \\";**

**String ten = "\\ " + " Your \\ answers \\ are \\ imaginary \\";**

**TeXFormula eef = new TeXFormula(ball + one + add + two + twomid + tree + four +**

**bw + five + six + seven + mid + eight + nine + ten);**

**/\* TeXIcon, to make the compiled string as a picture \*/**

**TeXIcon ti = eef.createTeXIcon(TeXConstants.DELIM\_SQUARE\_BRACKET, 19);**

**/\* Buffered Image to edit the Picture's features \*/**

**BufferedImage bud = new BufferedImage(ti.getIconWidth(), ti.getIconHeight(), BufferedImage.TYPE\_4BYTE\_ABGR);**

**/\* TeXICon code to modify the appearance of the picture, including the position \*/**

**ti.setInsets(new java.awt.Insets (-1, -1, 0, 0));**

**/\* TeXICon to add graphic quality to the picture \*/**

**ti.paintIcon(new JLabel(), bud.getGraphics(), 0, 0);**

**ti.setForeground(java.awt.Color.BLACK);**

**/\* SwingNode used to accommodate swing components in a JavaFX application \*/**

**SwingNode sn = new SwingNode();**

**/\* Swing TextPane to include the TeXICon produced \*/**

**JTextPane ht = new JTextPane(); /\* Initialization \*/**

**ht.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 0, java.awt.Color.DARK\_GRAY)); /\* Border settings between the Pane and the outside \*/**

**ht.insertIcon(ti); /\* The TextPane accommodates the Icon \*/**

**ht.setPreferredSize(new Dimension(550, 550)); /\* The Size of the TextPane \*/**

**ht.setForeground(java.awt.Color.BLUE); /\* Set the Text Color of the Icon \*/**

**ht.setBackground(java.awt.Color.WHITE); /\* Set the Background color of the TextPane \*/**

**ht.setVisible(true); /\* Make this visible, although it is always by default \*/**

**ht.setMargin(new java.awt.Insets(0,0,0,0)); /\* Set Margin between it and the Icon \*/**

**ht.setFocusable(true); /\* Remove the ability to focus \*/**

**ht.setEditable(false); /\* You cannot edit the items in the TextPane \*/**

**JPanel jp = new JPanel(); /\* The Framework for the TextPane \*/**

**jp.setLayout(new FlowLayout()); /\* The Layout of the Framework \*/**

**GridBagConstraints hi = new GridBagConstraints(); /\* The Constraints of the Layout \*/**

**hi.fill = GridBagConstraints.FIRST\_LINE\_START; /\* Positioning using the Constraints \*/**

**jp.add(ht, hi); /\* Apply the Constraints to the Layout and the Framework \*/**

**jp.setBackground(java.awt.Color.WHITE); /\* Background settings for the Framework \*/**

**sn.setContent(jp); /\* Add the Framework to the Node \*/**

**sn.setLayoutX(40); /\* Modify the positions of the Node \*/**

**sn.setLayoutY(300);**

**ScrollPane scp = new ScrollPane(sn); /\* Add the Node to the ScrollPane to make it Scrollable \*/**

**scp.setLayoutX(320); /\* Modify the Position of the ScrollPane \*/**

**scp.setLayoutY(250);**

**scp.setPrefWidth(570); /\* Set the width and height of the ScrollPane \*/**

**scp.setPrefHeight(350);**

**scp.setFitToWidth(true); /\* Make the Width of the Node the width of the Pane \*/**

**scp.setVisible(true); /\* Make the Pane visible \*/**

**scp.setStyle("-fx-background-color: white;");**

**gp.getChildren().add(scp); /\* Add everything you did to the Chief Layout \*/**

**}**

**else if ((b\*b - 4 \* a \* c) >= 0) {**

**Double d = Double.parseDouble(f.getText());**

**Double e = Double.parseDouble(g.getText());**

**Double f = Double.parseDouble(h.getText());**

**Double i = e \* e;**

**Double j = 4 \* d \* f;**

**Double k = 2 \* d;**

**Double l = i - j;**

**Double m = Math.sqrt(Math.abs(l)); /\* Piece of Code for finding square-root \*/**

**Double n = (-e + m) / k;**

**Double o = (-e - m) / k;**

**DecimalFormat df = new DecimalFormat("#.###");**

**String ball = "\\ The \\ Formula \\ is: " ;**

**String one = "\\ " + "\\dfrac {-b \u00B1 \u221A(b^2 - 4ac)} {2a}" + " \\";**

**String add= "\\ " + "And \\ the \\ Equation \\ is: ";**

**String two = "\\ { " + d + "x^2 + \\ " + e + "x + \\ " + f + " \\ = \\ 0" + "} \\" ;**

**String twomid = "\\ Here, \\ a \\ = " + d + " \\ b \\ = " + e + "\\ and \\ c \\ = " + f + " \\";**

**String tree = "\\ " + "By \\ our \\ formula, \\ we \\ have ";**

**String four = "\\ " + "\\dfrac { -" + e + '\u00B1' + '\u221A' + "( " + e + "^2 - 4 \* " + d + " \* " + f + ")} {2 \*" + d + "}" + " \\" ;**

**String bw = "\\ " + "Simplify \\ Further \\ to \\ get" + " \\";**

**String five = "\\ " + "\\dfrac {-" + e + '\u00B1' + '\u221A' + "( " + i + " - " + j + ")} {" + k + "}" + " \\";**

**String six = "\\ " + "\\dfrac {-" + e + '\u00B1' + '\u221A' + "( " + l + ")} {" + k + "}" + " \\";**

**String seven = "\\ " + "\\dfrac {-" + e + '\u00B1' + m + "} {" + k + "}" + " \\";**

**String mid = "\\ " + "Solve \\ differently \\ and \\ we \\ have" + " \\";**

**String eight = "\\ " + "\\dfrac {-" + e + " + " + m + "} {" + k + "}" + " \\";**

**String nine = "\\ " + " or " + " \\" + "\\ " + "\\dfrac {-" + e + " - " + m + "} {" + k + "}" + " \\";**

**String ten = "\\ " + " and \\ x =" + df.format(n) + " \\";**

**String done = "\\ " + " or \\ " + df.format(o);**

**TeXFormula eef = new TeXFormula(ball + one + add + two + twomid + tree + four +**

**bw + five + six + seven + mid + eight + nine + ten + done);**

**/\* TeXIcon, to make the compiled string as a picture \*/**

**TeXIcon ti = eef.createTeXIcon(TeXConstants.DELIM\_SQUARE\_BRACKET, 19);**

**/\* Buffered Image to edit the Picture's features \*/**

**BufferedImage bud = new BufferedImage(ti.getIconWidth(), ti.getIconHeight(), BufferedImage.TYPE\_4BYTE\_ABGR);**

**/\* TeXICon code to modify the appearance of the picture, including the position \*/**

**ti.setInsets(new java.awt.Insets (-1, -1, 0, 0));**

**/\* TeXICon to add graphic quality to the picture \*/**

**ti.paintIcon(new JLabel(), bud.getGraphics(), 0, 0);**

**ti.setForeground(java.awt.Color.black);**

**/\* SwingNode used to accommodate swing components in a JavaFX application \*/**

**SwingNode sn = new SwingNode();**

**/\* Swing TextPane to include the TeXICon produced \*/**

**JTextPane ht = new JTextPane(); /\* Initialization \*/**

**ht.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 0, java.awt.Color.DARK\_GRAY)); /\* Border settings between the Pane and the outside \*/**

**ht.insertIcon(ti); /\* The TextPane accommodates the Icon \*/**

**ht.setPreferredSize(new Dimension(550, 550)); /\* The Size of the TextPane \*/**

**ht.setForeground(java.awt.Color.BLUE); /\* Set the Text Color of the Icon \*/**

**ht.setBackground(java.awt.Color.white); /\* Set the Background color of the TextPane \*/**

**ht.setVisible(true); /\* Make this visible, although it is always by default \*/**

**ht.setMargin(new java.awt.Insets(0,0,0,0)); /\* Set Margin between it and the Icon \*/**

**ht.setFocusable(true); /\* Remove the ability to focus \*/**

**ht.setEditable(false); /\* You cannot edit the items in the TextPane \*/**

**JPanel jp = new JPanel(); /\* The Framework for the TextPane \*/**

**jp.setLayout(new FlowLayout()); /\* The Layout of the Framework \*/**

**GridBagConstraints hi = new GridBagConstraints(); /\* The Constraints of the Layout \*/**

**hi.fill = GridBagConstraints.FIRST\_LINE\_START; /\* Positioning using the Constraints \*/**

**jp.add(ht, hi); /\* Apply the Constraints to the Layout and the Framework \*/**

**jp.setBackground(java.awt.Color.WHITE); /\* Background settings for the Framework \*/**

**sn.setContent(jp); /\* Add the Framework to the Node \*/**

**sn.setLayoutX(40); /\* Modify the positions of the Node \*/**

**sn.setLayoutY(300);**

**ScrollPane scp = new ScrollPane(sn); /\* Add the Node to the ScrollPane to make it Scrollable \*/**

**scp.setLayoutX(320); /\* Modify the Position of the ScrollPane \*/**

**scp.setLayoutY(250);**

**scp.setPrefWidth(570); /\* Set the width and height of the ScrollPane \*/**

**scp.setPrefHeight(350);**

**scp.setFitToWidth(true); /\* Make the Width of the Node the width of the Pane \*/**

**scp.setVisible(true); /\* Make the Pane visible \*/**

**scp.setStyle("-fx-background-color: white;");**

**gp.getChildren().add(scp); /\* Add everything you did to the Chief Layout \*/**

**}**

**}**

**});**

**graph.setOnAction(new EventHandler<ActionEvent>(){**

**@SuppressWarnings({ "rawtypes", "unchecked" })**

**public void handle(ActionEvent ev) {**

**final Stage sag = new Stage();**

**sag.setTitle("Graph Co-ordinates");**

**Group gd = new Group();**

**Scene sc = new Scene(gd, 300, 300);**

**sag.setScene(sc);**

**sag.setResizable(false);**

**sag.show();**

**Label ol = new Label ("Enter the boundaries");**

**ol.setLayoutX(50);**

**ol.setLayoutY(20);**

**ol.setFont(Font.font("Times New Roman", FontWeight.BOLD, 17));**

**final TextField a = new TextField();**

**a.setTooltip(new Tooltip("Lower Bound"));**

**a.setLayoutX(50);**

**a.setPrefWidth(70);**

**a.setLayoutY(80);**

**final TextField mb = new TextField();**

**mb.setLayoutX(50);**

**mb.setLayoutY(140);**

**mb.setPrefWidth(70);**

**mb.setTooltip(new Tooltip("Sign (+/-)"));**

**final TextField b = new TextField();**

**b.setTooltip(new Tooltip("Upper Bound"));**

**b.setLayoutX(50);**

**b.setLayoutY(200);**

**b.setPrefWidth(70);**

**Button plot = new Button("Plot Chart");**

**plot.setLayoutX(50);**

**plot.setLayoutY(260);**

**plot.setFont(Font.font("Cambria", FontWeight.BOLD, 16));**

**gd.getChildren().addAll(ol, a, mb, b, plot);**

**plot.setOnAction(new EventHandler<ActionEvent>(){**

**public void handle (ActionEvent th) {**

**if(mb.getText().equals("+")) {**

**sag.close();**

**CategoryAxis xaxis = new CategoryAxis();**

**xaxis.setLabel("X-axis");**

**NumberAxis yaxis = new NumberAxis();**

**yaxis.setLabel("Y-axis");**

**LineChart lct = new LineChart(xaxis, yaxis);**

**lct.setData(getData());**

**lct.setStyle("-fx-background-color: white;");**

**StackPane root = new StackPane();**

**root.getChildren().add(lct);**

**root.setLayoutX(320);**

**root.setLayoutY(250);**

**root.setPrefSize(570, 350);**

**gp.getChildren().add(root);**

**}**

**else if (mb.getText().equals("-")) {**

**sag.close();**

**CategoryAxis xaxis = new CategoryAxis();**

**xaxis.setLabel("X-axis");**

**NumberAxis yaxis = new NumberAxis();**

**yaxis.setLabel("Y-axis");**

**LineChart lct = new LineChart(xaxis, yaxis);**

**lct.setData(postData());**

**lct.setStyle("-fx-background-color: white;");**

**StackPane root = new StackPane();**

**root.getChildren().add(lct);**

**root.setLayoutX(320);**

**root.setLayoutY(250);**

**root.setPrefHeight(350);**

**root.setPrefWidth(570);**

**gp.getChildren().add(root);**

**}**

**}**

**@SuppressWarnings({ "unchecked", "rawtypes" })**

**private ObservableList<XYChart.Series<String, Double>> getData() {**

**ObservableList<XYChart.Series<String, Double>> data = FXCollections.observableArrayList();**

**Series<String, Double> squad = new Series<String, Double>();**

**double ant = Double.parseDouble(f.getText());**

**double brant = Double.parseDouble(g.getText());**

**double crant = Double.parseDouble(h.getText());**

**double ace = Double.parseDouble(a.getText());**

**double aid = Double.parseDouble(b.getText());**

**squad.setName("Graphical Solution for " + f.getText() + "x\u00B2 + " + g.getText() + "x + " + h.getText() + " = 0");**

**for (double i = ace; i <= aid; i+=0.25) {**

**double squadValue = ((ant \* (i \* i)) + (brant \* i) + crant);**

**squad.getData().add(new XYChart.Data(Double.toString(i), squadValue));**

**}**

**data.add(squad);**

**return data;**

**}**

**@SuppressWarnings({ "unchecked", "rawtypes" })**

**private ObservableList<XYChart.Series<String, Double>> postData() {**

**ObservableList<XYChart.Series<String, Double>> data = FXCollections.observableArrayList();**

**Series<String, Double> squad = new Series<String, Double>();**

**double ant = Double.parseDouble(f.getText());**

**double brant = Double.parseDouble(g.getText());**

**double crant = Double.parseDouble(h.getText());**

**double ace = Double.parseDouble(a.getText());**

**double aid = Double.parseDouble(b.getText());**

**squad.setName("Graphical Solution for " + f.getText() + "x\u00B2 + " + g.getText() + "x + " + h.getText() + " = 0");**

**for (double i = ace ; i >= aid; i-=0.25) {**

**double squadValue = ((ant \* (i \* i)) + (brant \* i) + crant);**

**squad.getData().add(new XYChart.Data(Double.toString(i), squadValue));**

**}**

**data.add(squad);**

**return data;**

**}**

**});**

**}**

**});**

**gp.getChildren().addAll(info, vb, f1, f, g1, g, h1, h, mb);**

**stg.setScene(sc);**

**stg.show();**

**}**

**}**

**APPENDIX D: Source Code Listing for Cubic Equations**

**import java.awt.Dimension;**

**import java.awt.FlowLayout;**

**import java.awt.GridBagConstraints;**

**import java.awt.image.BufferedImage;**

**import java.text.DecimalFormat;**

**import javax.swing.BorderFactory;**

**import javax.swing.JLabel;**

**import javax.swing.JOptionPane;**

**import javax.swing.JPanel;**

**import javax.swing.JTextPane;**

**import org.scilab.forge.jlatexmath.TeXConstants;**

**import org.scilab.forge.jlatexmath.TeXFormula;**

**import org.scilab.forge.jlatexmath.TeXIcon;**

**import javafx.collections.FXCollections;**

**import javafx.collections.ObservableList;**

**import javafx.embed.swing.SwingNode;**

**import javafx.event.ActionEvent;**

**import javafx.event.EventHandler;**

**import javafx.geometry.Insets;**

**import javafx.scene.Group;**

**import javafx.scene.Scene;**

**import javafx.scene.SubScene;**

**import javafx.scene.chart.CategoryAxis;**

**import javafx.scene.chart.LineChart;**

**import javafx.scene.chart.NumberAxis;**

**import javafx.scene.chart.XYChart;**

**import javafx.scene.chart.XYChart.Series;**

**import javafx.scene.control.Button;**

**import javafx.scene.control.Label;**

**import javafx.scene.control.Menu;**

**import javafx.scene.control.MenuBar;**

**import javafx.scene.control.MenuButton;**

**import javafx.scene.control.MenuItem;**

**import javafx.scene.control.ScrollPane;**

**import javafx.scene.control.TextField;**

**import javafx.scene.control.Tooltip;**

**import javafx.scene.input.MouseEvent;**

**import javafx.scene.layout.HBox;**

**import javafx.scene.layout.StackPane;**

**import javafx.scene.layout.VBox;**

**import javafx.scene.paint.Color;**

**import javafx.scene.shape.Shape;**

**import javafx.scene.text.Font;**

**import javafx.scene.text.FontWeight;**

**import javafx.stage.\*;**

**public class Cubic {**

**public void start (final Stage stg) {**

**stg.setTitle("Cubic Equations");**

**stg.setResizable(false);**

**final HBox hb = new HBox();**

**hb.setLayoutX(-10);**

**hb.setLayoutY(-10);**

**hb.setPrefHeight(750);**

**hb.setPrefWidth(220);**

**hb.setPadding(new Insets(0, 0, 0, 0));**

**hb.setStyle("-fx-background-color: white;");**

**String img = Indexes.class.getResource("b2.png").toExternalForm();**

**MenuBar mb = new MenuBar();**

**mb.setLayoutY(10);**

**mb.setLayoutX(0);**

**mb.setPrefWidth(980);**

**mb.setPrefHeight(30);**

**mb.setStyle("-fx-background-color: lightgray;"**

**+ "-fx-background-opacity: 1;");**

**Menu mbt = new Menu("Main Menu");**

**mbt.setStyle("-fx-font-family: Times;"**

**+ "-fx-font-size: 15;"**

**+ "-fx-font-weight: BOLD;");**

**mbt.setOnAction(new EventHandler<ActionEvent>(){**

**public void handle (ActionEvent hm){**

**Indexes ind = new Indexes();**

**ind.start(stg);**

**}**

**});**

**mb.getMenus().addAll(mbt);**

**hb.setStyle("-fx-background-color: white;"**

**+ "-fx-background-size: 980 730");**

**final Group gp = new Group();**

**gp.setStyle("-fx-background-color: rgba(15,15,15,0.5);"**

**+ "-fx-background-opacity: 0.5;");**

**hb.getChildren().addAll(gp);**

**Scene sc = new Scene(hb, 960, 700);**

**VBox vb = new VBox();**

**vb.setSpacing(8);**

**vb.setLayoutX(0);**

**vb.setLayoutY(0);**

**vb.setPrefHeight(800);**

**vb.setPrefWidth(220);**

**vb.setPadding(new Insets(100, 0, 0, 0));**

**vb.setStyle("-fx-background-color: rgba(10,10,10,0.5);"**

**+ "-fx-background-opacity: 0.5;"**

**+ "-fx-border-color: darkgray;"**

**+ "-fx-border-width: 8px;");**

**final Button lb = new Button ("Choice Methods");**

**lb.setPrefHeight(100);**

**lb.setPrefWidth(350);**

**lb.setFont(Font.font("Cursive", FontWeight.BOLD, 18));**

**lb.setStyle("-fx-text-align: center;"**

**+ "-fx-background-color: transparent;"**

**+ "-fx-text-fill: white;");**

**final Button sub = new Button ("Quadratic Method");**

**sub.setPrefWidth(350);**

**sub.setPrefHeight(60);**

**sub.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**sub.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**sub.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**sub.setStyle("-fx-text-fill: orange;");**

**}**

**});**

**sub.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**sub.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**final Button elim = new Button ("Using Cubic Formula");**

**elim.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**elim.setPrefWidth(350);**

**elim.setPrefHeight(60);**

**elim.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**elim.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**elim.setStyle("-fx-text-fill: orange;");**

**}**

**});**

**elim.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**elim.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**final Button graph = new Button ("Graphical Solutions");**

**graph.setFont(Font.font("Cambria", FontWeight.BOLD, 15));**

**graph.setPrefWidth(350);**

**graph.setPrefHeight(60);**

**graph.setStyle("-fx-text-fill: white;"**

**+ "-fx-background-color: black;"**

**+ "-fx-background-radius: 0,0,0,0");**

**graph.setOnMouseEntered(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**graph.setStyle("-fx-text-fill: orange;");**

**}**

**});**

**graph.setOnMouseExited(new EventHandler<MouseEvent>(){**

**public void handle (MouseEvent hg) {**

**graph.setStyle("-fx-background-color: black;"**

**+ "-fx-text-fill: white;"**

**+ "-fx-background-radius: 0,0,0,0");**

**}**

**});**

**vb.getChildren().addAll(lb, sub, elim, graph);**

**//------------------------------------------------------------------------------------------------------------------------------------------------------//**

**Label info = new Label ("Fill the form below and submit using the options");**

**info.setFont(Font.font("Times New Roman", FontWeight.BOLD, 20));**

**info.setLayoutX(290);**

**info.setLayoutY(70);**

**info.setStyle("-fx-text-fill: black;");**

**Label f1 = new Label ("Value of A: ");**

**f1.setFont(Font.font("Times New Roman", FontWeight.BOLD, 20));**

**f1.setLayoutX(290);**

**f1.setLayoutY(120);**

**f1.setStyle("-fx-text-fill: black;");**

**final TextField f = new TextField();**

**f.setFont(Font.font("Cambria", FontWeight.MEDIUM, 17));**

**f.setLayoutX(420);**

**f.setLayoutY(120);**

**f.setPrefWidth(120);**

**f.setStyle("-fx-border-color: none;");**

**Label g1 = new Label ("Value of B: ");**

**g1.setFont(Font.font("Times New Roman", FontWeight.BOLD, 20));**

**g1.setLayoutX(590);**

**g1.setLayoutY(120);**

**g1.setStyle("-fx-text-fill: black;");**

**final TextField g = new TextField();**

**g.setFont(Font.font("Cambria", FontWeight.MEDIUM, 17));**

**g.setLayoutX(720);**

**g.setLayoutY(120);**

**g.setPrefWidth(120);**

**Label h1 = new Label ("Value of C: ");**

**h1.setFont(Font.font("Times New Roman", FontWeight.BOLD, 20));**

**h1.setLayoutX(290);**

**h1.setLayoutY(160);**

**h1.setStyle("-fx-text-fill: black;");**

**final TextField h = new TextField();**

**h.setFont(Font.font("Cambria", FontWeight.MEDIUM, 17));**

**h.setLayoutX(420);**

**h.setLayoutY(160);**

**h.setPrefWidth(120);**

**Label h2 = new Label ("Value of D: ");**

**h2.setFont(Font.font("Times New Roman", FontWeight.BOLD, 20));**

**h2.setLayoutX(590);**

**h2.setLayoutY(160);**

**h2.setStyle("-fx-text-fill: black;");**

**final TextField gh = new TextField();**

**gh.setFont(Font.font("Cambria", FontWeight.MEDIUM, 17));**

**gh.setLayoutX(720);**

**gh.setLayoutY(160);**

**gh.setPrefWidth(120);**

**sub.setOnAction(new EventHandler <ActionEvent>() {**

**public void handle(ActionEvent eg) {**

**final Double a = Double.parseDouble(f.getText());**

**final Double b = Double.parseDouble(g.getText());**

**final Double c = Double.parseDouble(h.getText());**

**final Double de = Double.parseDouble(gh.getText());**

**if(de != 0) {**

**JOptionPane.showMessageDialog(null, "Cannot solve because constant exists, Try another method", "Error", JOptionPane.INFORMATION\_MESSAGE);**

**}**

**else if((b\*b - 4 \* a \* c) < 0 && b < 0 && de == 0) {**

**Double d = Double.parseDouble(f.getText());**

**Double e = Double.parseDouble(g.getText());**

**Double f = Double.parseDouble(h.getText());**

**Double i = e \* e;**

**Double j = 4 \* d \* f;**

**Double k = 2 \* d;**

**Double l = i - j;**

**Double m = Math.sqrt(Math.abs(l)); /\* Piece of Code for finding square-root \*/**

**Double ed = Math.abs(e);**

**DecimalFormat df = new DecimalFormat("#.###");**

**String ball = "\\ The \\ Formula \\ is: " ;**

**String one = "\\ " + "\\dfrac {-b \u00B1 \u221A(b^2 - 4ac)} {2a}" + " \\";**

**String add= "\\ " + "And \\ the \\ Equation \\ is: ";**

**String two = "\\ { " + d + "x^3 + \\ " + e + "x^2 + \\ " + f + "x \\ = \\ 0" + "} \\" ;**

**String twomid = "\\ Here, \\ a \\ = " + d + " \\ b \\ = " + e + "\\ and \\ c \\ = " + f + " \\";**

**String twomidi = "\\ " + "Factor \\ out \\ x \\ from \\ this \\ equation \\ to \\ make \\ it \\ quadratic \\";**

**String towo = "\\ { " + "x(" + d + "x^2 + \\ " + e + "x + \\ " + f + ") \\ = \\ 0" + "} \\";**

**String tree = "\\ " + "By \\ our \\ formula, \\ we \\ have ";**

**String four = "\\ " + "\\dfrac { -" + e + '\u00B1' + '\u221A' + "( " + e + "^2 - 4 \* " + d + " \* " + f + ")} {2 \*" + d + "}" + " \\" ;**

**String bw = "\\ " + "Simplify \\ Further \\ to \\ get" + " \\";**

**String five = "\\ " + "\\dfrac {-" + e + '\u00B1' + '\u221A' + "( " + i + " - " + j + ")} {" + k + "}" + " \\";**

**String six = "\\ " + "\\dfrac {-" + e + '\u00B1' + '\u221A' + "( " + l + ")} {" + k + "}" + " \\";**

**String seven = "\\ " + "\\dfrac {-" + e + '\u00B1' + "i \\ \* " + l + "} {" + k + "}" + " \\";**

**String mid = "\\ " + "Solve \\ differently \\ and \\ we \\ have" + " \\";**

**String eight = "\\ x \\ = " + "\\dfrac {" + ed + " + " + df.format(m) + "i" + "} {" + k + "}" + " \\";**

**String nine = "\\ " + " or " + " \\" + "\\ x \\ = " + "\\dfrac {" + ed + " - " + df.format(m) + "i" + "} {" + k + "}" + " \\";**

**String ten = "\\ " + "Your \\ answers \\ are \\ imaginary \\";**

**String ele = "\\ " + "The \\ third \\ solution \\ of \\ x \\ is \\ zero \\ ";**

**TeXFormula eef = new TeXFormula(ball + one + add + two + twomid + twomidi + towo + tree + four +**

**bw + five + six + seven + mid + eight + nine + ten + ele);**

**/\* TeXIcon, to make the compiled string as a picture \*/**

**TeXIcon ti = eef.createTeXIcon(TeXConstants.DELIM\_SQUARE\_BRACKET, 19);**

**/\* Buffered Image to edit the Picture's features \*/**

**BufferedImage bud = new BufferedImage(ti.getIconWidth(), ti.getIconHeight(), BufferedImage.TYPE\_4BYTE\_ABGR);**

**/\* TeXICon code to modify the appearance of the picture, including the position \*/**

**ti.setInsets(new java.awt.Insets (-1, -1, 0, 0));**

**/\* TeXICon to add graphic quality to the picture \*/**

**ti.paintIcon(new JLabel(), bud.getGraphics(), 0, 0);**

**ti.setForeground(java.awt.Color.BLACK);**

**/\* SwingNode used to accommodate swing components in a JavaFX application \*/**

**SwingNode sn = new SwingNode();**

**/\* Swing TextPane to include the TeXICon produced \*/**

**JTextPane ht = new JTextPane(); /\* Initialization \*/**

**ht.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 0, java.awt.Color.DARK\_GRAY)); /\* Border settings between the Pane and the outside \*/**

**ht.insertIcon(ti); /\* The TextPane accommodates the Icon \*/**

**ht.setPreferredSize(new Dimension(550, 570)); /\* The Size of the TextPane \*/**

**ht.setForeground(java.awt.Color.BLUE); /\* Set the Text Color of the Icon \*/**

**ht.setBackground(java.awt.Color.WHITE); /\* Set the Background color of the TextPane \*/**

**ht.setVisible(true); /\* Make this visible, although it is always by default \*/**

**ht.setMargin(new java.awt.Insets(0,0,0,0)); /\* Set Margin between it and the Icon \*/**

**ht.setFocusable(true); /\* Remove the ability to focus \*/**

**ht.setEditable(false); /\* You cannot edit the items in the TextPane \*/**

**JPanel jp = new JPanel(); /\* The Framework for the TextPane \*/**

**jp.setLayout(new FlowLayout()); /\* The Layout of the Framework \*/**

**GridBagConstraints hi = new GridBagConstraints(); /\* The Constraints of the Layout \*/**

**hi.fill = GridBagConstraints.FIRST\_LINE\_START; /\* Positioning using the Constraints \*/**

**jp.add(ht, hi); /\* Apply the Constraints to the Layout and the Framework \*/**

**jp.setBackground(java.awt.Color.WHITE); /\* Background settings for the Framework \*/**

**sn.setContent(jp); /\* Add the Framework to the Node \*/**

**sn.setLayoutX(40); /\* Modify the positions of the Node \*/**

**sn.setLayoutY(300);**

**ScrollPane scp = new ScrollPane(sn); /\* Add the Node to the ScrollPane to make it Scrollable \*/**

**scp.setLayoutX(240); /\* Modify the Position of the ScrollPane \*/**

**scp.setLayoutY(200);**

**scp.setPrefWidth(570); /\* Set the width and height of the ScrollPane \*/**

**scp.setPrefHeight(370);**

**scp.setFitToWidth(true); /\* Make the Width of the Node the width of the Pane \*/**

**scp.setVisible(true); /\* Make the Pane visible \*/**

**scp.setStyle("-fx-background-color: white;");**

**gp.getChildren().add(scp); /\* Add everything you did to the Chief Layout \*/**

**}**

**else if((b\*b - 4 \* a \* c) < 0 && b >= 0 && de == 0) {**

**Double d = Double.parseDouble(f.getText());**

**Double e = Double.parseDouble(g.getText());**

**Double f = Double.parseDouble(h.getText());**

**Double i = e \* e;**

**Double j = 4 \* d \* f;**

**Double k = 2 \* d;**

**Double l = i - j;**

**Double m = Math.sqrt(Math.abs(l)); /\* Piece of Code for finding square-root \*/**

**Double ed = Math.abs(e);**

**DecimalFormat df = new DecimalFormat("#.###");**

**String ball = "\\ The \\ Formula \\ is: " ;**

**String one = "\\ " + "\\dfrac {-b \u00B1 \u221A(b^2 - 4ac)} {2a}" + " \\";**

**String add= "\\ " + "And \\ the \\ Equation \\ is: ";**

**String two = "\\ { " + d + "x^3 + \\ " + e + "x^2 + \\ " + f + "x \\ = \\ 0" + "} \\" ;**

**String twomid = "\\ Here, \\ a \\ = " + d + " \\ b \\ = " + e + "\\ and \\ c \\ = " + f + " \\";**

**String twomidi = "\\ " + "Factor \\ out \\ x \\ from \\ this \\ equation \\ to \\ make \\ it \\ quadratic \\";**

**String towo = "\\ { " + "x(" + d + "x^2 + \\ " + e + "x + \\ " + f + ") \\ = \\ 0" + "} \\";**

**String tree = "\\ " + "By \\ our \\ formula, \\ we \\ have ";**

**String four = "\\ " + "\\dfrac { -" + e + '\u00B1' + '\u221A' + "( " + e + "^2 - 4 \* " + d + " \* " + f + ")} {2 \*" + d + "}" + " \\" ;**

**String bw = "\\ " + "Simplify \\ Further \\ to \\ get" + " \\";**

**String five = "\\ " + "\\dfrac {-" + e + '\u00B1' + '\u221A' + "( " + i + " - " + j + ")} {" + k + "}" + " \\";**

**String six = "\\ " + "\\dfrac {-" + e + '\u00B1' + '\u221A' + "( " + l + ")} {" + k + "}" + " \\";**

**String seven = "\\ " + "\\dfrac {-" + e + '\u00B1' + "i \\ \* " + l + "} {" + k + "}" + " \\";**

**String mid = "\\ " + "Solve \\ differently \\ and \\ we \\ have" + " \\";**

**String eight = "\\ x \\ = " + "\\dfrac { \\ -" + ed + " + " + df.format(m) + "i" + "} {" + k + "}" + " \\";**

**String nine = "\\ " + " or " + " \\" + "\\ x \\ = " + "\\dfrac { \\ -" + ed + " - " + df.format(m) + "i" + "} {" + k + "}" + " \\";**

**String ten = "\\ " + " Your \\ answers \\ are \\ imaginary \\";**

**String ele = "\\ " + "The \\ third \\ solution \\ of \\ x \\ is \\ zero \\ ";**

**TeXFormula eef = new TeXFormula(ball + one + add + two + twomid + twomidi + towo + tree + four +**

**bw + five + six + seven + mid + eight + nine + ten + ele);**

**/\* TeXIcon, to make the compiled string as a picture \*/**

**TeXIcon ti = eef.createTeXIcon(TeXConstants.DELIM\_SQUARE\_BRACKET, 19);**

**/\* Buffered Image to edit the Picture's features \*/**

**BufferedImage bud = new BufferedImage(ti.getIconWidth(), ti.getIconHeight(), BufferedImage.TYPE\_4BYTE\_ABGR);**

**/\* TeXICon code to modify the appearance of the picture, including the position \*/**

**ti.setInsets(new java.awt.Insets (-1, -1, 0, 0));**

**/\* TeXICon to add graphic quality to the picture \*/**

**ti.paintIcon(new JLabel(), bud.getGraphics(), 0, 0);**

**ti.setForeground(java.awt.Color.BLACK);**

**/\* SwingNode used to accommodate swing components in a JavaFX application \*/**

**SwingNode sn = new SwingNode();**

**/\* Swing TextPane to include the TeXICon produced \*/**

**JTextPane ht = new JTextPane(); /\* Initialization \*/**

**ht.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 0, java.awt.Color.DARK\_GRAY)); /\* Border settings between the Pane and the outside \*/**

**ht.insertIcon(ti); /\* The TextPane accommodates the Icon \*/**

**ht.setPreferredSize(new Dimension(550, 570)); /\* The Size of the TextPane \*/**

**ht.setForeground(java.awt.Color.BLUE); /\* Set the Text Color of the Icon \*/**

**ht.setBackground(java.awt.Color.WHITE); /\* Set the Background color of the TextPane \*/**

**ht.setVisible(true); /\* Make this visible, although it is always by default \*/**

**ht.setMargin(new java.awt.Insets(0,0,0,0)); /\* Set Margin between it and the Icon \*/**

**ht.setFocusable(true); /\* Remove the ability to focus \*/**

**ht.setEditable(false); /\* You cannot edit the items in the TextPane \*/**

**JPanel jp = new JPanel(); /\* The Framework for the TextPane \*/**

**jp.setLayout(new FlowLayout()); /\* The Layout of the Framework \*/**

**GridBagConstraints hi = new GridBagConstraints(); /\* The Constraints of the Layout \*/**

**hi.fill = GridBagConstraints.FIRST\_LINE\_START; /\* Positioning using the Constraints \*/**

**jp.add(ht, hi); /\* Apply the Constraints to the Layout and the Framework \*/**

**jp.setBackground(java.awt.Color.WHITE); /\* Background settings for the Framework \*/**

**sn.setContent(jp); /\* Add the Framework to the Node \*/**

**sn.setLayoutX(40); /\* Modify the positions of the Node \*/**

**sn.setLayoutY(300);**

**ScrollPane scp = new ScrollPane(sn); /\* Add the Node to the ScrollPane to make it Scrollable \*/**

**scp.setLayoutX(240); /\* Modify the Position of the ScrollPane \*/**

**scp.setLayoutY(200);**

**scp.setPrefWidth(570); /\* Set the width and height of the ScrollPane \*/**

**scp.setPrefHeight(370);**

**scp.setFitToWidth(true); /\* Make the Width of the Node the width of the Pane \*/**

**scp.setVisible(true); /\* Make the Pane visible \*/**

**scp.setStyle("-fx-background-color: white;");**

**gp.getChildren().add(scp); /\* Add everything you did to the Chief Layout \*/**

**}**

**else if ((b\*b - 4 \* a \* c) >= 0) {**

**Double d = Double.parseDouble(f.getText());**

**Double e = Double.parseDouble(g.getText());**

**Double f = Double.parseDouble(h.getText());**

**Double i = e \* e;**

**Double j = 4 \* d \* f;**

**Double k = 2 \* d;**

**Double l = i - j;**

**Double m = Math.sqrt(Math.abs(l)); /\* Piece of Code for finding square-root \*/**

**Double n = (-e + m) / k;**

**Double o = (-e - m) / k;**

**DecimalFormat df = new DecimalFormat("#.###");**

**String ball = "\\ The \\ Formula \\ is: " ;**

**String one = "\\ " + "\\dfrac {-b \u00B1 \u221A(b^2 - 4ac)} {2a}" + " \\";**

**String add= "\\ " + "And \\ the \\ Equation \\ is: ";**

**String two = "\\ { " + d + "x^3 + \\ " + e + "x^2 + \\ " + f + "x \\ = \\ 0" + "} \\" ;**

**String twomid = "\\ Here, \\ a \\ = " + d + " \\ b \\ = " + e + "\\ and \\ c \\ = " + f + " \\";**

**String twomidi = "\\ " + "Factor \\ out \\ x \\ from \\ this \\ equation \\ to \\ make \\ it \\ quadratic \\";**

**String towo = "\\ { " + "x(" + d + "x^2 + \\ " + e + "x + \\ " + f + ") \\ = \\ 0" + "} \\";**

**String tree = "\\ " + "By \\ our \\ formula, \\ we \\ have ";**

**String four = "\\ " + "\\dfrac { -" + e + '\u00B1' + '\u221A' + "( " + e + "^2 - 4 \* " + d + " \* " + f + ")} {2 \*" + d + "}" + " \\" ;**

**String bw = "\\ " + "Simplify \\ Further \\ to \\ get" + " \\";**

**String five = "\\ " + "\\dfrac {-" + e + '\u00B1' + '\u221A' + "( " + i + " - " + j + ")} {" + k + "}" + " \\";**

**String six = "\\ " + "\\dfrac {-" + e + '\u00B1' + '\u221A' + "( " + l + ")} {" + k + "}" + " \\";**

**String seven = "\\ " + "\\dfrac {-" + e + '\u00B1' + m + "} {" + k + "}" + " \\";**

**String mid = "\\ " + "Solve \\ differently \\ and \\ we \\ have" + " \\";**

**String eight = "\\ " + "\\dfrac {-" + e + " + " + m + "} {" + k + "}" + " \\";**

**String nine = "\\ " + " or " + " \\" + "\\ " + "\\dfrac {-" + e + " - " + m + "} {" + k + "}" + " \\";**

**String ten = "\\ " + " and \\ x =" + df.format(n) + " \\";**

**String done = "\\ " + " or \\ " + df.format(o) + " \\";**

**String ele = "\\ " + "The \\ third \\ solution \\ of \\ x \\ is \\ zero \\ ";**

**TeXFormula eef = new TeXFormula(ball + one + add + two + twomid + twomidi + towo + tree + four +**

**bw + five + six + seven + mid + eight + nine + ten + done + ele);**

**/\* TeXIcon, to make the compiled string as a picture \*/**

**TeXIcon ti = eef.createTeXIcon(TeXConstants.DELIM\_SQUARE\_BRACKET, 19);**

**/\* Buffered Image to edit the Picture's features \*/**

**BufferedImage bud = new BufferedImage(ti.getIconWidth(), ti.getIconHeight(), BufferedImage.TYPE\_4BYTE\_ABGR);**

**/\* TeXICon code to modify the appearance of the picture, including the position \*/**

**ti.setInsets(new java.awt.Insets (-1, -1, 0, 0));**

**/\* TeXICon to add graphic quality to the picture \*/**

**ti.paintIcon(new JLabel(), bud.getGraphics(), 0, 0);**

**ti.setForeground(java.awt.Color.black);**

**/\* SwingNode used to accommodate swing components in a JavaFX application \*/**

**SwingNode sn = new SwingNode();**

**/\* Swing TextPane to include the TeXICon produced \*/**

**JTextPane ht = new JTextPane(); /\* Initialization \*/**

**ht.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 0, java.awt.Color.DARK\_GRAY)); /\* Border settings between the Pane and the outside \*/**

**ht.insertIcon(ti); /\* The TextPane accommodates the Icon \*/**

**ht.setPreferredSize(new Dimension(550, 570)); /\* The Size of the TextPane \*/**

**ht.setForeground(java.awt.Color.BLUE); /\* Set the Text Color of the Icon \*/**

**ht.setBackground(java.awt.Color.white); /\* Set the Background color of the TextPane \*/**

**ht.setVisible(true); /\* Make this visible, although it is always by default \*/**

**ht.setMargin(new java.awt.Insets(0,0,0,0)); /\* Set Margin between it and the Icon \*/**

**ht.setFocusable(true); /\* Remove the ability to focus \*/**

**ht.setEditable(false); /\* You cannot edit the items in the TextPane \*/**

**JPanel jp = new JPanel(); /\* The Framework for the TextPane \*/**

**jp.setLayout(new FlowLayout()); /\* The Layout of the Framework \*/**

**GridBagConstraints hi = new GridBagConstraints(); /\* The Constraints of the Layout \*/**

**hi.fill = GridBagConstraints.FIRST\_LINE\_START; /\* Positioning using the Constraints \*/**

**jp.add(ht, hi); /\* Apply the Constraints to the Layout and the Framework \*/**

**jp.setBackground(java.awt.Color.WHITE); /\* Background settings for the Framework \*/**

**sn.setContent(jp); /\* Add the Framework to the Node \*/**

**sn.setLayoutX(40); /\* Modify the positions of the Node \*/**

**sn.setLayoutY(300);**

**ScrollPane scp = new ScrollPane(sn); /\* Add the Node to the ScrollPane to make it Scrollable \*/**

**scp.setLayoutX(240); /\* Modify the Position of the ScrollPane \*/**

**scp.setLayoutY(200);**

**scp.setPrefWidth(570); /\* Set the width and height of the ScrollPane \*/**

**scp.setPrefHeight(370);**

**scp.setFitToWidth(true); /\* Make the Width of the Node the width of the Pane \*/**

**scp.setVisible(true); /\* Make the Pane visible \*/**

**scp.setStyle("-fx-background-color: white;");**

**gp.getChildren().add(scp); /\* Add everything you did to the Chief Layout \*/**

**}**

**}**

**});**

**elim.setOnAction(new EventHandler <ActionEvent>() {**

**public void handle(ActionEvent eg) {**

**final Double a = Double.parseDouble(f.getText());**

**final Double b = Double.parseDouble(g.getText());**

**final Double c = Double.parseDouble(h.getText());**

**final Double de = Double.parseDouble(gh.getText());**

**final Double ace = (c/a) - (Math.pow(b, 2) / (3 \* (Math.pow(a, 2) ) ) );**

**final Double brace = (de/a) - (2 \* (Math.pow(b, 3))) / (27 \* (Math.pow(a, 3))) - ( (b \* c) / (3 \* (Math.pow(a, 2))));**

**final Double A = Math.pow(ace, 3);**

**final Double B = Math.pow(brace, 2);**

**final Double trace = (-brace/2) + Math.sqrt( ( (brace\*brace) / 4 ) + ( (ace\*ace\*ace) / 27 ) );**

**final Double grace = (brace/2) + Math.sqrt( ( (brace\*brace) / 4 ) + ( (ace\*ace\*ace) / 27 ) );**

**final Double rax = Math.cbrt(trace) - Math.cbrt(grace) - (b/(3\*a));**

**DecimalFormat df = new DecimalFormat("#.###");**

**final Double tin = brace / 2;**

**String ball = "\\ This \\ is \\ a \\ cubic \\ equation: \\" ;**

**String ball1 = "\\ " + "\\ To \\ be \\ solved \\ using \\ the \\ cubic \\ formula \\";**

**String ball2 = "\\ " + "\\ We \\ are \\ to \\ arrive \\ at \\ one \\ root \\ of \\ the \\ equation \\";**

**String ball3 = "\\ " + "\\ The \\ cubic \\ formula \\ is; \\";**

**String ball4 = "\\ x \\ = \\ ^3 \u221A( \\dfrac {-B} {2} + \u221A (\\dfrac {B^2} {4} + \\dfrac{A^3} {27}) ) ";**

**String ball5 = "\\ - \\ ^3 \u221A( \\dfrac {B} {2} + \u221A (\\dfrac {B^2} {4} + \\dfrac{A^3} {27}) ) ";**

**String ball6 = "\\ " + "\\ - \\dfrac {b} {3a} \\";**

**String ball7 = "\\ Where \\ A \\ = \\dfrac{c} {a} - \\dfrac{b^2} {3a^2} \\ and \\";**

**String ball8 = "\\ B \\ = \\dfrac{d} {a} - \\dfrac{2b^3} {27a^3} - \\dfrac{bc} {3a^2} \\";**

**String ball9 = "\\ The \\ values \\ for \\ this \\ equation \\ are; \\";**

**String ball10 = "\\ a \\ = " + a + "\\ b \\ = " + b + "\\ c \\ = " + c + "\\ and \\ d \\ = " + de + "\\";**

**String ball11 = "\\ To \\ begin, \\ we \\ calculate \\ A \\ to \\ be: \\";**

**String ball12 = "\\ " + "\\dfrac{" + c + "} {" + a + "} - \\dfrac{" + b + "^2} {3 \* " + a + "^2} = " + df.format(ace) + "\\";**

**String ball13 = "\\ Calculate \\ the \\ value \\ of \\ B \\ to \\ get: \\";**

**String ball14 = "\\" + "\\dfrac{" + de + "} {" + a + "} - \\dfrac{2 \* " + b + "^3} {27 \* " + a + "^3} - \\dfrac{" + b + " \* " + c + "} {3 \* " + a + "^2} = " + df.format(brace) + "\\";**

**String ball15 = "\\ Having \\ got \\ these, \\ we \\ insert \\ into \\ the \\ cubic \\ equation \\ to \\ get: \\";**

**String ball16 = "\\ x \\ = \\ ^3 \u221A( \\dfrac {-" + brace + "} {2} + \u221A (\\dfrac {" + brace + "^2} {4} + \\dfrac{" + ace + "^3} {27}) ) \\";**

**String ball17 = "\\ - \\ ^3 \u221A( \\dfrac {" + brace + "} {2} + \u221A (\\dfrac {" + brace + "^2} {4} + \\dfrac{" + ace + "^3} {27}) ) \\";**

**String ball18 = "\\ - \\dfrac {" + b + "} {3 \* " + a + "} \\";**

**String ball19 = "\\ And \\ our \\ x \\ = " + df.format(rax) + "\\";**

**TeXFormula eef = new TeXFormula(ball + ball1 + ball2 + ball3 + ball4 + ball5 +**

**ball6 + ball7 + ball8 + ball9 + ball10 + ball11 + ball12 + ball13 + ball14 +**

**ball15 + ball16 + ball17 + ball18 + ball19);**

**/\* TeXIcon, to make the compiled string as a picture \*/**

**TeXIcon ti = eef.createTeXIcon(TeXConstants.DELIM\_BRACKET, 18);**

**/\* Buffered Image to edit the Picture's features \*/**

**BufferedImage bud = new BufferedImage(ti.getIconWidth(), ti.getIconHeight(), BufferedImage.TYPE\_4BYTE\_ABGR);**

**/\* TeXICon code to modify the appearance of the picture, including the position \*/**

**ti.setInsets(new java.awt.Insets (-1, -1, 0, 0));**

**/\* TeXICon to add graphic quality to the picture \*/**

**ti.paintIcon(new JLabel(), bud.getGraphics(), 0, 0);**

**ti.setForeground(java.awt.Color.black);**

**/\* SwingNode used to accommodate swing components in a JavaFX application \*/**

**SwingNode sn = new SwingNode();**

**/\* Swing TextPane to include the TeXICon produced \*/**

**JTextPane ht = new JTextPane(); /\* Initialization \*/**

**ht.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 0, java.awt.Color.DARK\_GRAY)); /\* Border settings between the Pane and the outside \*/**

**ht.insertIcon(ti); /\* The TextPane accommodates the Icon \*/**

**ht.setPreferredSize(new Dimension(670, 670)); /\* The Size of the TextPane \*/**

**ht.setForeground(java.awt.Color.BLUE); /\* Set the Text Color of the Icon \*/**

**ht.setBackground(java.awt.Color.white); /\* Set the Background color of the TextPane \*/**

**ht.setVisible(true); /\* Make this visible, although it is always by default \*/**

**ht.setMargin(new java.awt.Insets(-1,-1,0,0)); /\* Set Margin between it and the Icon \*/**

**ht.setFocusable(true); /\* Remove the ability to focus \*/**

**ht.setEditable(false); /\* You cannot edit the items in the TextPane \*/**

**JPanel jp = new JPanel(); /\* The Framework for the TextPane \*/**

**jp.setLayout(new FlowLayout()); /\* The Layout of the Framework \*/**

**GridBagConstraints hi = new GridBagConstraints(); /\* The Constraints of the Layout \*/**

**hi.fill = GridBagConstraints.FIRST\_LINE\_START; /\* Positioning using the Constraints \*/**

**jp.add(ht, hi); /\* Apply the Constraints to the Layout and the Framework \*/**

**jp.setBackground(java.awt.Color.WHITE); /\* Background settings for the Framework \*/**

**sn.setContent(jp); /\* Add the Framework to the Node \*/**

**sn.setLayoutX(40); /\* Modify the positions of the Node \*/**

**sn.setLayoutY(300);**

**ScrollPane scp = new ScrollPane(sn); /\* Add the Node to the ScrollPane to make it Scrollable \*/**

**scp.setLayoutX(240); /\* Modify the Position of the ScrollPane \*/**

**scp.setLayoutY(220);**

**scp.setPrefWidth(700); /\* Set the width and height of the ScrollPane \*/**

**scp.setPrefHeight(480);**

**scp.setFitToWidth(true); /\* Make the Width of the Node the width of the Pane \*/**

**scp.setVisible(true); /\* Make the Pane visible \*/**

**scp.setStyle("-fx-background-color: white;");**

**gp.getChildren().add(scp); /\* Add everything you did to the Chief Layout \*/**

**}**

**});**

**graph.setOnAction(new EventHandler<ActionEvent>(){**

**@SuppressWarnings({ "rawtypes", "unchecked" })**

**public void handle(ActionEvent ev) {**

**final Stage sag = new Stage();**

**sag.setTitle("Graph Co-ordinates");**

**Group gd = new Group();**

**Scene sc = new Scene(gd, 300, 300);**

**sag.setScene(sc);**

**sag.setResizable(false);**

**sag.show();**

**Label ol = new Label ("Enter the boundaries");**

**ol.setLayoutX(50);**

**ol.setLayoutY(20);**

**ol.setFont(Font.font("Times New Roman", FontWeight.BOLD, 17));**

**final TextField a = new TextField();**

**a.setTooltip(new Tooltip("Lower Bound"));**

**a.setLayoutX(50);**

**a.setPrefWidth(70);**

**a.setLayoutY(80);**

**final TextField mb = new TextField();**

**mb.setLayoutX(50);**

**mb.setLayoutY(140);**

**mb.setPrefWidth(70);**

**mb.setTooltip(new Tooltip("Sign (+/-)"));**

**final TextField b = new TextField();**

**b.setTooltip(new Tooltip("Upper Bound"));**

**b.setLayoutX(50);**

**b.setLayoutY(200);**

**b.setPrefWidth(70);**

**Button plot = new Button("Plot Chart");**

**plot.setLayoutX(50);**

**plot.setLayoutY(260);**

**plot.setFont(Font.font("Cambria", FontWeight.BOLD, 16));**

**gd.getChildren().addAll(ol, a, mb, b, plot);**

**plot.setOnAction(new EventHandler<ActionEvent>(){**

**public void handle (ActionEvent th) {**

**if(mb.getText().equals("+")) {**

**sag.close();**

**CategoryAxis xaxis = new CategoryAxis();**

**xaxis.setLabel("X-axis");**

**NumberAxis yaxis = new NumberAxis();**

**yaxis.setLabel("Y-axis");**

**LineChart lct = new LineChart(xaxis, yaxis);**

**lct.setData(getData());**

**lct.setStyle("-fx-background-color: white;");**

**StackPane root = new StackPane();**

**root.getChildren().add(lct);**

**root.setLayoutX(320);**

**root.setLayoutY(300);**

**root.setPrefSize(570, 370);**

**gp.getChildren().add(root);**

**}**

**else if (mb.getText().equals("-")) {**

**sag.close();**

**CategoryAxis xaxis = new CategoryAxis();**

**xaxis.setLabel("X-axis");**

**NumberAxis yaxis = new NumberAxis();**

**yaxis.setLabel("Y-axis");**

**LineChart lct = new LineChart(xaxis, yaxis);**

**lct.setData(postData());**

**lct.setStyle("-fx-background-color: white;");**

**StackPane root = new StackPane();**

**root.getChildren().add(lct);**

**root.setLayoutX(320);**

**root.setLayoutY(300);**

**root.setPrefHeight(570);**

**root.setPrefWidth(370);**

**gp.getChildren().add(root);**

**}**

**}**

**@SuppressWarnings({ "unchecked", "rawtypes" })**

**private ObservableList<XYChart.Series<String, Double>> getData() {**

**ObservableList<XYChart.Series<String, Double>> data = FXCollections.observableArrayList();**

**Series<String, Double> squad = new Series<String, Double>();**

**double ant = Double.parseDouble(f.getText());**

**double brant = Double.parseDouble(g.getText());**

**double crant = Double.parseDouble(h.getText());**

**double drant = Double.parseDouble(gh.getText());**

**double ace = Double.parseDouble(a.getText());**

**double aid = Double.parseDouble(b.getText());**

**squad.setName("Graphical Solution for " + f.getText() + "x\u00B2 + " + g.getText() + "x + " + h.getText() + " = 0");**

**for (double i = ace; i <= aid; i+=0.5) {**

**double squadValue = ((ant \* (i \* i \* i)) + (brant \* i \* i) + (crant \* i) + drant);**

**squad.getData().add(new XYChart.Data(Double.toString(i), squadValue));**

**}**

**data.add(squad);**

**return data;**

**}**

**@SuppressWarnings({ "unchecked", "rawtypes" })**

**private ObservableList<XYChart.Series<String, Double>> postData() {**

**ObservableList<XYChart.Series<String, Double>> data = FXCollections.observableArrayList();**

**Series<String, Double> squad = new Series<String, Double>();**

**double ant = Double.parseDouble(f.getText());**

**double brant = Double.parseDouble(g.getText());**

**double crant = Double.parseDouble(h.getText());**

**double drant = Double.parseDouble(gh.getText());**

**double ace = Double.parseDouble(a.getText());**

**double aid = Double.parseDouble(b.getText());**

**squad.setName("Graphical Solution for " + f.getText() + "x\u00B2 + " + g.getText() + "x + " + h.getText() + " = 0");**

**for (double i = ace ; i >= aid; i-=0.5) {**

**double squadValue = ((ant \* (i \* i \* i)) + (brant \* (i \* i)) + (crant \* i) + drant);**

**squad.getData().add(new XYChart.Data(Double.toString(i), squadValue));**

**}**

**data.add(squad);**

**return data;**

**}**

**});**

**}**

**});**

**gp.getChildren().addAll(info, f1, f, g1, g, h1, h, h2, gh, vb, mb);**

**stg.setScene(sc);**

**stg.show();**

**}**

**}**