**User Manual**

*Online UML Class Diagram Drawing Tool*

**CS5351/1920A - Group 3**

**Revision Sheet**

|  |  |  |
| --- | --- | --- |
| **Release No.** | **Date** | **Revision Description** |
| 0.1 | 19/11/2020 | First Draft |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**USER'S MANUAL**

**TABLE OF CONTENTS**

Page #

1. GENERAL INFORMATION 1-1

1.1 System Overview 1-1

2. SYSTEM SUMMARY 2-1

2.1 System Configuration 2-1

2.2 Data Flows 2-1

3. USING the SYSTEM 3-1

3.1 UML Diagram Generator 3-1

3.1.1 Input Text Script 3-1

3.1.2 Generate Class Diagram 3-2

3.1.3 Display Error Message 3-2

3.1.4 Reset Input Text 3-2

3.2 Diagram Viewer 3-2

3.1.1 Zoom In / Out Diagram 3-2

3.1.2 Move Diagram 3-2

3.1.4 Export Diagram (Pending) 3-2

4. Appendix 4-1

4. 1 Screen Shot 4-1

4.1.1 Input Text Script 4-2

4.1.2 Display Error Message 4-3

4.1.3 Display Class Diagram 4-3

4.1.4 Viewer Function 4-3

# GENERAL INFORMATION

## 1.1 System Overview

Online UML Class Diagram Drawing Tool:

1. A web-based tool written in HTML, CSS and JavaScript.
2. Generate Class Diagram based on the input text captured.
3. System category:
4. *Text Parsing:* parses and validates the input text captured and constructs object(s)
5. *Diagram Visualization:* generates diagram bases on the object(s) created by text parser

# SYSTEM SUMMARY

## 2.1 System Configuration

The online drawing tool is a client-scripting program (HTML, CSS, JavaScript). It can run on any web browser.

## 2.2 Data Flows

Users input text in the designated textbox on the html page. On every key press, the keypress event will be fired. The input text will be sent to the text parser module. The parser will parse and validate the input text line by line. Object(s) will be constructed. The created object(s) will be passed to the drawer module. The drawer will use a tree structure to layout the graph and display the resultant Class Diagram.

# USING the SYSTEM

## 3.1 UML Diagram Generator

### 3.1.1 Input Text Script

1. The textbox accepts input text of multiple lines.
2. The first line is the definition of an object.  
   E.g. Class A or Interface B
3. The object can either inherit a class or implement an interface. The system uses “>>” to define inheritance and “||” to define association. A class can have 0 or 1 superclass and it can implement 0 to N interface(s). An interface can implement 0 to 1 interface.  
   E.g. Class A >> B or Class A || B || C or Interface A || B
4. The second lines onwards define (a) the attributes and methods of a class, or (b) the methods of an interface.
5. If the line defines an attribute, the text format is something like “<m><attribute\_name><:><type>”.

<m> = modifier of the attribute. Values include:

|  |  |
| --- | --- |
| + | public |
| - | private |
| # | protected |
| ~ | Package |

<attribute\_name> = attribute name

<:> = separator

<type> = attribute type

E.g. +attr1:int

1. If the line defines a method, the text format is something like   
   “<m><method\_name (mtype0 mattr0, mtype1 mattr1, …. ,mtypeN mattrN)><:><type>”.

<m> = modifier of the attribute. Values include:

|  |  |
| --- | --- |
| + | public |
| - | private |
| # | protected |
| ~ | Package |

<method\_name(mtype0 mattr0, mtype1 mattr1, …. ,mtypeN mattrN)> = method name and parameters definition. A method can have 0 to N method parameters.

<:> = separator

<type> = return type  
E.g. +methodA(int param1, double param2):void

1. The textbox accepts definition of multiple objects. If an object inherits a class or implements an interface, the referenced object must have already been defined in previous lines. Object name cannot be reused in the input text, in other words, duplication is not allowed.

### 3.1.2 Generate Class Diagram

1. A tree structure graph is generated according to the input text.

### 3.1.3 Display Error Message

1. When an error is found for an object during validation, an error message will be displayed in the error message box. Error messages for multiple objects will be displayed in the same error message box.

### 3.1.4 Reset Input Text

1. The user can click  to clear the input text.

## 3.2 Diagram Viewer

### 3.1.1 Zoom In / Out Diagram

1. After a diagram is generated, the user can click  or  to zoom in or zoom out to view the diagram.

### 3.1.2 Move Diagram

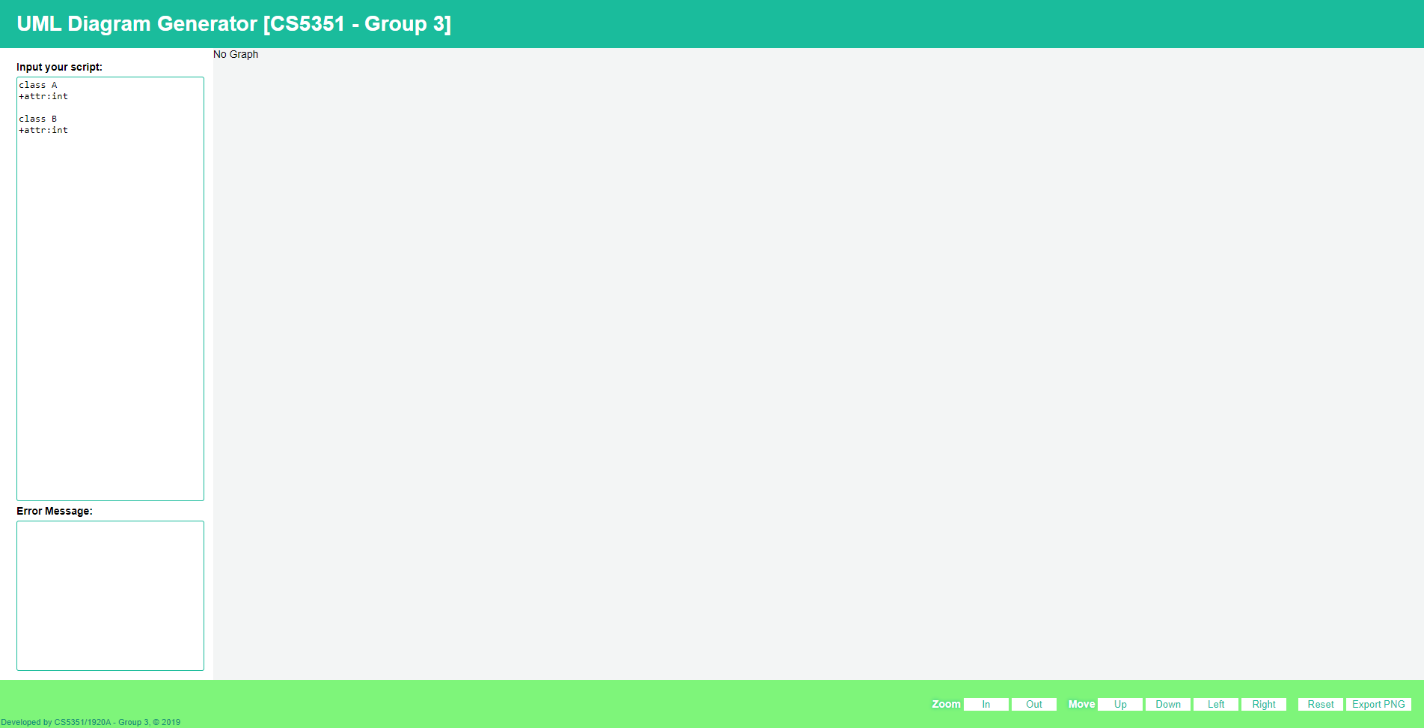
1. After a diagram is generated, the user can click  to position the diagram on display area.

### 3.1.4 Export Diagram (Pending)

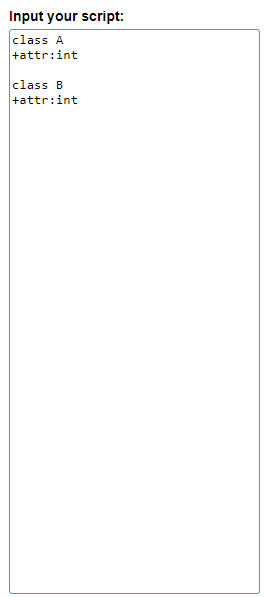
1. After a diagram is generated, the user can click  export the diagram to an image.

# Appendix

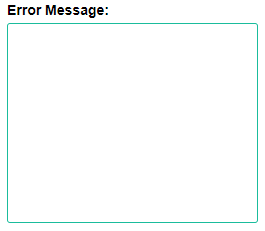
## 4. 1 Screen Shot

**

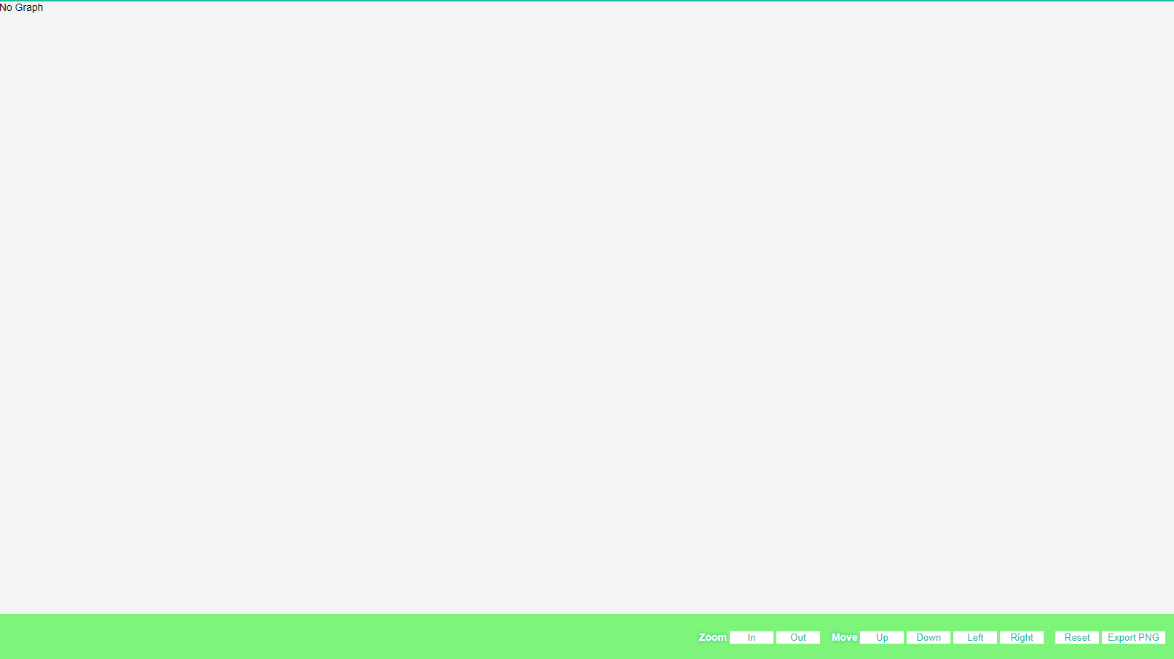
### 4.1.1 Input Text Script



### 4.1.2 Display Error Message



### 4.1.3 Display Class Diagram



### 4.1.4 Viewer Function

