MEF UNIVERSITY

Department of Engineering
Programming Studio

STUDENT SURVEY ANALYSIS

Muhammed Rahmetullah Kartal
Ramazan Yetişmiş
Ertuğrul Yılmaz
Instructor: Muhittin Gökmen
Date: 14 April 2019

Abstract

In this project, we developed a program which can analyze one or more student surveys and creating multiple charts from them. This paper contains an explanation of reading an excel file, creating a graphic user interface by using JavaFX, making three different charts (Pie, Radar, and Bar) and getting a specific from big data.

Also, it contains a few samples of graphic user interface and charts.

Problem Definition

In the 21st century, there is a whole bunch of data about everything. It is an advantage to have big data. But with raw data, you cannot learn anything significant. People need to process on that big data to get a meaning from it. It is the same with student surveys at a university. Surveys getting more and more. Universities need to use big data on the most efficient way. It is nearly impossible to process on that kind of big data with human power. Big data must be analyzed by computer programs.

Solution

Developing a Computer Program

It is the fastest way to analyze data. Surveys may consist of different lectures by they are on the same order on an excel file. An excel file can be formed as an array. And file can be stored in an array. With that, a survey can be formed in the virtual environment. And the program should be used by everybody. That is a point for creating a GUI (Graphic User Interface). The GUI has to be simple and useful. And to get charts, firstly program need to calculate values on the survey. To do those calculations it uses different methods with different algorithms. For example, creating an algorithm that calculates the average of a section should be written with true parameters to use for every section and every survey. The most important thing is that the program must be compatible with all surveys in the long-term.

Work

Java

Java was very useful for a work like this because a survey will have different sections, different subsections, an instructor, a year and etc.. That means you will need many variables to store them. And Java is an Object-Oriented Programming Language. For many variables like this, we made classes to create a hierarchy inside a survey. That made the excel files virtualization much more understandable.

Reading an Excel File

To read excel files we implemented the library Apache POI library. We created a file class, section class and subsection class for an excel file. And we implemented them hierarchically.

Creation of the Main Chart Classes

For charts we implemented the xchart library. Then, we decided our main radar, bar and pie chart form. We created class for every different chart types. To create a chart in our main class, we used our calculation algorithms an created as object.

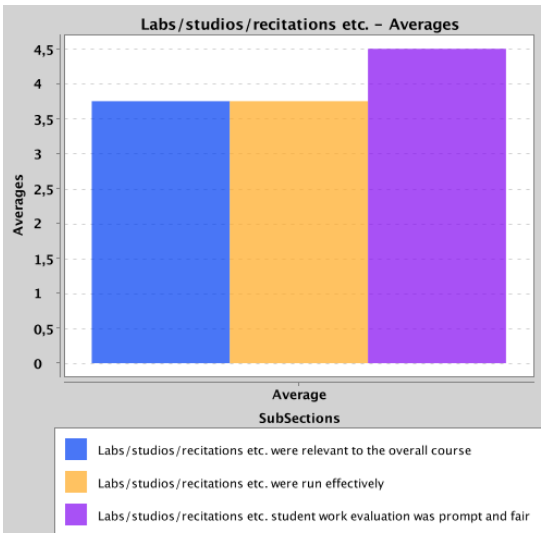

Figure 1: A Bar Chart Sample

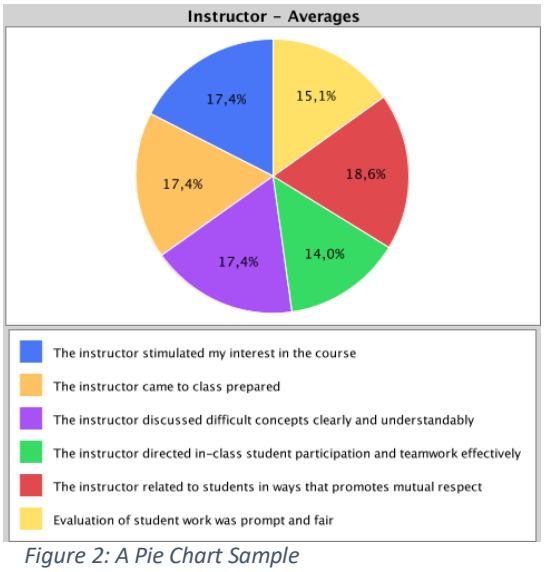

Figure 2: A Pie Chart Sample

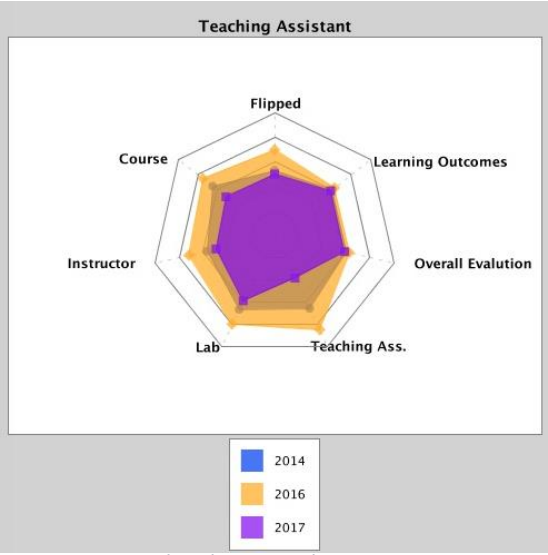

Figure 3: A Radar Chart Sample

Developing Algorithms

We wrote our algorithms as much as compatible with every survey. To do that we looked more than one excel files and understand the general frame of them. After that, we wrote the charts that we going to need before coding. On coding, we wrote our variables clear for our group work. We developed an algorithm for the average of a section, an algorithm for the average of all sections, an algorithm for the votes on a section, etc.. With all algorithms we combined them in different methods for each chart.

Load_SingleFile(File file):

This method takes a File object to get the path and open the excel file. Then it begins reading the file cell by cell, later it fills the File_f object which has been created. Then , the method returns a File_f object. This return statement is an easy way to multi_file operations.

Load_MultileFile(File files[]):

```
//Reading multifiles.
public static void  Load_Multi_Files(ArrayList<File_f> files, ArrayList<File> multi) {
    for (int i = 0; i < multi.size(); i++) {
        File single = multi.get(i);
        files.add(Load_File(single));
    }
}
```

Figure 4: Multi file reading method

As we can see from(Figure4) this method calls the load_File() method and add the returned File_f object to the files ArrayList.

givenCourseSection(File file):

```
//This method returns 1d array like   sectionX[2014,2015,2016,2017]
public  static  double []  givenCourseSection(ArrayList<File_f> files,String section){
    ArrayList<File_f> x1=files;
    double average[]=new double[4];
    int div[]=new int[4];
    for (int i = 0; i < x1.size(); i++) {
        double total=0;
        int div1=0;
        File_f f=x1.get(i);
        for (int j=0;j<7;j++) {
            if (f.Sections[j].getText().equals(section)){
                for (Subsection sb:f.Sections[j].subsections) {
                    total += sb.average;
                    div1++;
                }
            }
        }
        int index=(f.year-2)%4;
        div[index]+=f.getÖğrenciSayısı();
        average[index]+=total/div1*f.getÖğrenciSayısı();
        }
    for (int i = 0; i <average.length ; i++) {
        average[i]=average[i]/div[i];
        }
        return average;
}
```

Figure 5: Multi file getting Averages

As we can see from(Figure5) the method takes multi _files and calculates their averages of all the years separately and the method also takes into consideration the number of students. Then it returns a double array size of 4 like the first index is average of 2014, second index 2015 carries on up to 2017.

Creation of GUI (Graphic User Interface)

On the GUI we used mainly two(Java FXML, Scenebuilder) and one extra program(Swing). FXML is for the general design of the program like assigning work to a button to do, and Scenebuilder is for helping the placements of buttons and etc. on the main program.
We divided GUI to 5 parts in order to meet the requires:

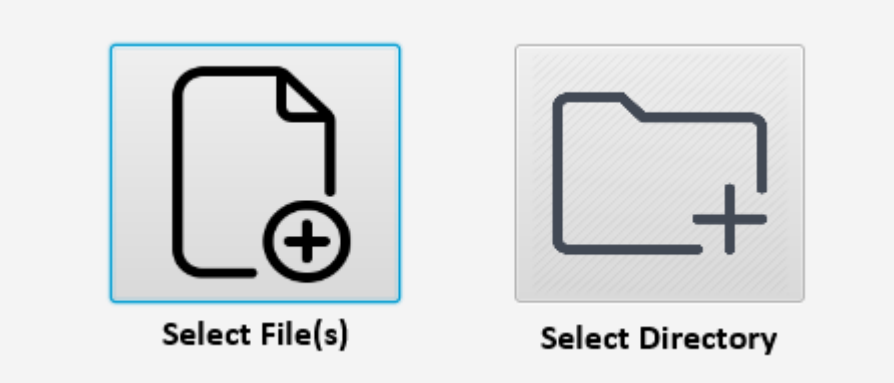

Figure 6: First part of the GUI, Selecting Survey Files

The first part is about selecting survey files from the computer, as you can see in the Figure 6 this part have two options selecting the specific file(s) or selecting a directory, if you press

“select file(s)” button you can select just one or multiple files at the same time, if you press “select directory” button you can select a directory and it gives all the files in that directory with Course_Code_Section_Year_Term.xls format.

Course Code	Section	Year	Term
CSSE120	1	2016	1.
CSSE120	1	2017	1.
CSSE137	1	2014	1.
CSSE137	1	2015	1.
CSSE137	1	2016	1.
CSSE137	1	2017	1.
CSSE149	1	2014	1.

Figure 7: Second part of the GUI, Course Table

At the second part, we created a Course Table as you can see in the Figure 7 which has Course Code, Section, Year, Term sections and imported the selected files into that Course Table with necessary formats. It refreshes itself on every different file selection(at one time).

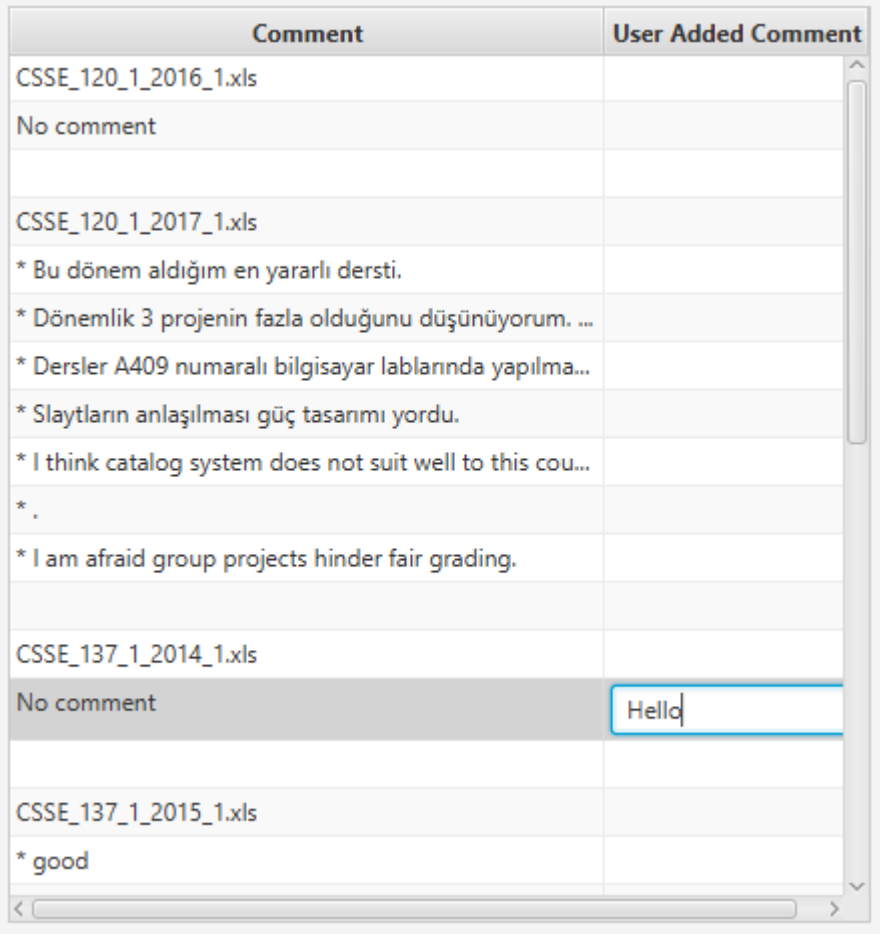

Figure 8: Third part of the GUI, Comments Table

At the third part, we created a Comments table as you can see in Figure 8 which has Comment and User-Added Comments sections. This table has a format like below

File 1 Name
Comments on File 1
-Space-

It allows you to add a translation or a new comment for a comment and visualize it on GUI.

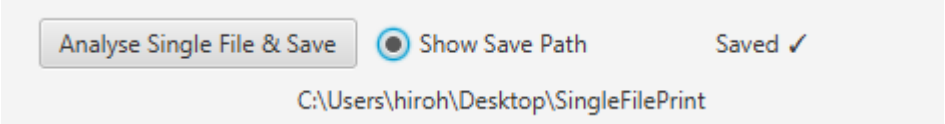

Figure 9: Fourth part of GUI, Single File Analyze and Save

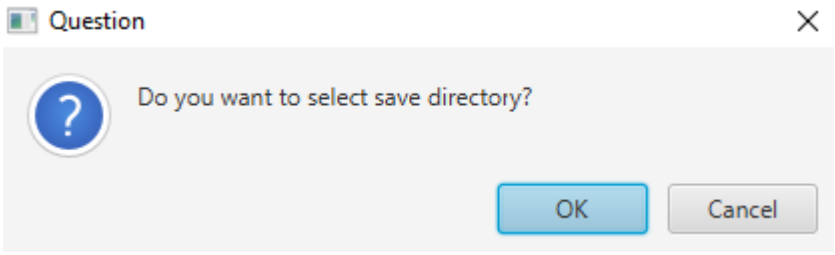

Figure10: Single file analysis, choosing to save directory

The fourth part which is below the Course Table is Single file analysis, when you select a single file from Course Table and you clicked "Analyse Single File & Save" button it allows you to select a save directory like on Figure 10 from your computer if you press cancel to that it saves the analyses to your working directory. The analysis has 100 different PNG typed charts about a single file and a PDF file to store that charts, their information, and file analysis like Course name, Instructor name, Comments, etc.. After that save operations as you can see in Figure 4 you get a warning about is file saved and after that you can see the current save path.

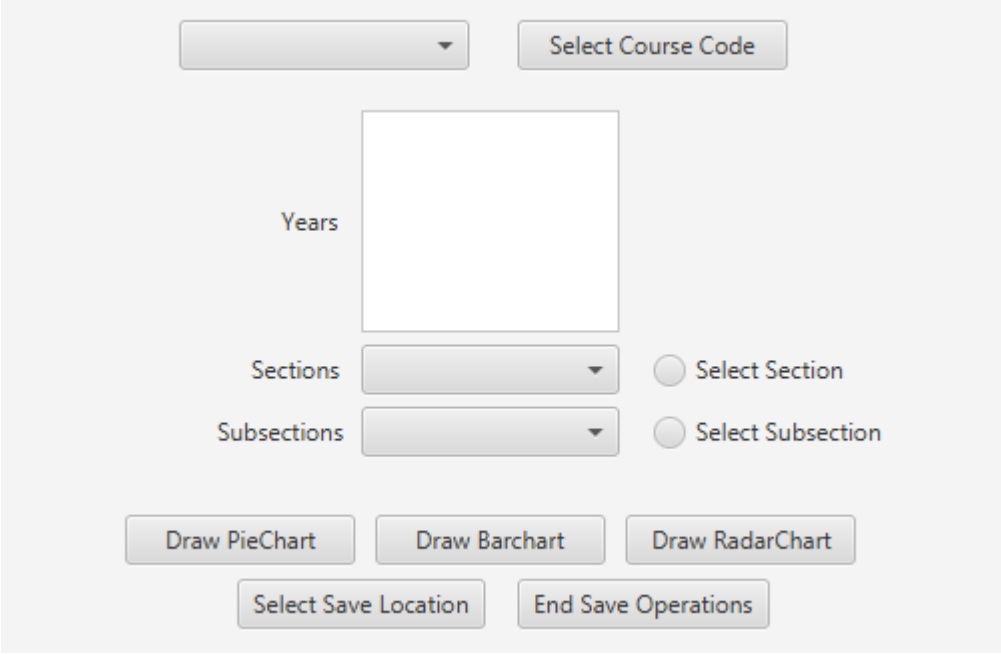

Figure 11: Fifth part of GUI, Multiple File Analysis

At the fifth part of GUI as you can see in Figure 11 there is a "Select Course Code" button and there is a choice box next to it, inside that choice box you can see all the Course Codes which are in your selected files and an extra option for selecting ALL FILES, when you select a course code and press "Select Course Code" button the years, sections and subsections of that courses appears and you can select them to make your analyze, you can select multiple years at one time. By the way to use in your analyse you can select Section or Subsection after all those selections you can draw Three different charts by clicking draw chart buttons, they pops up your selected chart in a new screen and actually saves it on your selected directory, but at first of all you need to select a save location by clicking "Select Save Location". When you pressed "End Save Operations" button it makes your PDF File ready.

If you want to create a new PDF file named Multiple File Analysis2.pdf you can select a new save location after pressed “End Save Operations” button.

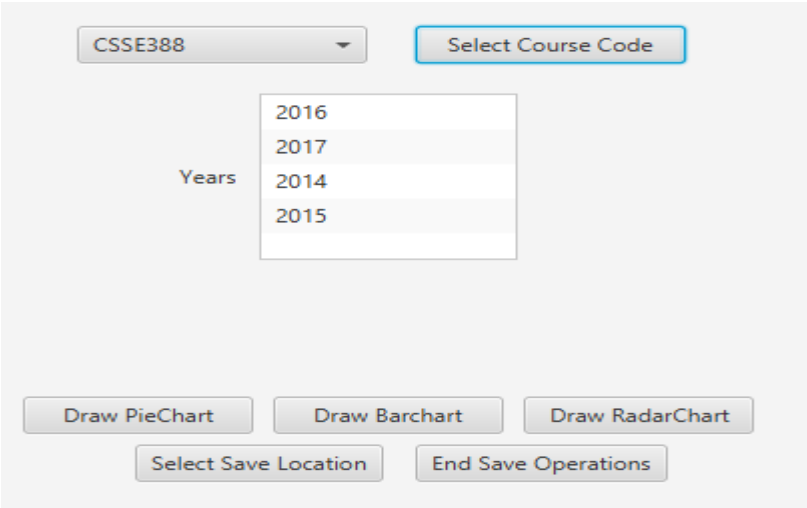

Figure 13: Select the single course code on multiple file analysis.

By the way on the fifth part if you select a single course for the course Sections and Subsections parts become invisible as you can see in Figure 13.

Results

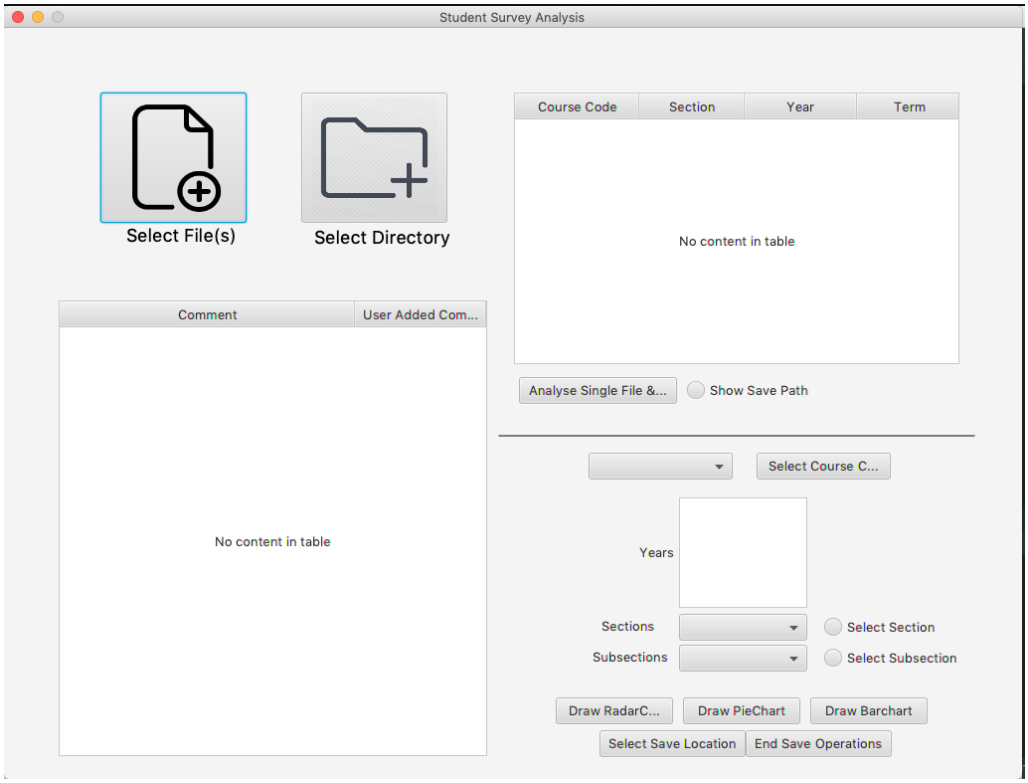

Figure 14: The Opening Screen of Program

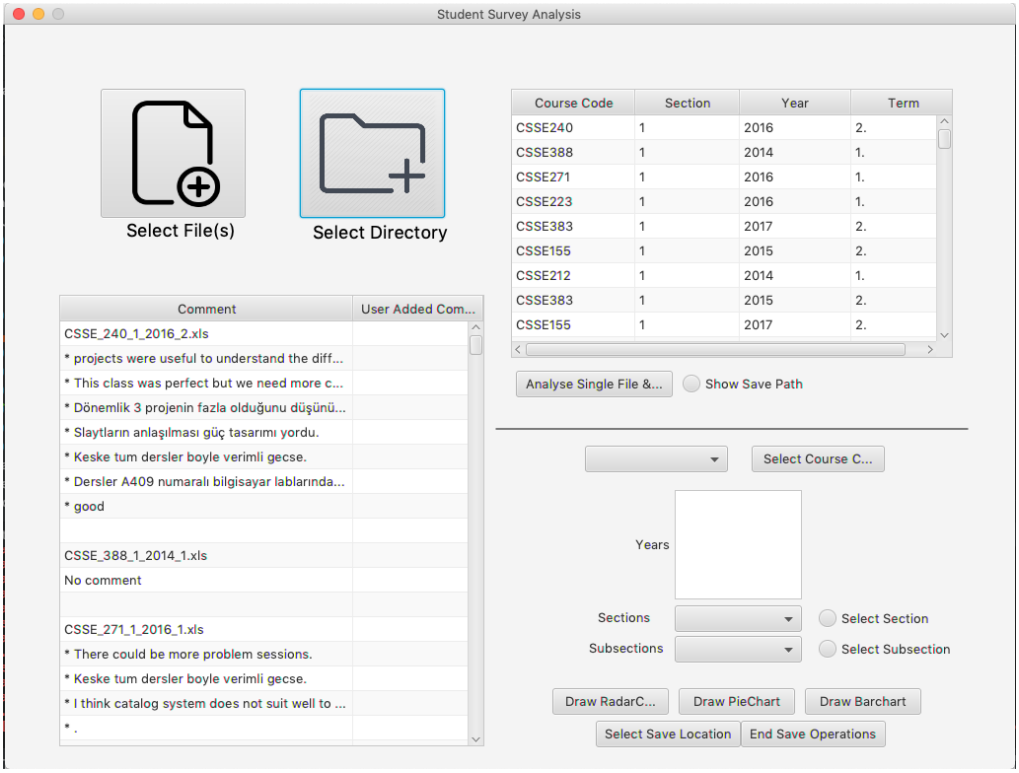

Figure 15: Selected Files are on the Program

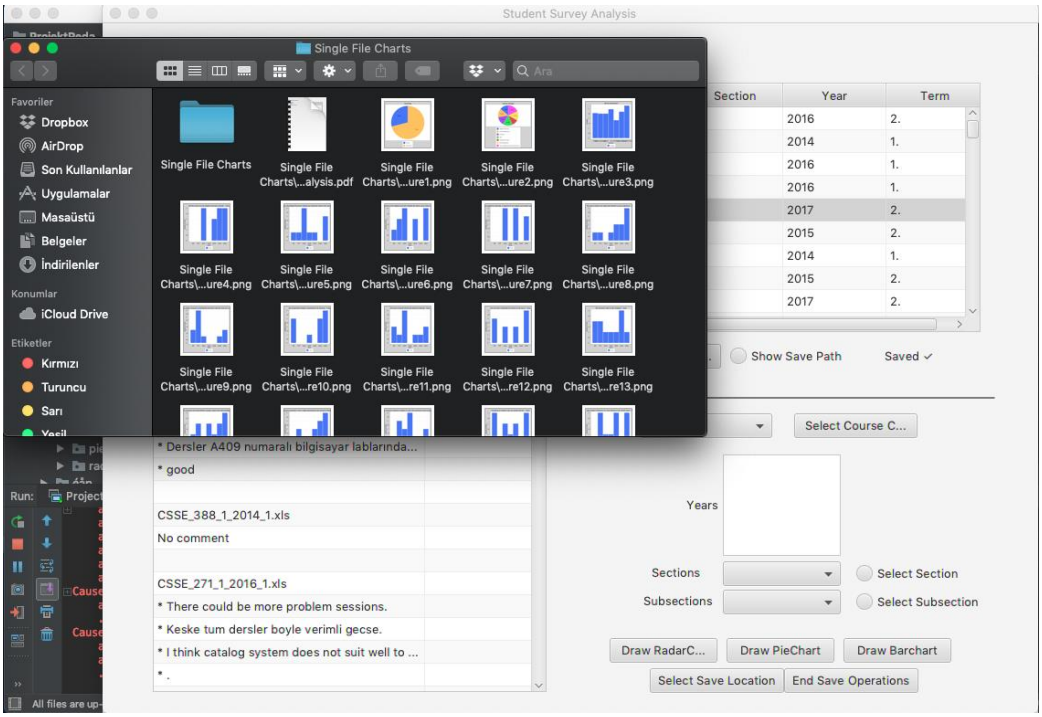

Figure 16: Selected Single File Charts Saved To Wanted Directory

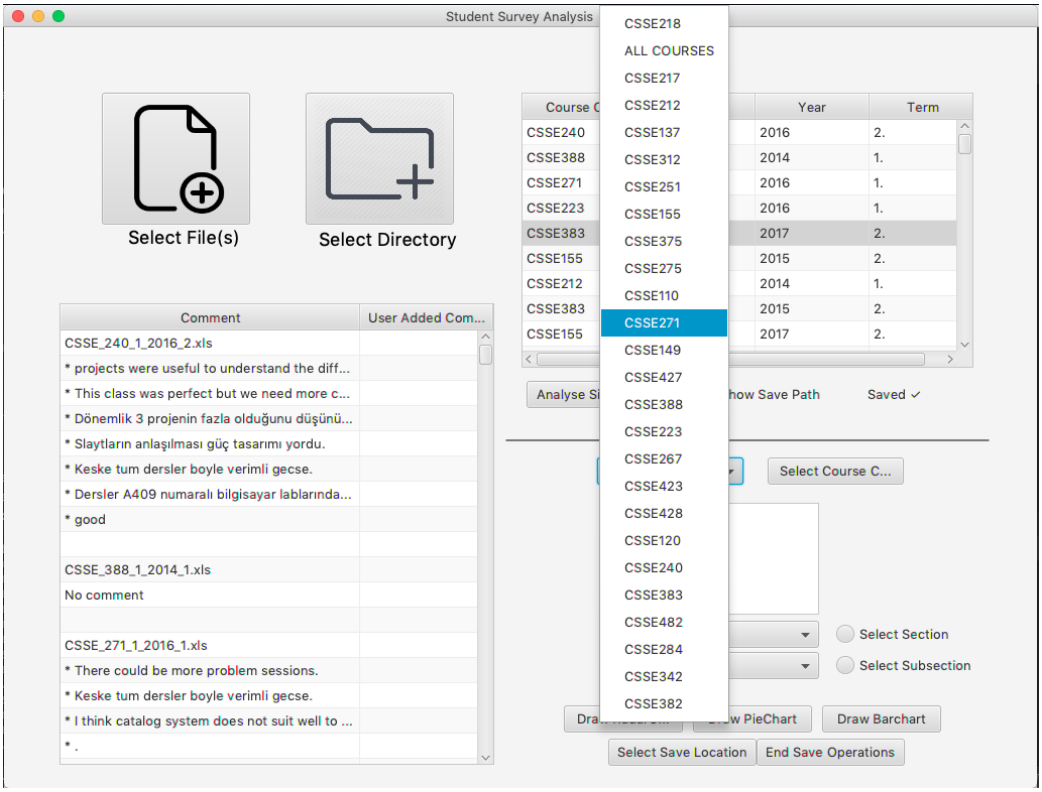

Figure 17: Selection of Wanted Course or All Courses

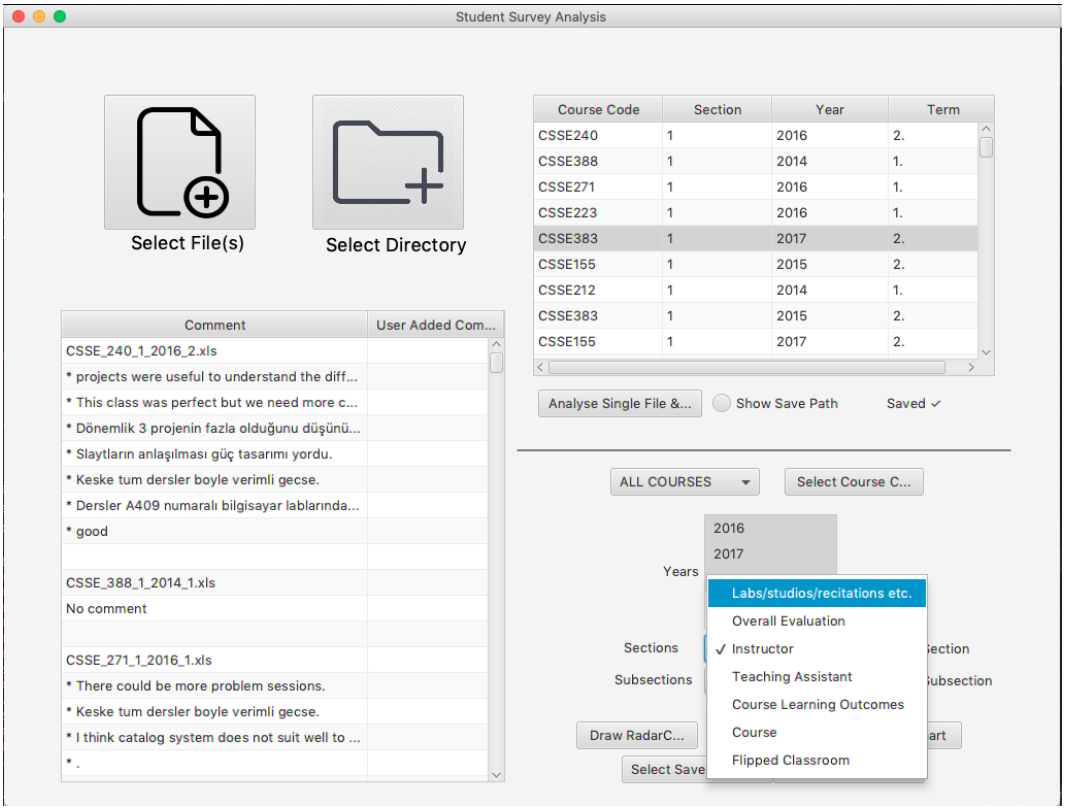

Figure 18: Selection of Wanted Section

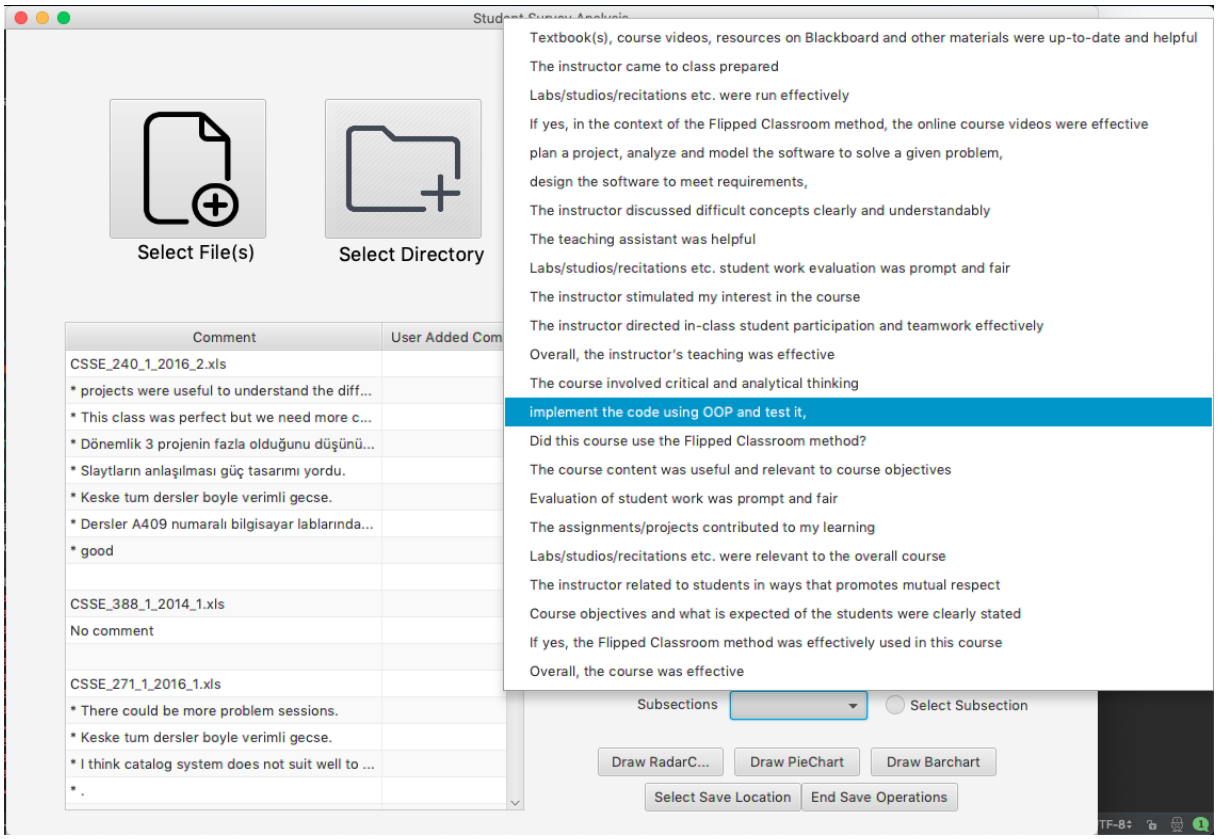

Figure 19: Selection of Wanted Subsection

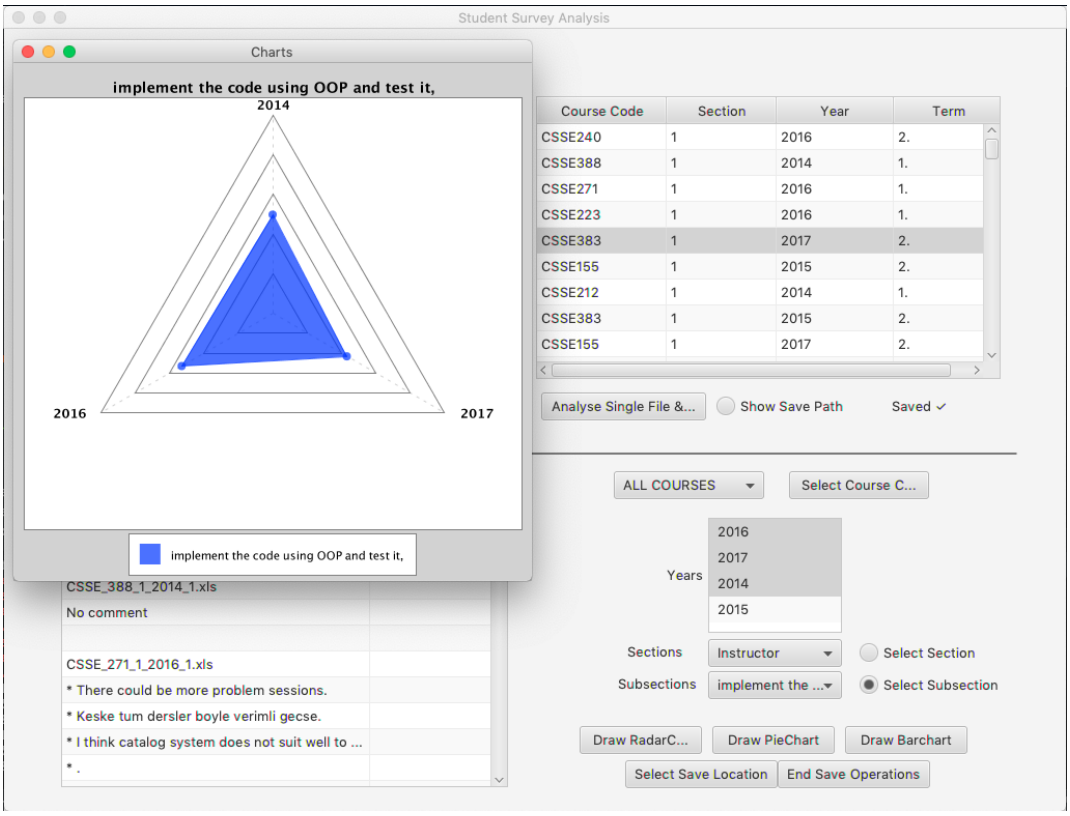

Figure 20: Sample Radar Chart from Program

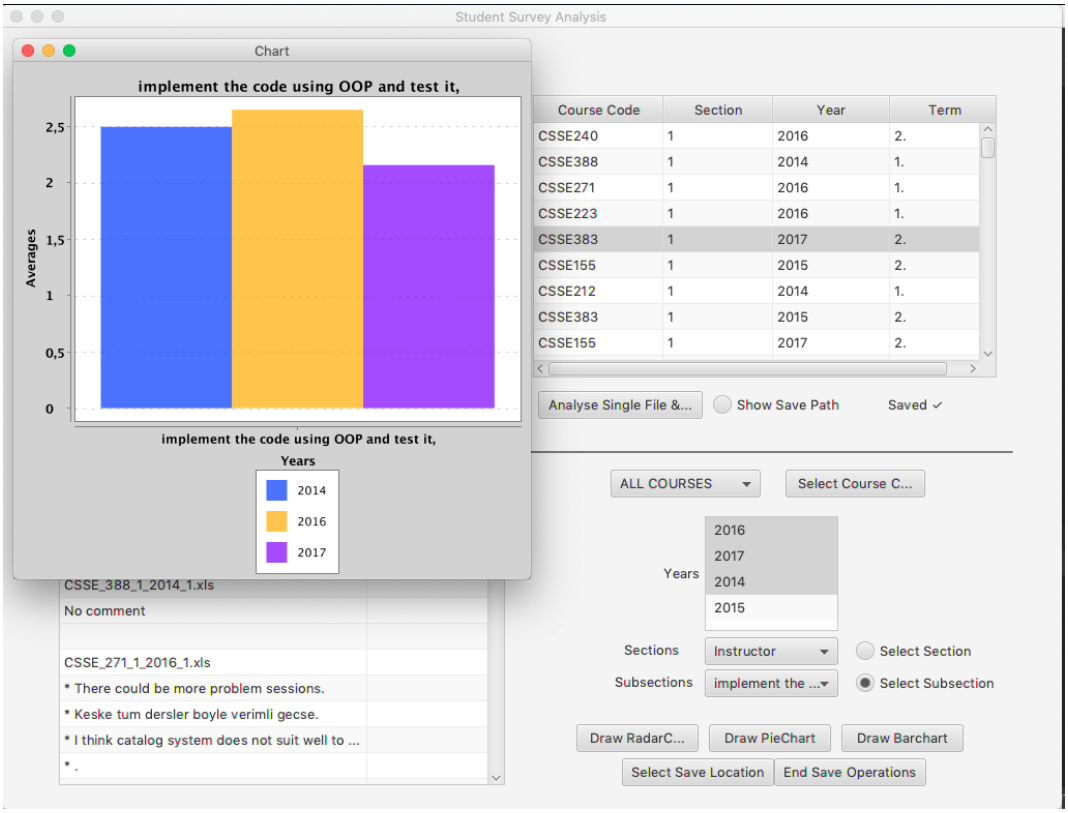

Figure 21: Sample Bar Chart from Program

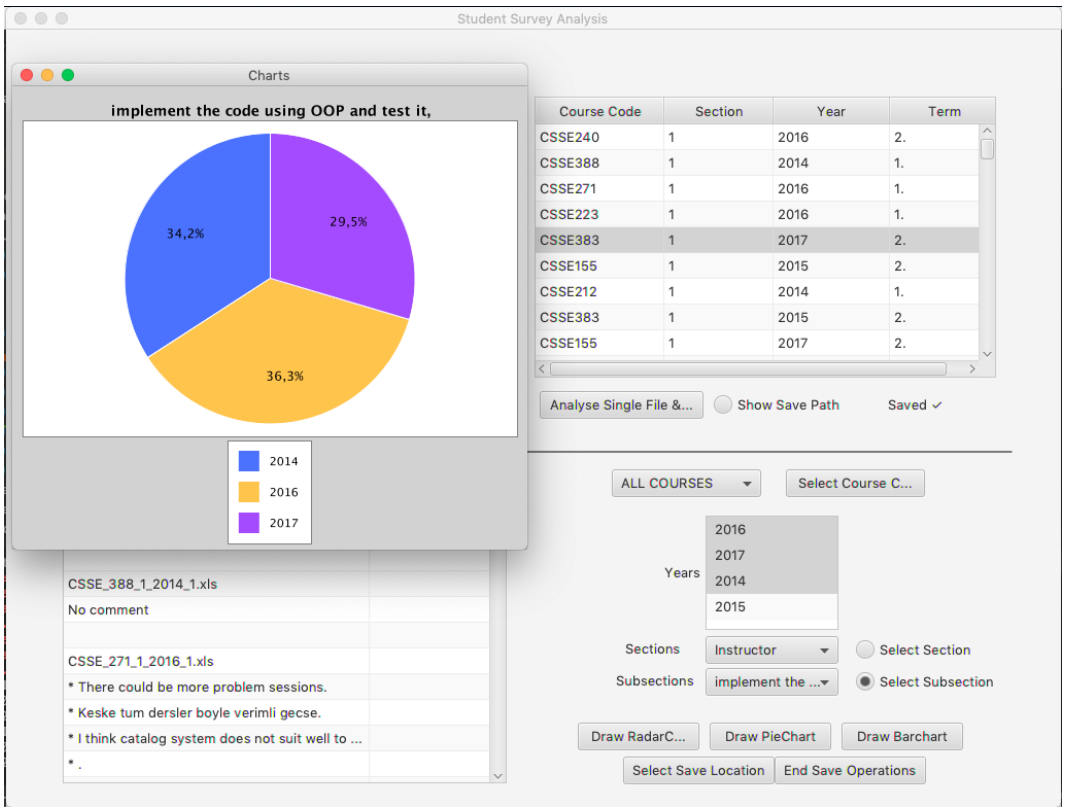

Figure 22: Sample Pie Chart from Program

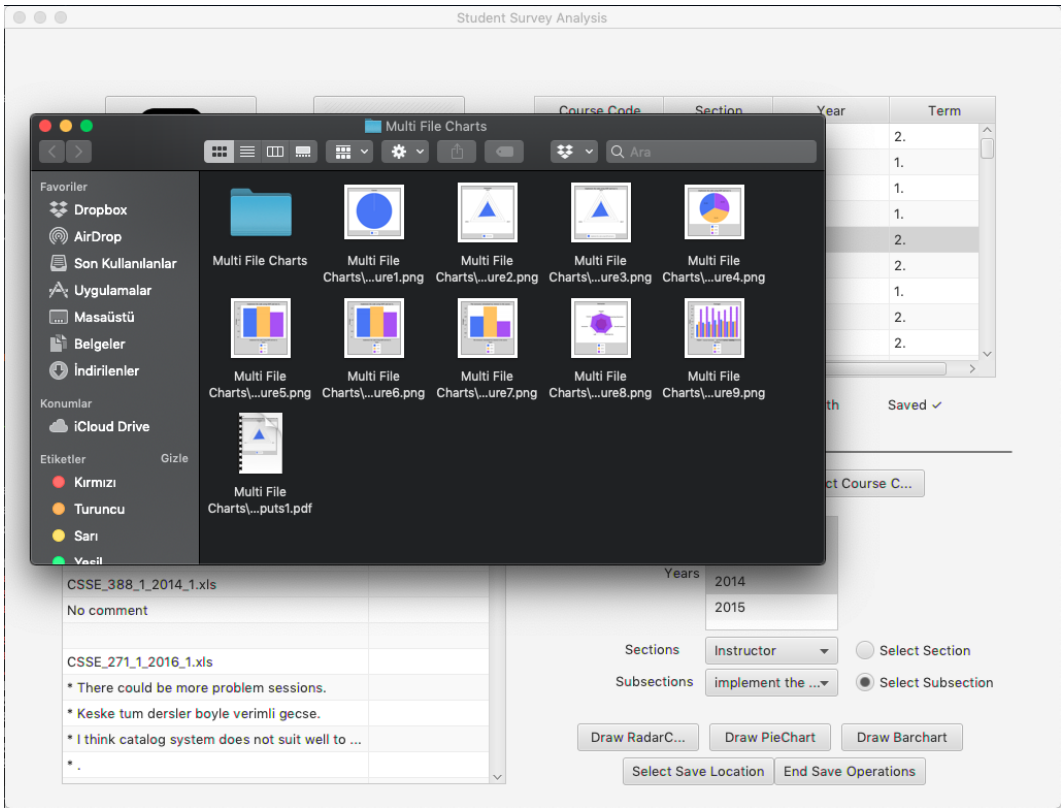

Figure 23: Multi File Charts Saved to Wanted Directory

The contribution of this project

In this project we learned how to analyse the big data and get a significant information from it. Also, we had learnt group work. That was very fruitful to develop a project with group. And we use github to share our work each other. Now we a familiar with github that very useful for us. We learned how to write most general and efficient algorithms with less work. At the and of the project it was very nice to have program which can be used by everyone.

References

1. https://poi.apache.org
2. https://knowm.org/open-source/xchart/
3. https://gluonhq.com/products/scene-builder/
4. https://www3.ntu.edu.sg/home/ehchua/programming/java/j4a_gui.html