

object Oreinted Analysis and Design

Project Report



Made By:

Mohammed Ehab Elsaeed 16P8160

Ahmed Sameh Shahin 16P6063

Youssef Assem Mohammed 16P6064

To Be Submitted To:

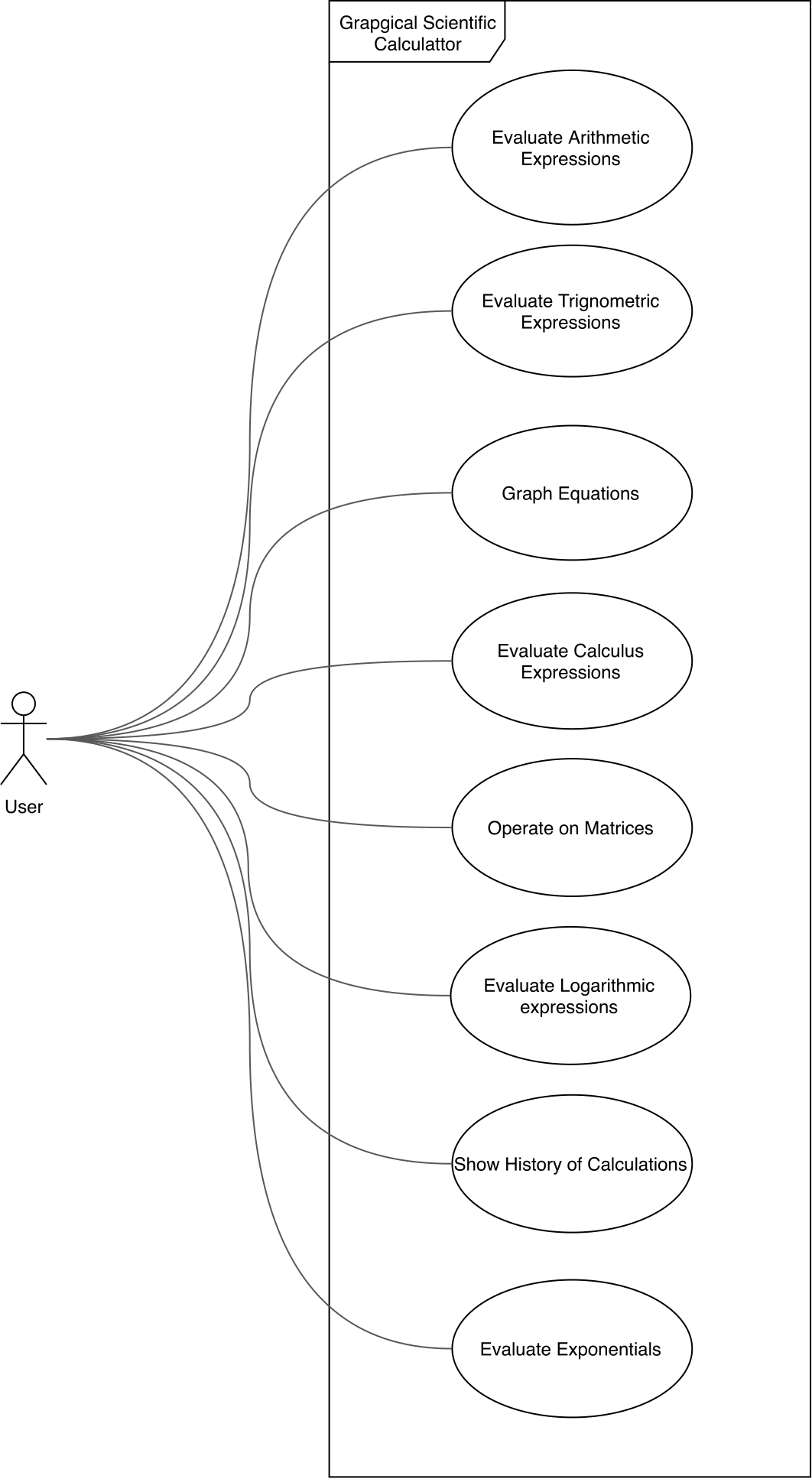
Dr. Islam El Maddah

Eng. Sarah Abdelazim

# Use cases Used

1. Graph Equations
2. Operate on Matrices
3. Evaluate Arithmetic Expressions
4. Evaluate Logarithmic Expressions
5. Evaluate Exponential Expressions
6. Evaluate Trigonometric Expressions
7. Evaluate Calculus Expressions
8. Show History of calculations

# Use Case Diagram



Use Case Descriptions

**Evaluate Arithmetic Expressions**

Use Case Name: Evaluate arithmetic expression

Goal in context: Add, subtract, multiply or divide two numbers

Pre-Conditions: User have entered the arithmetic expression

Successful end conditions: User gets the result

Failure End Conditions: User does not get the result

Primary Actors: User

**Secondary actors: ---**

Trigger: User presses on the equals button

Main Flow:

1. User chooses arithmetic section form main menu
2. User enters the first operand
3. User enters the operator
4. User enters the second operand
5. User presses the equals button
6. System verifies that it is a valid expression
7. System shows the result

Extensions:

6.1. Expression is invalid

**Evaluate Trigonometric Expressions**

Use Case Name: Evaluate trigonometric function

Goal in context: Get the value of trigonometric function

Pre-Conditions: User has chosen the trigonometric function

Successful end conditions: User gets the result

Failure End Conditions: User does not get the result

Primary Actors: User

**Secondary actors: ---**

Trigger: User presses on the equals button

Main Flow:

1. User chooses arithmetic section form main menu
2. User chooses the trigonometric function he wants
3. User enters the angle in radians
4. User presses the Done button
5. User presses the equals button
6. System evaluates the trigonometric function
7. System shows the result

**Graph Equations**

Use Case Name: Graph Equation

Goal in context: Draw the equation

Pre-Conditions: User enters an equation and limits of the graph

Successful end conditions: User gets graph

Failure End Conditions: User does not get the graph

Primary Actors: User

**Secondary actors: ---**

Trigger: User presses on the Graph button

Main Flow:

1. User chooses graph section form main menu
2. User writes the equation of the graph
3. User enters limits of the graph
4. User presses the graph button
5. System evaluates the draws the equation

Extensions:

2.1. User enters invalid equation

3.1 User enters invalid limit

**Evaluate Calculus Expressions**

Use Case Name: Evaluate calculus expression

Goal in context: get the numeric value of the expression

Pre-Conditions: User enters an equation and limits/point to calculate at

Successful end conditions: User gets the numeric value

Failure End Conditions: User does not get the numeric value

Primary Actors: User

**Secondary actors: ---**

Trigger: User presses on the Integrate/Differentiate Button

Main Flow:

1. User chooses calculus section form main menu
2. User chooses whether to differentiate or integrate
3. User writes the equation
4. User enters limits/point to calculate at
5. User presses on the Integrate/Differentiate Button
6. System evaluates result

Extensions:

3.1 User enters invalid equation

4.1 User enters invalid limits/point

**Operate on Matrices**

Use Case Name: Operate on matrices

Goal in context: Get the resultant matrix form an add, subtract or multiply operation

Pre-Conditions: User enters two matrices

Successful end conditions: User gets the resultant matrix

Failure End Conditions: User does not get the resultant matrix

Primary Actors: User

**Secondary actors: ---**

Trigger: User presses on the needed-operation Button

Main Flow:

1. User chooses matrix section form main menu
2. User enter number of rows and columns
3. User press “Create” button
4. User enters the first matrix
5. User press “Save” button
6. User repeat the above steps with the second matrix
7. User choose the type of operations
8. System verify if the operation is valid
9. User gets the resultant matrix

Extensions:

6.1 User enters invalid combination for any operation

**Evaluate Logarithmic Expressions**

Use Case Name: Evaluate logarithmic expression

Goal in context: Apply logarithmic operations on a number

Pre-Conditions: User have entered the logarithmic expression

Successful end conditions: User gets the result

Failure End Conditions: User does not get the result

Primary Actors: User

**Secondary actors: ---**

Trigger: User presses on the equals button

Main Flow:

1. User chooses arithmetic section form main menu
2. User chooses the Logarithmic function he wants
3. User enters the number to be entered in his chosen function
4. User presses the Done button
5. User presses the equals button
6. System evaluates the logarithmic function
7. System shows the result

Extensions:

6.1. Expression is invalid

**Show History of calculations**

Use Case Name: Show history of calculations

Goal in context: Save user calculations

Pre-Conditions: User have made a calculation

Successful end conditions: calculation saved in history

Failure End Conditions: calculation not saved in history

Primary Actors: User

**Secondary actors: ---**

Trigger: User presses on the equals button

Main Flow:

1. User does a calculation
2. User presses show history
3. System shows history list

**Evaluate Exponential Expressions**

Use Case Name: Evaluate exponential

Goal in context: Apply exponential operations on a number

Pre-Conditions: User have entered the exponential expression

Successful end conditions: User gets the result

Failure End Conditions: User does not get the result

Primary Actors: User

**Secondary actors: ---**

Trigger: User presses on the equals button

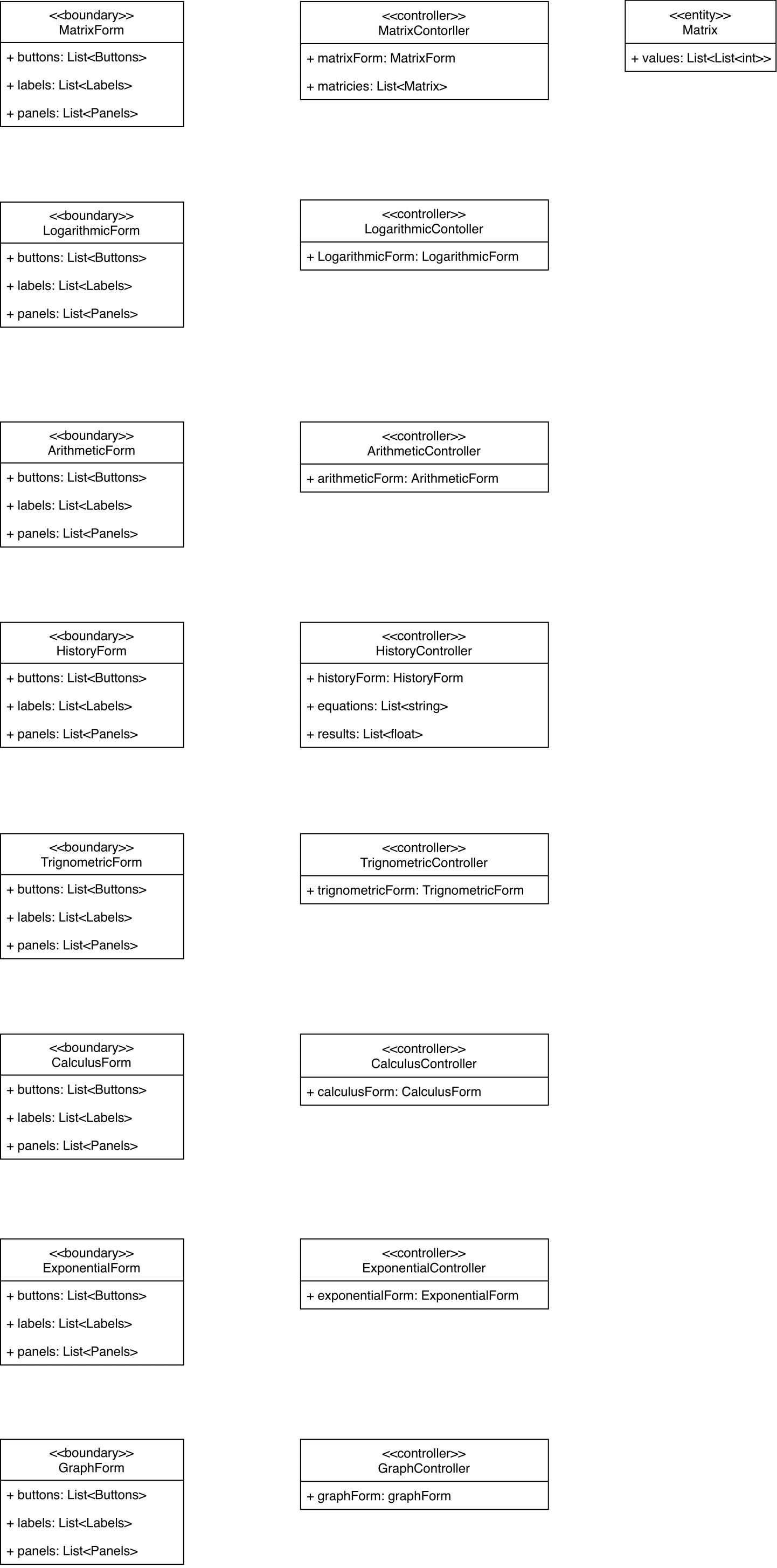
Main Flow:

1. User chooses arithmetic section form main menu
2. User chooses the exponential function he wants
3. User enters the number to be entered in his chosen function
4. User presses the Done button
5. User presses the equals button
6. System evaluates the logarithmic function
7. System shows the result

Extensions:

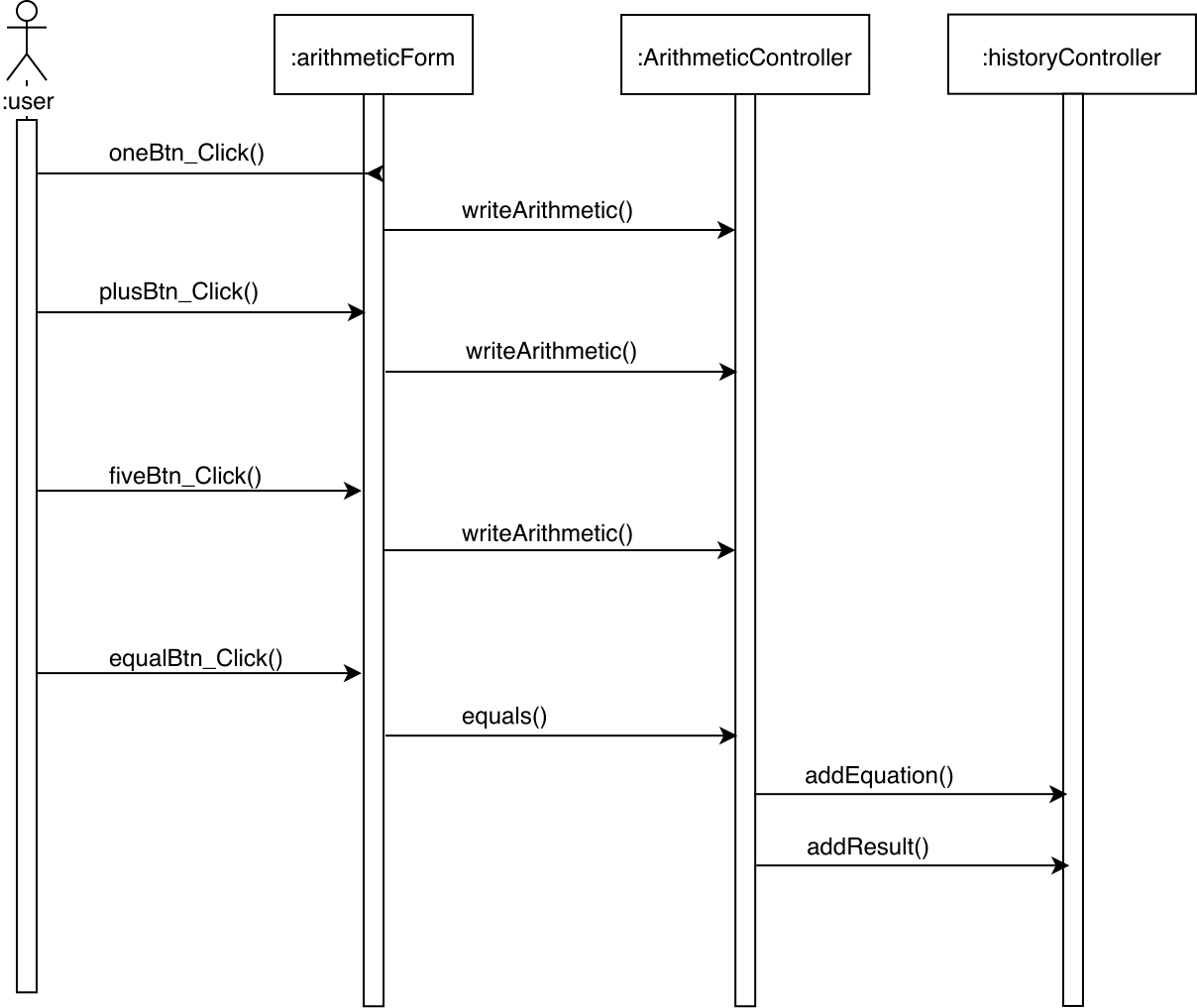
6.1. Expression is invalid

# Analysis Classes

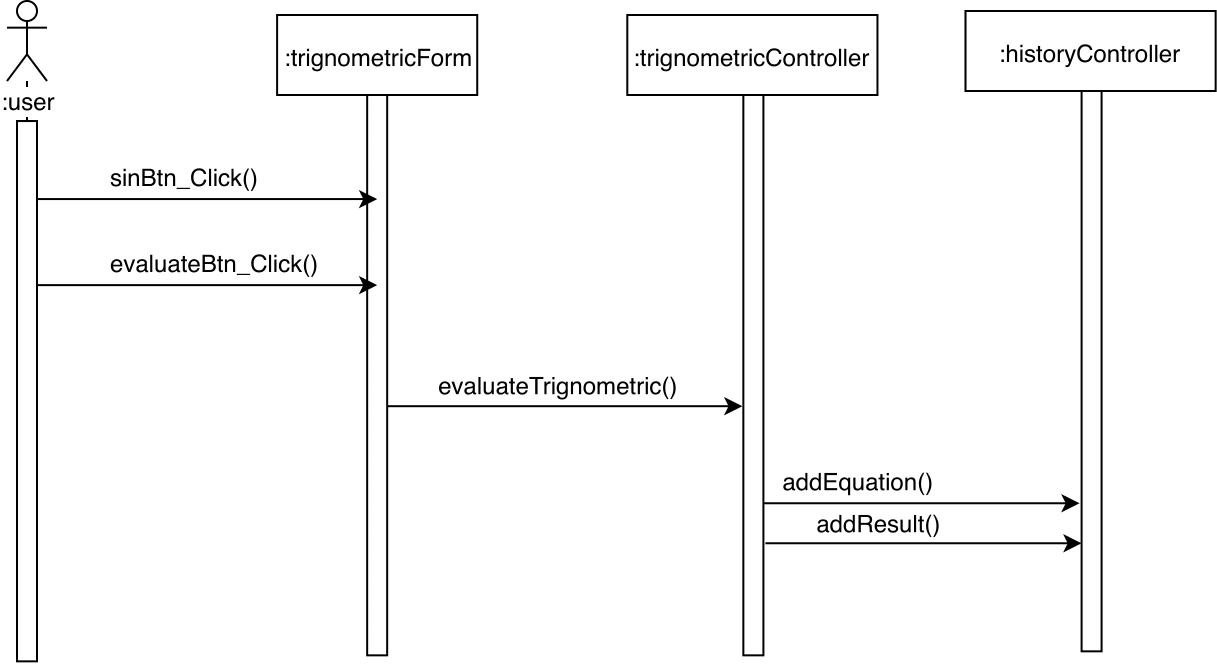


Sequence Diagrams

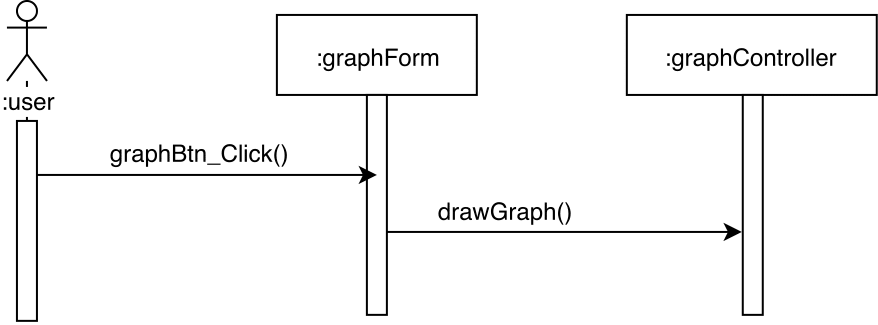
## Evaluate Arithmetic Expression



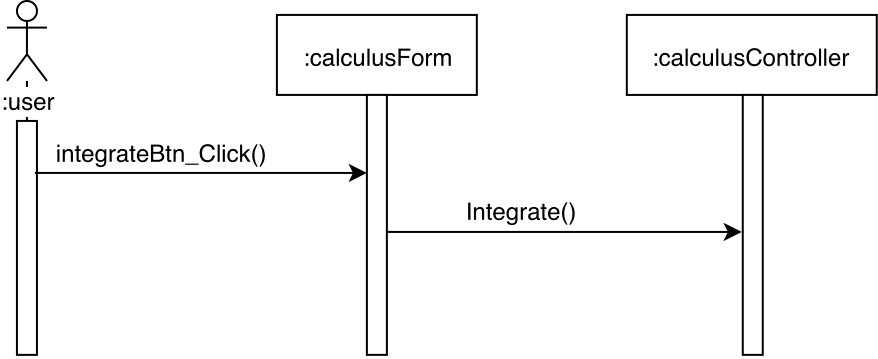
## Evaluate Trigonometric Expression



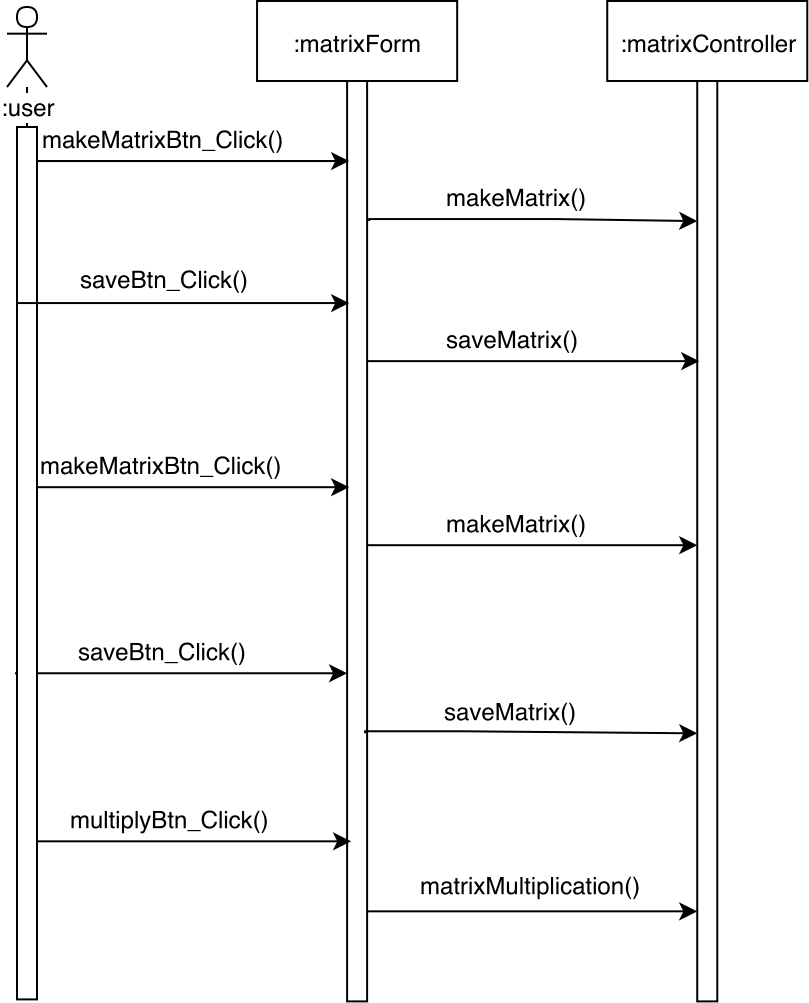
## Graph Equations



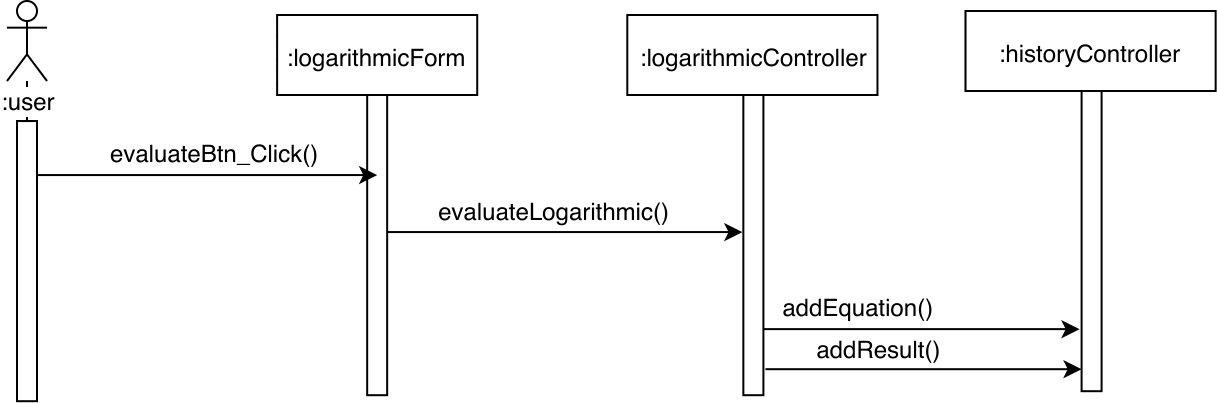
## Evaluate Calculus Expressions



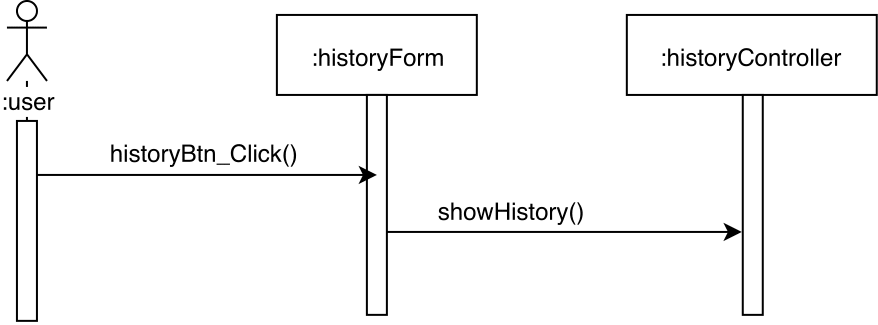
## Operate on Matrices



## Evaluate logarithmic Expressions



## Show History of Calculations



## Evaluate Exponentials

