Automatic Evaluation for Engineering Drawing

by

Luthfan Areka Adi

15209

Dissertation Draft Submitted in Partial Fulfilment of the Requirement for the

Bachelor of Technology (Hons) in Information & Communication Technology

December 2013

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

SAMPLE OF CERTIFICATION OF ORIGINALITY

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

____

LUTHFAN AREKA ADI

ABSTRACT

Engineering Drawing is the way engineers visualize, describe and communicate their design of an object in form of drawings. Engineers use Engineering Drawing technique in order to illustrate the requirement of their engineered items. Usually engineers use the drawings to give a clear information and specification of the object to the people who will make the object.

Consider as one of the most important knowledge for engineers, many university which offer engineering course put Engineering Drawing as one of their main course. Usually most of universities offer this course in the beginning of engineering student's study year because Engineering Drawing is the foundation knowledge for engineering student and students have to complete it as well in order to get the basic knowledge of engineering-related field. So there will be many numbers of students from any engineering programs enrolling this course per semester. As an overview the number of lecturer and tutor who taught Engineering Drawing in universities is not enough to accommodate such a big number of students.

The big difference number of student and lecturer will cause a problem when it comes to evaluating and mark each of student works since the only way to evaluate the drawings are still manual fashioned by printing out the drawing. Previous study has shown the capability of having an automatic system although it still at the beginning state. The system is able to read the DXF file from the basic shape drawings and give a mark only on specific shape although it is just a dummy value. Therefore, the objective of this study is continuing the possibility of having automatic evaluation and a proper marking scheme for the drawings.

The investigation of this study will cover the conversion of a DXF file to SVG file from Engineering Drawing 2D drawings format using AutoCAD. The last part is giving the proper marking scheme according to the Engineering Drawing rules and practice from the lecturer.

Finally, this study is expected to expand the technique of reading from various drawings scenarios. The reading technique of the result from the conversion will be the important

part of this study as this is the way to compare between lecturer and student works and determine the different point. The research for marking scheme is ongoing although still doesn’t find the precise method.

Contents

CHAPTER 1

1. PROJECT BACKGROUND

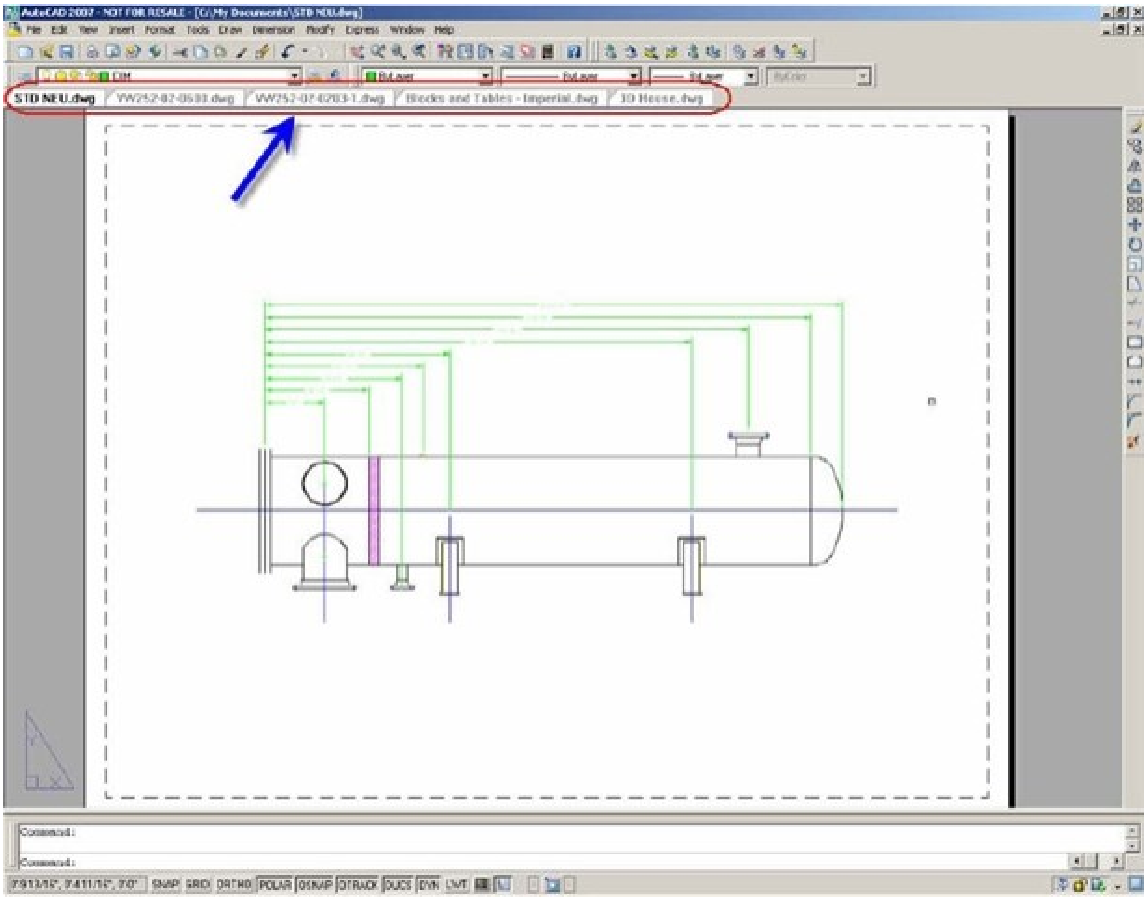
1.1 BACKGROUND STUDY

Engineering drawing is one of the types from technical drawing to explain, illustrate and represent the requirement for engineered items. It is also a language that communicates ideas and information from one to another mind. It communicates the needed information for engineers [2]. Like a programming language, this drawing is a graphical language used by engineers and technical personnel that have associated in engineering profession. Engineering drawing will represent the projection or illustration of an item that engineers work on it such as machine parts, floor plan for building, assembly part and many other types of engineering objects. The projection may be in a form of two-dimensional or in three-dimensional drawings.

Engineering drawings usually include such features as various types of lines, dimensions, section view and symbols. The object is consists from many basic shape where they combined together into one unity object. Many objects in the drawing requires a clear view from all the side in order to make engineers easy to determine which section they will work onto.

Layout drawings of different types are used in different fields for various purposes. One is the plant layout drawing, in which the outline of the building, work areas, aisles, and individual items of equipment are all drawn to scale. Another type of layout, or preliminary assembly, drawing is the design layout, which establishes the position and clearance of parts of an assembly. A set of working drawings usually includes detail drawings of all parts and an assembly drawing of the complete unit. Assembly drawings vary somewhat in character according to their use, as design assemblies or layouts, working drawing assemblies, general assemblies, installation assemblies, and check assemblies [3]. The drawing may be in the form of planned and checked mechanical drawings, or they may be freehand sketches.

Nowadays most engineers will use Computer-Aided Design (CAD) that helps them to draw such thing. AutoCAD is one of the examples of the most popular CAD for engineering drawing. With the help from CAD, engineers able to draw correctly and precisely. Not only able to draw the object but also engineers must able to interpret the meaning of the drawing. It could be to interpret the section of its part or symbol in the drawing. Thus, drawing and interpretation of the drawing is a basic important aspect for engineers.


AutoCAD 2007 - NOT FOR RESALE - [C:\My Documents\STD NEU.dwg]
File Edit View Insert Format Tools Draw Dimension Modify Express Window Help
ByLayer
STD NEU.dwg
VW252-02-0503.dwg
VW252-02-0203-1.dwg
Blocks and Tables - Imperial.dwg
3D House.dwg
SNAP GRID ORTHO POLAR OSNAP OTRACK DUCS DYN LWT

Command:

Command:

Figure 1 : Drawing Figures in AutoCAD

As this study is an important skill for engineers, many universities offer Engineering Drawing as their main course for engineering programs. Usually university offer engineering drawing program at the beginning of student study year, most of it on the first year. Students are being taught which will develop their skills for drawings and interpret the drawing result. While student skills are trained, lecturer in other hand must give them the right guidance. As an overview the numbers of lecturers or tutor who teach engineering drawing at university are not enough to accommodate student who enrolled. The difference number of lecture to student comes to a problem when lecturer or tutor are giving the student evaluation and marking to students works. Because the result form the drawing could be in a various drawings, it is not possible to have a fast evaluation system if the lecturers still use the manual way to determine the correctness from student work.

As one that often used for Engineering Drawing purposes, AutoCAD has DXF (Drawing Exchange Format) data file format which enable to interoperate with other program. The DXF format is an ASCII based tagged data representation of all the information contained in an AutoCAD drawing file. Inside DXF it contains different section such as HEADER, CLASSES, TABLE, BLOCKS, ENTITIES, OBJECTS and IMAGES. Inside documentation gives clear outline on every single code and terms used in DXF, allowing developers to build software or API that can communicate with the file format.

1.2 Problem Statement

Evaluate and marking an assignment for engineering drawing is a tiring job to be done by many lecturers in the university. They have to compare between the student assignment and the actual working. Completion on the evaluation could take days. Some of the comparison may not be accurate if they done manually. However it could be done easily by creating a system that could provide an automatic evaluation for engineering drawing. The system will input the data from the student assignment and compare it automatically with actual working from lecturer. The data input source is generated from AutoCAD as the main CAD source tools in this research. From that comparison system will get the DXF (Drawing Exchange Format) files and converted in the automatic system into SVG (Scalable Vector Graphic). System will compare SVG files from the student and the actual work or master work from lecturer. Then from the comparison result of SVG, system will give mark or grade the assignment based on the range of the difference.

The ability to read and convert drawing is also considered as the main problem. Limitation on previous research can be improves with the addition of reading and converting other basic shape from the drawing. The addition of the basic shape will include circle, line and polyline. Marking scheme is developed in order to complete the automatic evaluation. However, more investigation regarding the marking scheme is still need to be done. Previous research proves that the differences between lecturer and student drawing can be identified through the utilization of DXF file from AutoCAD. Thus ensure that every drawing will produce information that can be identified through their extension file. However the method used in Engineering Drawing marking will be different. It depends on each lecturer practice in their university.

1.3 Objective

The objective of this study is as follow:

a. Creating a system that will test system for efficiency in any actual case scenarios for Engineering Drawing drawings.
b. Compare between the actual (lecturer) and student work from engineering drawing assignment.
c. Add capabilities into the system to read the basic drawing.
d. Improving marking scheme from the result of comparison to meet the standard.

The study above is done in order to see the continuity of having an automatic evaluation for engineering drawing course. The continuity expressed in form of different shape and different types of drawing which the automated system able to read. Previous research could enable the automated system to read a basic shape. This basic shape consists of rectangle, circle, ellipse, line, polygon and text. For this research it will be more broad view in the drawing. It will cover the combination of basic shapes in order to form another drawing for example an Olympic logo. Also it includes more complicated drawings like a floor plan. Furthermore, the grading mechanism standard is also a part of the continuation of work from previous research. Having a standard in grading or marking will make beginning point for further marking scheme. Grading scheme will be determined by referring to Engineering Drawing lecturer. Lecturer will guide the method how to determine the marking system.

1.4 SCOPE OF STUDY

The basic of the study is about creating an automatic assessment system. First of all the capability of having an automated system will bring benefits over manual-made system. It will increase the productivity and performance for them who use it. In some institutions, the number of students surpasses the number of instructors by many fold, thus there are not enough instructors to assess assignments from these students. This raises the need for automatic ED assessment software [1]. Furthermore the conversion file types from AutoCAD especially DXF to SVG will be more investigate. Previous research only did on a basic shape conversion. However this research will focus on more broad type of drawings. It cover the various combination of basic shape, simple complicated drawing. It will be a different contain of file when it comes to different drawings and shape as well. Marking scheme also will be develop to meet the standard of Engineering Drawing requirement.

CHAPTER 2

2. LITERATURE REVIEW

2.1 AUTOMATIC ASSESSMENT

The need for computer-assisted assessment of learning outcomes is connected because of two inter-related factors in education. First the lecturer needs to automate the evaluation or assessment process especially in mass courses. Secondly is because students want to assess their works in a fast way especially for those who enrolling an online course [4]. From that we can see that most of automatic assessment is addressed to those courses which have a lot number of student. Since then nowadays the automatic assessment is becoming the trend in the education world especially for large number of student in one course in order to easing lecturer's work. This automatic assessment technique also enable lecturer monitoring their student by looking their behavior in submitting the assignment given. The entire process of automatic assessment can be done by computer [10]. However there also type of automatic assessment which can be in form of semi-automated. It also helps the lecturer because partially the works are done by the computer. Lecturer able to access the system however the computer will simplifies the process. It can be done by giving or providing lecturer with really-made form and also checking student grade consistency [11].

Traditionally assessments given by lecturer are should be check manually. This way not only takes up much time of the lecturer and need a long time for the lecturer to complete the correcting, but also can't send the feedback to the students timely, so it is not advantageous to the students' study. Therefore developing automatic assessment system is very helpful for improving the quality of teaching. Automated system greatly improved the efficiency of checking process, reduced the burden of the teacher. Moreover the students can correct the mistake which they can get from the feedback; and the system is very useful for mobilizing students' study [5].

Automatic assessment is substantially more demanding and expensive than assess multiples choice or other selection tasks. Automated assessment systems must preserve the benefits of written responses, be able to perform as accurately as human raters, increase the phase of assessment and reduce grading-related costs. In addition to reduced costs, automated assessment of essays can help achieve better accuracy and objectivity [8, 9]. Since computers can grade essays more rapidly than humans, essay writers can get instant feedback. An automated assessment system is not affected by errors caused by lack of consistency, fatigue or bias [4].

Computer as a perfect tool will provide the entire requirement in order to make an automated system. The trend nowadays is to create such an automatic system where all the works will be completely fast. There are numbers of developing automatic systems especially regarding the automatic evaluation or assessment such as programming, mathematic to language check automatic assessment.

2.2 ENGINEERING DRAWING FORMAT

Current CAD that supports Engineering Drawing purpose which commonly used is AutoCAD. In order to understand user must know the symbol representation in Engineering Drawing. Symbol recognition play significant role in Engineering Drawing. Once the interpretation false, the other could follow. Thus, learning symbol recognition is important [12].

The works in AutoCAD basically represented as DXF format. In previous research [13] there is a technique in order to represent DXF file into SVG. This SVG extension is easily to understand because it is a XML-based. Based on that, a DXF file is transformed into SVG with the same attribute. Beside real images also able to convert to SVG. SVG is an effective combination graphic, raster image and text. It can display various kinds of high quality vector graphics and can inherit all features of XML. More importantly, SVG is described by the text, which is designed for efficient [14]. Thus, Engineering Drawing works could be transferred into some code. Once it transferred developers able to build a system which enable it to read through the code.

2.3 AUTOMATED ASSESSMENT FOR ENGINEERING DRAWING

Referring to many works regarding the automatic assessment there is certain thing to build the same technique to be implemented in Engineering Drawing assessment. In order to make thing work, the drawings should be transforms into code in order to be able to make the automatic system. The reason is because Engineering Drawing is a mass course in university with small number of lecturer. Problem also arise when the lecturer assess student work and give proper mark. Thus, automatic assessment in Engineering Drawing is certainly needed. With the current capability which only takes number of basic drawing shape such as circle, a chance to develop to more complex drawing is considerably big [1].

As it is know that in Engineering Drawing student may be given any assessment with various type of drawing. The grading system also becomes one of the main concerns to develop to more accurate. With current research and prototype which only take a group of circle.

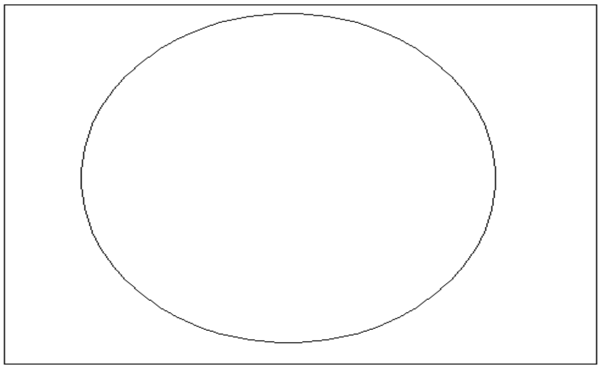


Figure 2 : Representation of Circle

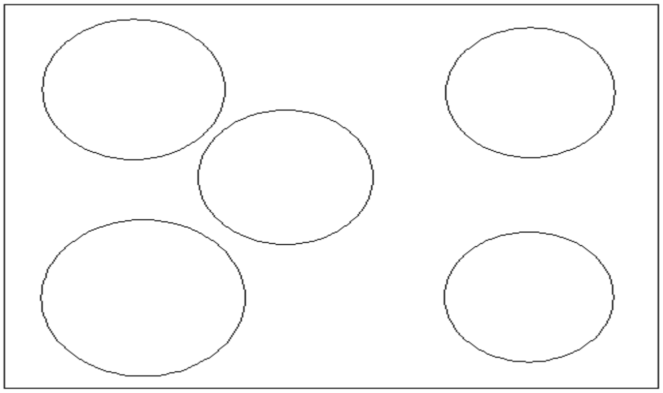


Figure 3 : Group of Circle

From the circle tag in SVG format there are three important attributes which are ‘r’, ‘cy’ and ‘cx’. ‘r’ represent the radius of a circle meanwhile ‘cy’ and ‘cx’ represent y-coordinate of circle’s midpoint and x-coordinate of circle’s midpoint respectively. Previous works dealing with Error Value to give the based grade for student. The Error Value is obtained by observing certain attribute in circle which is ‘r’, ‘cy’ and ‘cx’. System will compare both attribute from lecturer and student works. The schema of Error Value will be $EV = |r_m - r_s| + |cx_m - cx_s| + |cy_m - cy_s|$. Once system gets result of Error Value, there is also a Deduction Factor. The bigger Error Value the larger mark will be deducted. This marking scheme is to award marks for creativity based on the proposed formulas [1]. At that point, only circle to be tested using the prototype.

CHAPTER 3

3. METHODOLOGY

3.1 Research Methodology

As it stated in the introduction part the objective of this study is continuing the existing system in automatic assessment for Engineering Drawing course. This research aim to test for any other case scenarios in the drawings, as the previous research has done with the basic and group of circle drawings. Previously the research used prototype model for the system development life cycle developed with .Net framework Therefore it is also being implemented on this current study. The reason choosing prototype model is because it will reduce time of development. Prototyping can improve the quality of requirement and specification. Beside prototyping requires involvement and provide better feedback from the user. With the existing prototype that already been developed, this current research is meant to add another feature to the existing prototype.

The additional features require two main important aspects. The first aspect is the system ability to read another scenario then the current shape or drawing. The other one is improving the marking scheme for certain scenarios. For both aspects, this study requires a consultation from the Engineering Drawing practitioner which in this study referring to the lecturer. One of the senior lecturers at Universiti Teknologi PETRONAS is being interviewed on how Engineering Drawing course in the university assignments are being marked.

3.2 Data Collection

3.2.1 Interview

In order to know the current method on how Engineering Drawing assignment is being marked, an interview has been carried out. Most of the Engineering Drawing assignment is about drawing the object given by the lecturer. Some of the important notes during the interview are become the method of the system as the system need to assess the drawing and give a mark or reward. Lecturer single out that there are two main drawing element need to inspect before determine the mark for the drawings. Lecturer need to check the dimensional and projection comparison manually by printing out the drawings.

Using the two main aspect which is dimensional and projection comparison will help lecturer in common practice to determine the mark. Dimensional is all about the shape of the drawing itself. Some of the information for dimensional aspect can be seen in the DXF file that produced by AutoCAD. The things that need to know is which section that represents the drawing information. As an example, the HEADER entities contain the settings of variable associated with the drawings. Each of variables is specified by a group code giving the variable's name, followed by groups that supply the variable's value. Lecturer also point out that the important thing is to get the right value for its drawing. Because in DXF contains seven section, it must be precise to look for the right section that represent the drawing dimensional information. A more investigation related to this dimensional comparison will be carried out during the second phase of this study.

The projection comparison related to the angle of vision from the object. In order to draw an object in requires a look from different angle. There are four quadrant of it, meaning it could be projected from four different angle of vision. Beside the dimensional comparison, this projection comparison is also important aspect in order to give the right mark to student. Manually lecturer can see the projection directly from the print out result of the drawing.

In order for assignment to get mark lecturer need to see dimensional and projection. As in the DXF, dimensional information is already stated. It just needs the exact way to capture the right information. As for projection comparison needs a further understanding and yet to be investigate

3.2.2 OBSERVATION

The observation is carried out in order to know the lecture situation during Engineering Drawing course hour in class. The objective is to know how

Engineering Drawing assignments are given to the student. Beside that observation related to the marking system also being carried out. Marking scheme and any other related criteria that lecturer give to the student is also being observed. Figure below show the common print-out version of Engineering Drawing assignment by the student. Lecturer have to evaluate by calculate the height of the drawing manually.

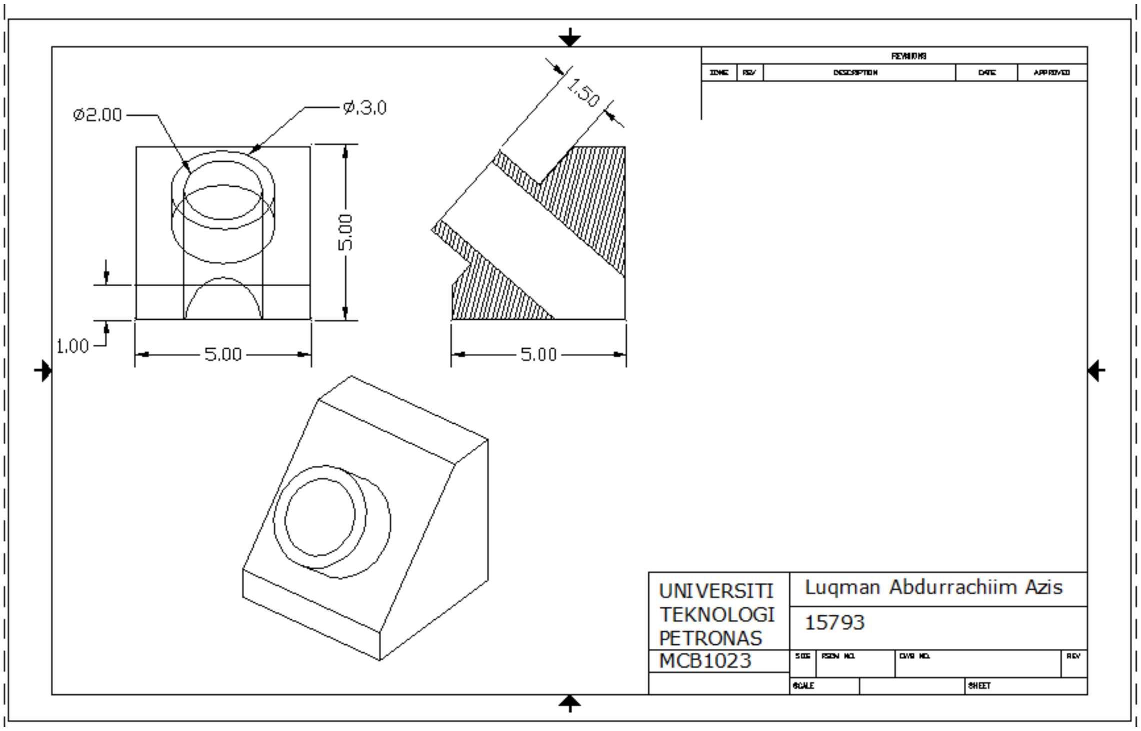


Figure 4 : Print-out version of assignment

3.3 Data Analysis

In order to get used to the different drawing scenarios, numbers of data are collected from student assignment given by the lecturer. The drawings are gathered and examined. This aims to know the variety of student assignment. Are the student exposed to the simple drawing, moderate drawings or complicated drawings. The data analysis is also aims to get to know the information inside DXF file. Therefore it able to convert to SVG files and captures the important section of the drawings.

3.4 Project Activities

3.4.1 Finding relevant literature

The project was started by finding relevant literatures related to the topic. It is intended to get deeper insight on current trend and work done on similar field. By having such information, effective and structured strategy can be developed to achieve the objectives of the project. The finding are focused on these topics, which are; engineering drawing general knowledge including on the evaluation and marking scheme, DXF extension and SVG extension knowledge.

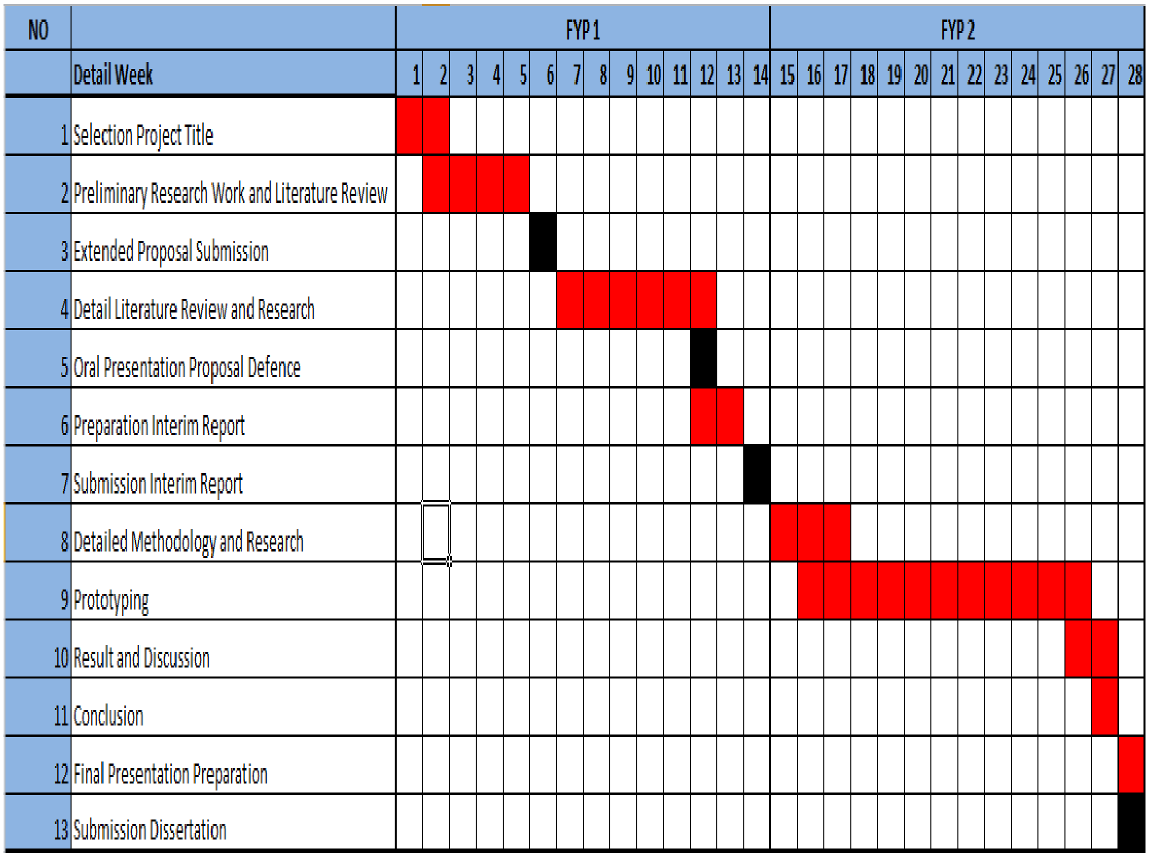
3.4.2 Learning C# language, Microsoft Visual Studio and AutoCAD

Deeper knowledge on developing application using C# language on the Microsoft Visual Studio software with .NET framework is highly needed before proceeding to make the application and prototype. Since the project is related to drawing object on AutoCAD, basic knowledge on using the software is also needed. Learning AutoCAD is done by attending some class for Engineering Drawing course.

3.4.3 Sample of drawing

In order to get used to DXF extension, some sample drawing from the student are being examined. This drawing includes a simple object drawing and some complicated drawing. The examination will look for the correct information of the drawing from DXF file. The information of the drawing will be saved on different section in the DXF file. Currently the research will be focused on Entities section, since most of the information of the drawing are being saved there.

3.5 GANTT CHART



NO		FYP 1														FYP 2													
	Detail Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	Selection Project Title	🟥	🟥																										
2	Preliminary Research Work and Literature Review		🟥	🟥	🟥	🟥																							
3	Extended Proposal Submission						⬛																						
4	Detail Literature Review and Research							🟥	🟥	🟥	🟥	🟥	🟥																
5	Oral Presentation Proposal Defence												⬛																
6	Preparation Interim Report												🟥	🟥															
7	Submission Interim Report														⬛														
8	Detailed Methodology and Research															🟥	🟥	🟥											
9	Prototyping																🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥		
10	Result and Discussion																										🟥	🟥	
11	Conclusion																											🟥	
12	Final Presentation Preparation																												🟥
13	Submission Dissertation																												⬛

Figure 5 : Gantt Chart

3.6 KEY MILESTONE

No	Milestone	Year 2013
1	Completion of Extended Proposal	25th June
2	Completion of Proposal Defense	31st July

3	Submission of Interim Report	21st August
4	Prototyping	14th October
5	Completion of Prototype	12nd December
6	Completion of Final Presentation	19th December
7	Completion of Dissertation	13rd December

Table 1: Milestone

CHAPTER 4

4. RESULT AND DISCUSSION

4.1 DATA INPUT

4.1.1 DXF FILE INPUT

In order to get a file converted to be readable, the system or application will need data input from AutoCAD in DXF (Drawing Exchange Format) format. There will be two files that have to be uploaded, first are the actual or master files from the lecturer the other one is student work. The system need to compare between these two files in order to get evaluate and marked. This DXF contain several sections such as Header, Classes, Table and Entities. The section which contains important information of the drawings for the data conversion is inside the Entities section. Each of different drawings will produce different information in the Entities section as well. It has to be clear for interpreting the information inside Entities section.

For different object of drawing will produce different Entities information. For the beginning this project will examine the Entities information on the each drawing object. Before, in order to understand the information needed to process the drawing, the information of the SVG shapes has to be examined. Thus, on later process determines the information on the DXF file will be easier, it will only look for the information on SVG file which contain the code for the object or shape. Basically there are 6 basic shape or object that SVG able to process. Each of the shapes has different code for information purposes. The shapes which SVG able to process are: rectangle, circle, ellipse, line, polygon and polyline. From this basic shape will be possible to create a combination object.

The following table is some sample of SVG code representation for each basic shape. From this representation code the system must be able to determine the same value from DXF Entities section, thus we are able to catch the information inside DXF Entities to be converted into SVG extension. This code will be the guidance for determining the information that should be taken from DXF file extension.

Shape	SVG Code
Circle	<svg xmlns="http://www.w3.org/2000/svg" version="1.1"> <circle cx=".." cy=".." r=".." stroke=".." stroke-width=".." fill="…"/> </svg>
Line	<svg xmlns="http://www.w3.org/2000/svg" version="1.1"> <line x1=".." y1=".." x2=".." y2=".." style="stroke:rgb(255,0,0);stroke-width:2"/> </svg>
Polyline	<svg xmlns="http://www.w3.org/2000/svg" version="1.1"> <polyline points="0,0 0,0 0,0 0,0 0,0 0,0" style="fill:..;stroke:..;"/> </svg>
Rectangle	<svg xmlns="http://www.w3.org/2000/svg" version="1.1"> <rect width=".." height=".." stroke-width: .."/> </svg>
Ellipse	<svg xmlns="http://www.w3.org/2000/svg" version="1.1"> <ellipse cx=".." cy=".." rx=".." ry=".." stroke-width: .."/> </svg>
Polygon	<svg xmlns="http://www.w3.org/2000/svg" version="1.1"> <polygon points="0,0 0,0 0,0" style="stroke-width:1"/> </svg>

Table 2 : SVG code representation

This combination of basic shape or group of basic shape will produce another drawing. These combination will be investigate in the next step for determining the value of Entities inside DXF file.

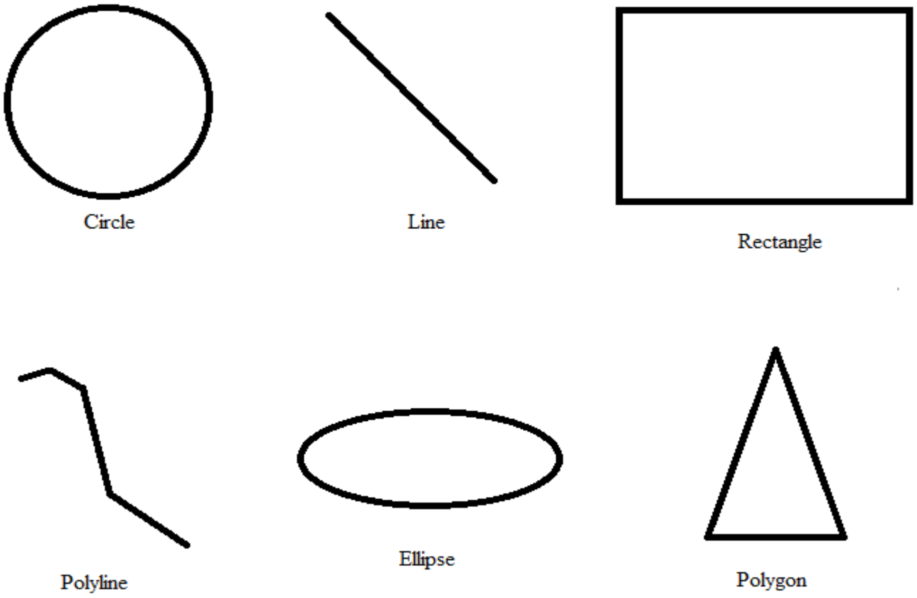
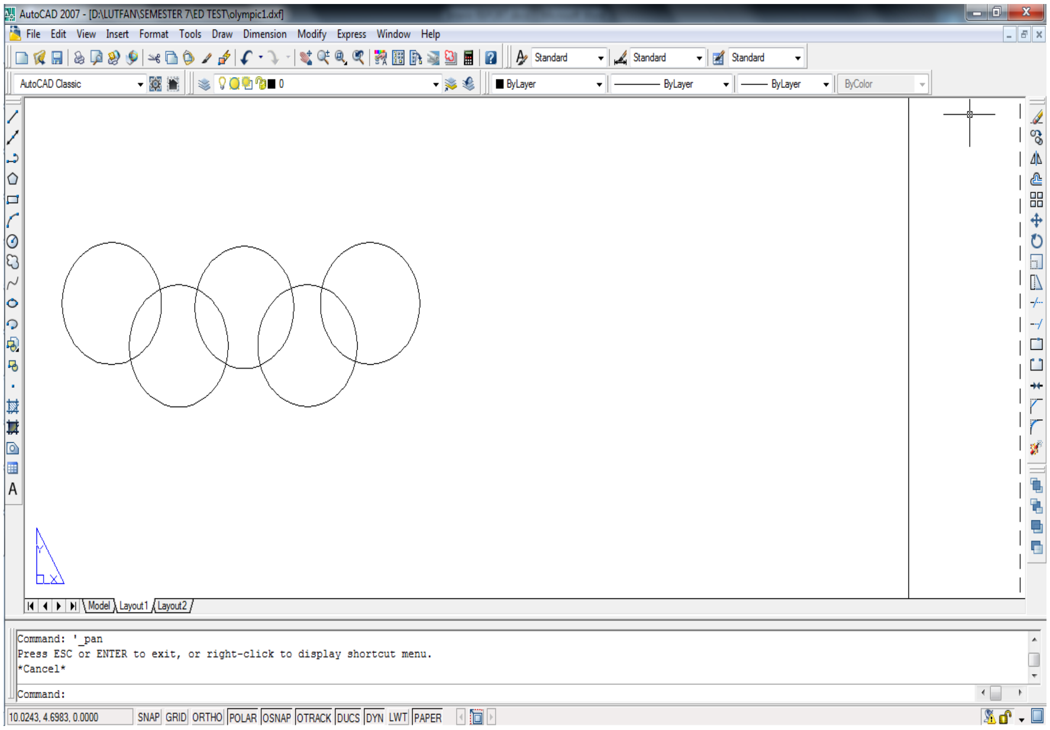


Figure 6 : Basic Shape

The actual or master file will be placed as the main reference for other student work. Once the system has the DXF file, system will take a conversion action. The DXF file will be converted into SVG file. A set of Java classes is created to transform all set of data stored in DXF format to its SVG counterparts This is because SVG is type of XML based which XML structured is easy to understand and it defines graphic attribute.

4.1.2 DXF TO SVG EXTENSION

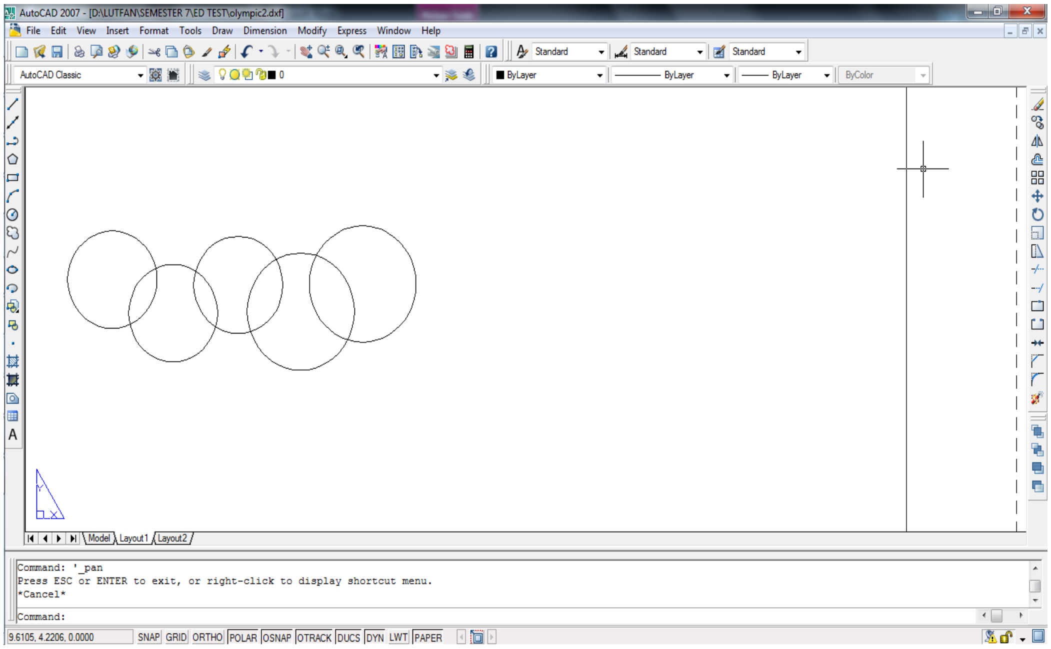
In this trial, the system will be accessing a group of circle shape to form an “Olympic” logo. There are 2 different drawing to be access. The first logo comes from lecturer drawing which is made as a reference to the other drawing by student. Then the second drawing is student drawing. If person see the drawing directly from the AutoCAD, it is look exactly the same. However, there will be slightly different position and size of the shape if person see from the DXF file. As the drawing finish, it should be saved under .dxf file.


AutoCAD 2007 - [D:\LUTFAN\SEMESTER 7\ED TEST\olympic1.dxf]
File Edit View Insert Format Tools Draw Dimension Modify Express Window Help
Standard
Standard
Standard
AutoCAD Classic
0
ByLayer
ByLayer
ByLayer
ByColor
Model
Layout1
Layout2
SNAP GRID ORTHO POLAR OSNAP OTRACK DUCS DYN LWT PAPER

```
Command: '_pan
Press ESC or ENTER to exit, or right-click to display shortcut menu.
*Cancel*
Command:
```

10.0243, 4.6983, 0.0000

Figure 7 : Lecturer drawing

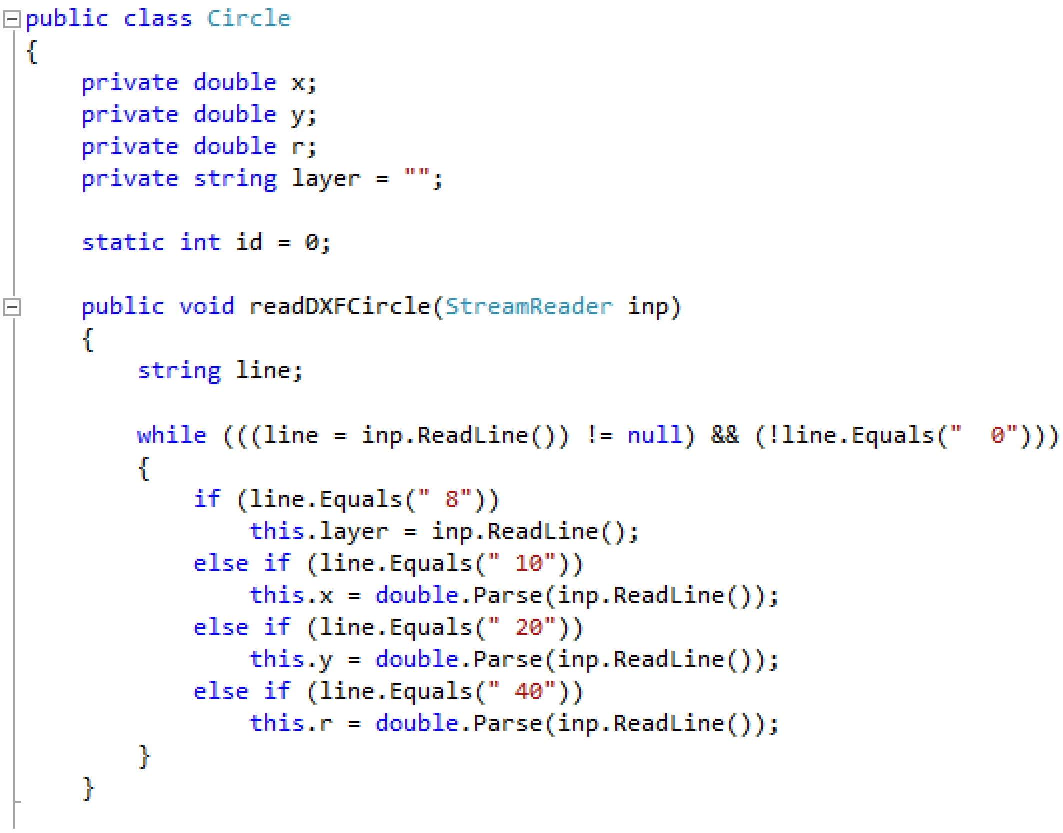

AutoCAD 2007 - [D:\LUTFAN\SEMESTER 7\ED TEST\olympic2.dxf]
File Edit View Insert Format Tools Draw Dimension Modify Express Window Help
Standard
Standard
Standard
AutoCAD Classic
0
ByLayer
ByLayer
ByLayer
ByColor
Model
Layout1
Layout2
SNAP GRID ORTHO POLAR OSNAP OTRACK DUCS DYN LWT PAPER

```
Command: '_pan
Press ESC or ENTER to exit, or right-click to display shortcut menu.
*Cancel*
Command:
```

9.6105, 4.2206, 0.0000

Figure 8 : Student drawing

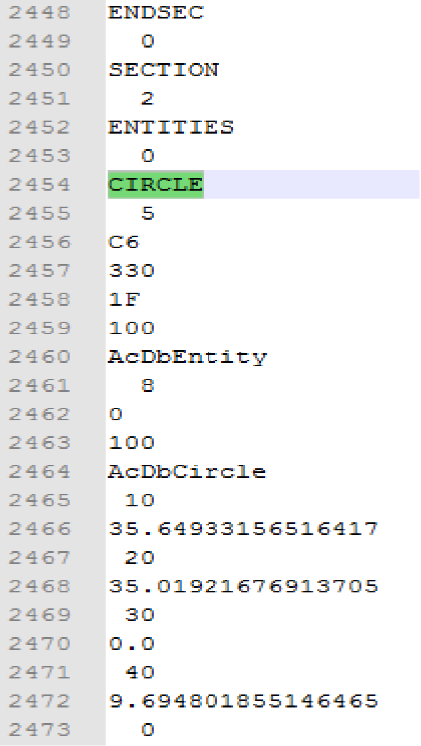
In order to accessing all the basic shape, in the system there will be a class to access each of the basic objects. In this class there are codes to read the DXF file and find the value of each object in Entities section. The following figure is the sample code used in circle shape for reading DXF file extension.



```
public class Circle
{
    private double x;
    private double y;
    private double r;
    private string layer = "";

    static int id = 0;

    public void readDXFCircle(StreamReader inp)
    {
        string line;

        while (((line = inp.ReadLine()) != null) && (!line.Equals("  0")))
        {
            if (line.Equals(" 8"))
                this.layer = inp.ReadLine();
            else if (line.Equals(" 10"))
                this.x = double.Parse(inp.ReadLine());
            else if (line.Equals(" 20"))
                this.y = double.Parse(inp.ReadLine());
            else if (line.Equals(" 40"))
                this.r = double.Parse(inp.ReadLine());
        }
    }
```

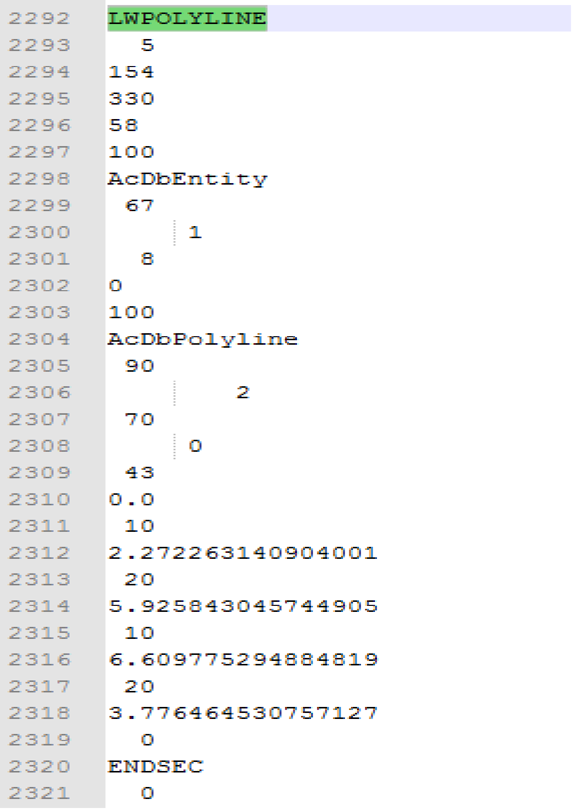
Figure 9 : Circle class

Basically the code is looking for the value in Entities section in the DXF file which has the same information as it describe on Table 2. In this trial, information which has the same information in circle is lines: “8”, “10”, “20” and “40”. This line is located under section “ENTITIES” in “CIRCLE” section. In order to get known the location of the Entities section, DXF file extension also able to open manually using notepad. Another sample of class code for some basic shape can be seen in Appendix.



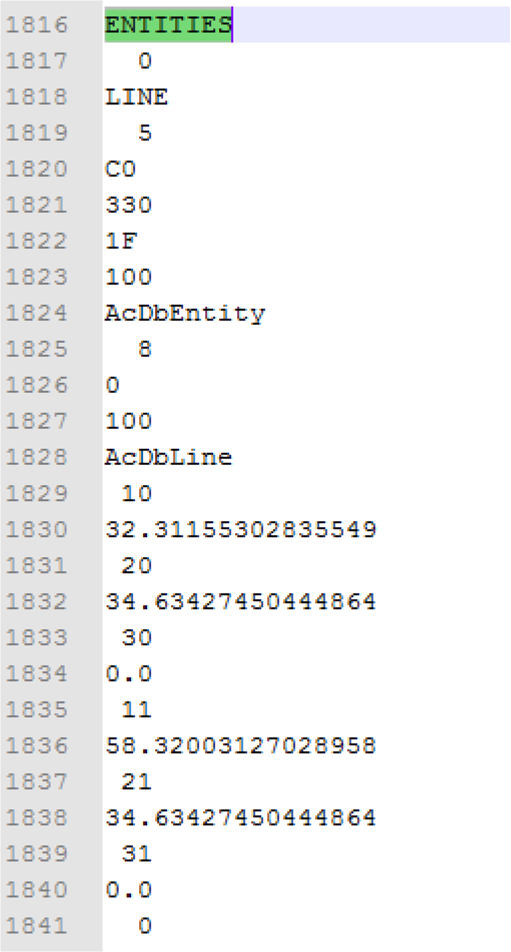
```
ENDSEC
  0
SECTION
  2
ENTITIES
  0
CIRCLE
  5
C6
330
1F
100
AcDbEntity
  8
0
100
AcDbCircle
 10
35.64933156516417
 20
35.01921676913705
 30
0.0
 40
9.694801855146465
  0
```

Figure 10 : Entities section in DXF for Circle



```
LWPOLYLINE
  5
154
330
58
100
AcDbEntity
 67
     1
  8
0
100
AcDbPolyline
 90
        2
 70
     0
 43
0.0
 10
2.272263140904001
 20
5.925843045744905
 10
6.609775294884819
 20
3.776464530757127
  0
ENDSEC
  0
```

Figure 11 : Entities section in DXF for Polyline



```
ENTITIES
  0
LINE
  5
C0
330
1F
100
AcDbEntity
  8
0
100
AcDbLine
 10
32.31155302835549
 20
34.63427450444864
 30
0.0
 11
58.32003127028958
 21
34.63427450444864
 31
0.0
  0
```

Figure 12 : Entities section in DXF for Line

Figure 10, 11 and 12 are meant to be showing the important in the DXF file. This part reveals the drawing information. The information is consists of the location of the drawing pointed by number.

4.1.3 DATA READING

After DXF file uploaded, the system will find the information regarding the drawing inside the Entities section. Once it converted to SVG, the information in the Entities is taken exactly as the drawings meaning. System will also search for any other information inside the DXF section regarding the importance of the drawing. Different drawings will produce different information in entities as well, so the system will put in group it accordingly. There are certain patterns in the entities which will show the drawing information such as its height, position and another graphical attribute. Recognizing this pattern is important. This pattern would be the reference in next step in order to determine its correctness.

Student	Lecturer

<svg
xmlns="http://www.w3.org/2000/svg"
version="1.1">
<circle cx="**9.667172840374078**"
cy="13.48236319755094"
r="2.702626290815926"
stroke="black" stroke-width="1"/>
</svg>
	<svg
	xmlns="http://www.w3.org/2000/svg"
	version="1.1">
	<circle cx="**8.667172840374078**"
	cy="13.48236319755094"
	r="2.702626290815926"
	stroke="black" stroke-width="1"/>
	</svg>

Shown above is the example of SVG code after conversion from the DXF file. It still contains the same information as the DXF file. The source is taken from student and lecturer works on a simple drawing of a circle. It shows us that there are a different value of ‘cx’, and ’cy’. It means that on the circle shape this part of SVG code shows the important information of the drawing. Different drawings will also have the part that contains the important information. On this trial, this system only shows the availability on circle shape. During next progress it will be accessing all the basic shape and combination of basic shape.

Another test in different basic shape drawing also show the same information as it is stated in DXF file. A simple polyline shape has been drawn in the AutoCAD. From that it produces a DXF file as it is shown below.

DXF file	SVG file

LWPOLYLINE
5
154
330
58
100
AcDbEntity
67
1
8
0
100
AcDbPolyline
90
2
	<svg
	xmlns='http://www.w3.org/2000/svg'
	version='1.1'>
	<polyline points = ' **2.272263140904 , 5.92584304574491 6.60977529488482 , 3.77646453075713**

70 0 43 0.0 10 **2.272263140904001** 20 **5.925843045744905** 10 **6.609775294884819** 20 **3.776464530757127** 0	-0.250000240295891,- 0.249999996245377 0,0 0,0'/> </svg>

After the system convert DXF file into SVG file, it stills show the same information as it is stated in DXF file. The information from DXF will be placed in the polygon point in SVG file. The code of conversion from DXF to SVG can be seen in the appendix section part polyline code.

4.1.3 MARKING SCHEMA

Generally in this part it will look into the difference between the actual or master work and student drawings. There are certain points or attribute that will show the differences and the system will capture it as a references. The parameters will be based on the lecturer works. As for now it is only able to capture the dimensional comparison. The previous marking schema based on Error Value still being used in this project.

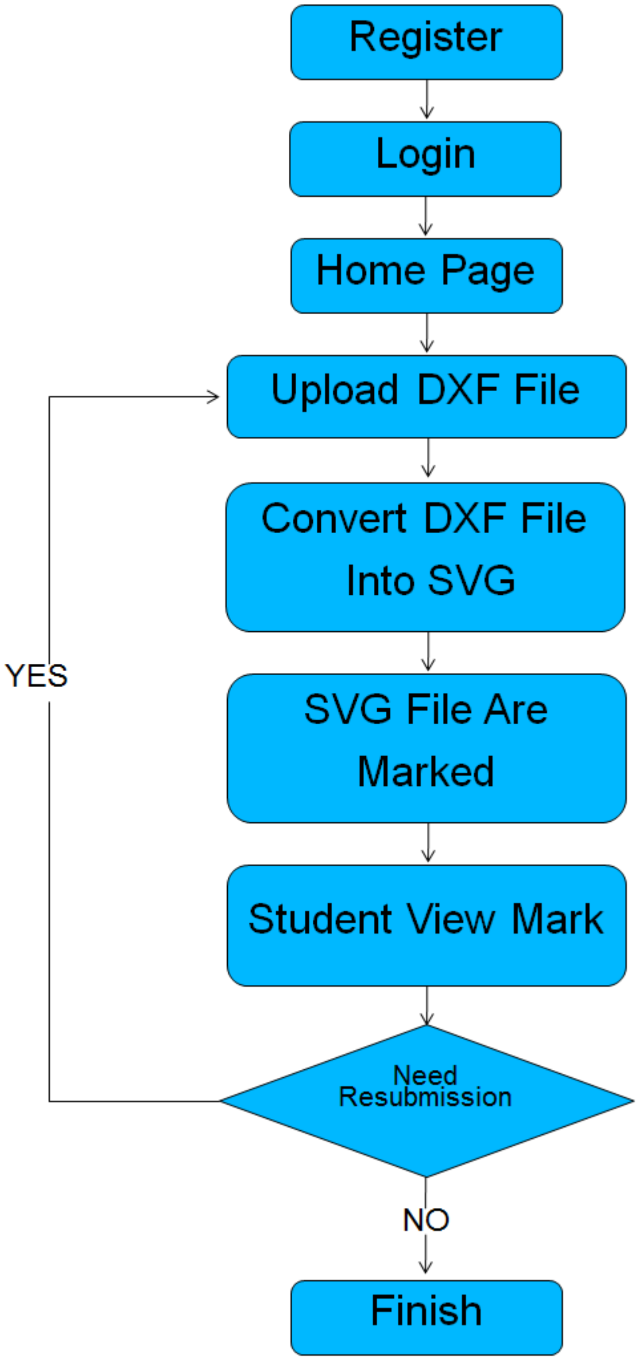
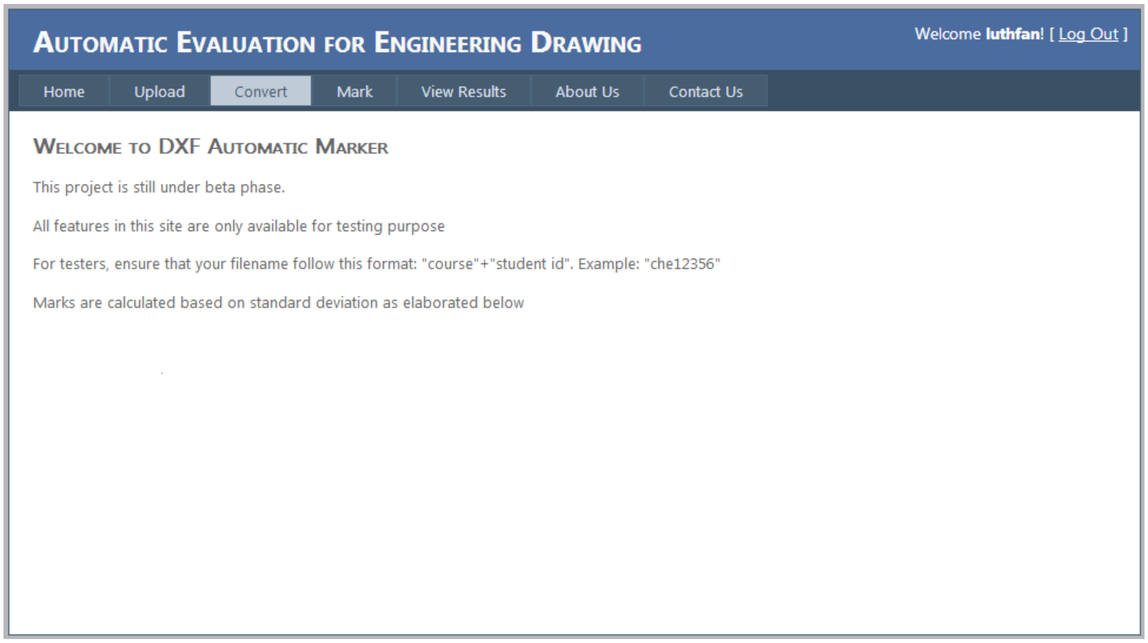


Figure 13 : Flowchart

4.2 DESIGN



AUTOMATIC EVALUATION FOR ENGINEERING DRAWING
Welcome luthfan! [Log Out]
Home
Upload
Convert
Mark
View Results
About Us
Contact Us

WELCOME TO DXF AUTOMATIC MARKER

This project is still under beta phase.

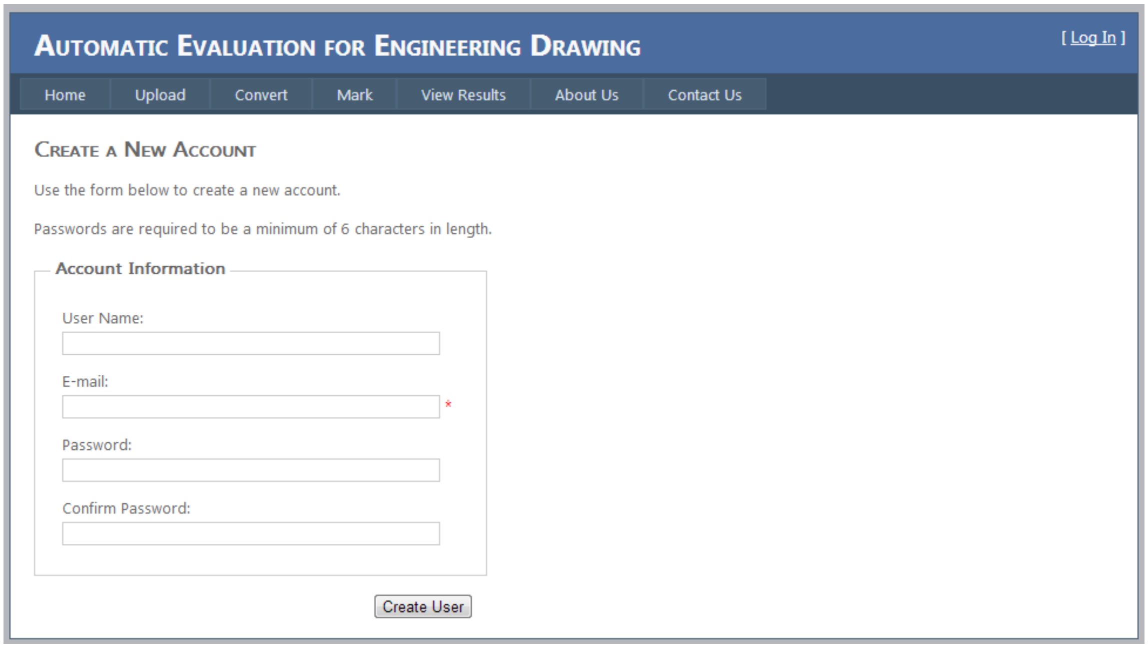
All features in this site are only available for testing purpose

For testers, ensure that your filename follow this format: "course"+"student id". Example: "che12356"

Marks are calculated based on standard deviation as elaborated below

Figure 14: Home Design

Figure 14 show the proposed home interface of the system. As it show that it have four function name Upload, Convert, Mark, View Result. Upload feature is for uploading the current assignment. Convert is for converting DXF to SVG file. Mark and View Result is for marking the assignment and view the result of it.



CREATE A NEW ACCOUNT

Use the form below to create a new account.

Passwords are required to be a minimum of 6 characters in length.

Account Information

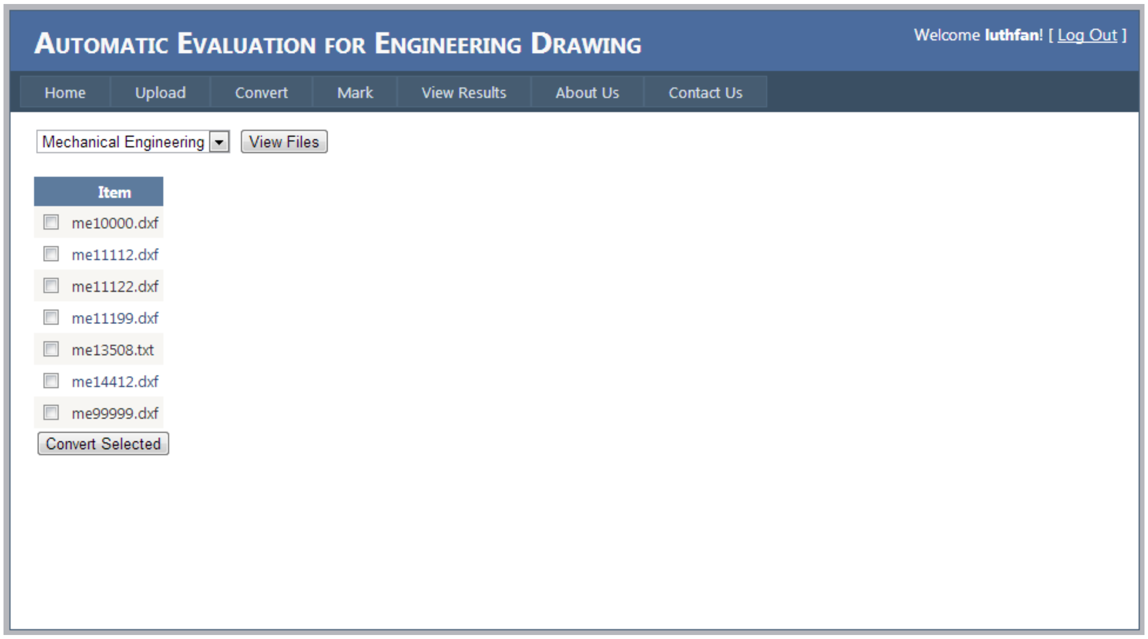
User Name:

E-mail: *

Password:

Confirm Password:

Figure 15: Login screen



Item
□ me10000.dxf
□ me11112.dxf
□ me11122.dxf
□ me11199.dxf
□ me13508.txt
□ me14412.dxf
□ me99999.dxf

Figure 16: Convert DXF to SVG

Figure 16 shows the .dxf file that will be converted into SVG extension. Student must save their drawing file into the following format: "course name" + "matric number". Thus, it will make easier to know the owner of the drawing. As an example "me13509" is belong to Mechanical Engineering student with matric number 13509

4.3 DISCUSSION

The final result of this project still can be developed into a more advance system in the future. Apart from the data reading and marking scheme that still being further investigation, the system also need some additional features. As it is describe before that this system only support for 2D (two-dimensional) object. Different practices in Engineering Drawing marking made the system still can be developed into more specific than the current system. The limitation of time and sources made this system not enough to accommodate bigger complexity of drawing. Current system support basic shape which still far from the common practice of Engineering Drawing that contain many complex drawing.

Marking scheme seems become the toughest challenge to overcome. Different lecture will have different way of evaluating and marking the assignment from student. While considering of having a new method that fits the entire marking scheme, this system still implements the previous method of marking by having the Error Value calculated. By having an Error Value it able to conclude the deduction factor.

Another important feature for the system that can be included for example, first it needs some feature that will limit the number of submission from student. This will be useful in order to the system accept multiple submission of an assignment. Student will only have two times submission per assignment given. Therefore, if student submit more than two submissions per assignment given it will not be counted. For the first trial of the system the number of submission will be available for more than two per assignment given. The purpose of doing this is to make student get used to the system. After the student and lecturer get used to the system, the possibility of having only one submission per assignment likely happen. The other is the time limit. System must be equipped with the time limitation for the assignment to be submitted. This will ensure that student submit the assignment given on time. The time limit will be set by the lecturer side into the system.

CHAPTER 5

5. CONCLUSION

This study is designed based on the main objectives which has been set. Firstly is to test system for efficiency in any actual case scenarios for Engineering Drawing drawings. This will determine that the system is able to read drawing from the assignment. The second objective is comparison of the drawing between actual from lecturer and student work. Lastly additional basic shape reading technique has been added into the system to improve the way of evaluating assignment.

The result from previous study shows that there is a way for Engineering Drawing assignment to become automatically assess and marked. The ability to read basic drawing which is circle has been proved. Conversion from the DXF file to SVG proved the efficient way of evaluating the information of the drawing from AutoCAD software. From the experiment, the additional basic shape evaluation has been added into the system to make it more reliable in order the system to evaluate different drawing. However, it is still in the early stage of single drawing from the basic shape which consists of circle, line and polyline. Lastly the same marking scheme is used in this project. This is because the existing marking scheme already show the important different from both lecturer and student work. Some modification to the scheme will be added if there is a possibility of error in next experiment.

REFERENCES

[1] K. N. Goh and R. B. H. Manao (2013). Awarding Grade / Assessment for Engineering Drawing through Automated Assessment. International Journal of Soft Computing and Software Engineering

[2] French,Vierck et al 1953. A Manual of Engineering Drawing for Student and Draftsmen,

[3] McGraw-Hill Concise Encyclopedia of Engineering, 2002. Retrieved from http://encyclopedia2.thefreedictionary.com/Engineering+drawing

[4] Kakkonen, Tuomo & Sutinen, Erkki (2004). Automatic Assessment of the Content of Essays Based on Course Material

[5] Ying, Xu et al (2008) Design of Online Automatic Assessment. International Conference on Computer Science and Software Engineering, 1275-1278.

[7] Kajatenowicz, Przemyslaw & Wierzejewski,Jedrzej (2005) Automatic Assessment of Math Skills

[8] Chung, G. K. W. K., O'Neil, H. F. Methodological Approaches to Online Scoring of Essays. CSE Technical Report 461. National Center for Research on Evaluation, Los Angeles, USA, 1997

[9] Page, E. B., Petersen, N. S. The Computer Moves Into Essay Grading. Phi Delta Kappan, 76(7): 561-565, 1995

[10] Malmi, L et al (2002) Experiences in Automatic Assessment on Mass Couse and Issues for Designing Virtual Courses. Innovation and Technology in Computer Science Education, 7th

[11] Jackson, David (2000) A Semi-Automated Approach to Online Assessment. Innovation and Technology in Computer Science, 5th, 164-167

[12] Yang, Ruo-Yu et al. Research of the Structural-Learning-Based symbol recognition mechanism for Engineering Drawing. International Conference on Digital Content Multimedia Technology and its Application (IDC), 6th

[13] Fahiem, Muhammad Abuzar & Farhan, Saima (2007). Representation of Engineering Drawings in SVG and DXF for Information Interexchange. International Conference on Circuits, Systems, Electronics,Control & Signal Processing.

[14] Jiazheng, Yuan et al. (2008) Converting Real Images to SVG based on XML. International Conference on Wireless, Mobile and Multimedia Networks

Appendix

Line class code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.IO;

namespace DXF_Reader.Entities
{
   public class Line
   {
      private double x1;
      private double y1;
      private double x2;
      private double y2;

      public void readDXFLine(StreamReader inp)
      {
         string line;

         while (((line = inp.ReadLine()) != null) && (!line.Equals("  0")))
         {
            if (line.Equals(" 20"))
               this.x1 = double.Parse(inp.ReadLine());
            else if (line.Equals(" 21"))
               this.x2 = double.Parse(inp.ReadLine());
            else if (line.Equals(" 30"))
               this.y1 = double.Parse(inp.ReadLine());
            else if (line.Equals(" 31"))
               this.y2 = double.Parse(inp.ReadLine());
         }
      }



  public void writeSVGLine(StreamWriter svg)
      {
         svg.WriteLine("<line x1='" + x1 + "' y1='" + y1 + "' x2='" + x2 + "' y2='" + y2
+"'/>");

         //svg.WriteLine(" id='CIR" + id++ + "' onmouseover='over(evt)' ");
         //svg.WriteLine("onmouseout='out(evt' onclick='click(evt)' ");
         //svg.WriteLine("style='fill:yellow;fill-opacity:0.25'");

         //svg.WriteLine(" aec:page='" + layer + "'");
```

```
            //svg.WriteLine("/>");

        }
    }
}
```

Appendix B

Polyline class code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.IO;

namespace DXF_Reader.Entities
{
    public class Point
    {
        public double x, y;

        public Point() {
            x = 0.0;
            y = 0.0;
        }

        public Point(double x, double y)
        {
            this.x = x;
            this.y = y;
        }
    }

    public class LWPolyLine
    {
        private List<Point> pts = new List<Point>();
        private string layer = "";

        static int id = 0;

        public void readDXFLWPolyLine(StreamReader inp)
        {
            string line;

            Point pt = new Point();
            while (((line = inp.ReadLine()) != null) && (!line.Equals(" 0")))
            {
                if (line.Equals(" 8"))
                    this.layer = inp.ReadLine();
                else if (line.Equals(" 70"))
                {
                    line = inp.ReadLine();
                }
                else if (line.Equals(" 10"))
                {
                    pt = new Point();
                    pt.x = Double.Parse(inp.ReadLine());
                }
                else if (line.Equals(" 20"))
                {
                    pt.y = Double.Parse(inp.ReadLine());
                    pts.Add(pt);
                }
            }

        }

        public void writeSVGLWPolyLine(StreamWriter svg)
        {
            svg.Write("<polyline ");

            svg.Write(" points=\"");
            foreach (Point pnt in pts)
            {
                //if (pnt.x > 0 && pnt.y > 0)
                        svg.Write(" " + pnt.x + "," + (pnt.y));
            }

            svg.WriteLine("\"/>");
        }

    }
}
```

Appendix C

Circle class code

```
using System.IO;

public class Circle
{
   private double x;
   private double y;
   private double r;
   private string layer = "";

   static int id = 0;

   public void readDXFCircle(StreamReader inp)
   {
      string line;

      while (((line = inp.ReadLine()) != null) && (!line.Equals("  0")))
      {
         if (line.Equals(" 8"))
            this.layer = inp.ReadLine();
         else if (line.Equals(" 10"))
            this.x = double.Parse(inp.ReadLine());
         else if (line.Equals(" 20"))
            this.y = double.Parse(inp.ReadLine());
         else if (line.Equals(" 40"))
            this.r = double.Parse(inp.ReadLine());
      }
   }

   public void writeSVGCircle(StreamWriter svg)
   {
      svg.WriteLine("<circle cx='" + x + "' cy='" + (y) + "' r='" + r + "'/>");

      svg.WriteLine(" id='CIR" + id++ + "' onmouseover='over(evt)' ");
      svg.WriteLine("onmouseout='out(evt' onclick='click(evt)' ");
      svg.WriteLine("style='fill:yellow;fill-opacity:0.25'");

      svg.WriteLine(" aec:page='" + layer + "'");
      svg.WriteLine("/>");

   }
}
```

Appendix D

Convert DXF to SVG partial code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.IO;
using DXF_Reader.Entities;

namespace DXF_Reader
{
    public partial class Convert : System.Web.UI.Page
    {
        static string classess = "";
        List<String> filelist = new List<string>();
        List<String> fullpathlist = new List<string>();

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btnCheckFiles_Click(object sender, EventArgs e)
        {
            if (ddlClass.SelectedItem.Text == "Mechanical Engineering")
            {
                classess = "me";
            }

            else if (ddlClass.SelectedItem.Text == "Chemical Engineering")
            {
                classess = "che";
            }

            else if (ddlClass.SelectedItem.Text == "Civil Engineering")
            {
                classess = "cv";
            }

            EnumerateFiles(classess);

            if (filelist.Count > 0)
            {
                gvFiles.DataSource = filelist;
                gvFiles.DataBind();
                btnConvertSelected.Visible = true;
            }
        }

        protected void EnumerateFiles(string classess)
        {
            string[] dxffiles = Directory.GetFiles(Server.MapPath("~/Student Files/DXF/"),
classess+"*");

            foreach (string file in dxffiles)
            {
                filelist.Add(Path.GetFileName(file));
                fullpathlist.Add(file);
            }
        }

        protected void btnConvertSelected_Click(object sender, EventArgs e)
        {
            List<string> tobeconverted = new List<string>();

            foreach (GridViewRow row in gvFiles.Rows)
            {
                CheckBox cb = (CheckBox)row.FindControl("chkStatus");

                if (cb.Checked == true)
                {
                    tobeconverted.Add(Server.MapPath("~/Student Files/DXF/" +
row.Cells[1].Text));

                }
            }

            foreach (string fullpath in tobeconverted)
            {
                ConvertDXF(fullpath);
            }
        }

        protected void ConvertDXF(string infilepath)
        {
            string infilename = Path.GetFileName(infilepath);
            infilename = infilename.Remove(infilename.IndexOf(".dxf"));
            string line = "";

            try
            {
```

```
        StreamReader inp = File.OpenText(infilepath);
        StreamWriter outp = new StreamWriter
(File.OpenWrite(Server.MapPath("~/Student Files/SVG/"+infilename+".svg")));
        //outp.WriteLine("<?xml version=\"1.0\" encoding=\"iso-8859-1\"?>");
        //outp.WriteLine("<!DOCTYPE svg PUBLIC \"-//W3C//DTD SVG
1.0//EN\"");
        //outp.WriteLine("\"http//www.w3.org/TR/SVG/DTD/svg10.dtd\">");
        outp.WriteLine("<svg xmlns=\'http://www.w3.org/2000/svg' version='1.1'> ");
        //outp.Write("preserveAspectRatio='xMidYMax'");
        //outp.WriteLine("\txlmns='http://www.w3.org/2000/svg'");
        //outp.WriteLine("\txlmns:xlink='http://www.w3.org/1999/xlink'");
        //outp.WriteLine("\txlmns:aec='http://www.sams.com/2002/aec'");
        //outp.WriteLine("<title>" + infilename + "</title>");
        //outp.WriteLine("<g id='ACTIVE' style='fill:none;stroke:black'>");

        //while (((line = inp.ReadLine()) != null) &&
(!line.ToUpper().Equals("ENTITIES")))
        //{ }

        while (((line = inp.ReadLine()) != null) &&
(!line.ToUpper().Equals("ENDSEC")))
        {
          if (line.Equals("CIRCLE"))
          {
            Circle c = new Circle();
            c.readDXFCircle(inp);
            c.writeSVGCircle(outp);
          }

          else if (line.Equals("LWPOLYLINE"))
          {
            LWPolyLine lw = new LWPolyLine();
            lw.readDXFLWPolyLine(inp);
            lw.writeSVGLWPolyLine(outp);
          }

          else if (line.Equals("LINE"))
          {
            Line objLine = new Line();
            objLine.readDXFLine(inp);
            objLine.writeSVGLine(outp);
          }
        }

        //outp.WriteLine("</g>");
        outp.WriteLine("</svg>");
```

```
                inp.Close();
                outp.Close();
            }

            catch (Exception e)
            {
                StatusLabel.Text = e.Message;
            }
        }

    }
}
```