CSE 360 Final Project

By: Austin Wright

Attendance Logger using Java Swing

Requirements:

* Load and store a table of information from a CSV file
* Write a table of stored information to a CSV file
* Display a table of student information
* Display a scatter plot of attendance data from the student table

Diagram

Description automatically generated

Class Diagram:

Diagram

Description automatically generated

Testing

Given the high level of dependency of UI testing, I did not use many Junit test cases apart from making a “dummy” program to experiment with the API used in the model. The components use locally declared data sets within the main method of this program. This was purely created to familiarize myself with using JMenu and JFree Charts. Here is a screenshot of this dummy program.

Chart, scatter chart

Description automatically generated

jfreeChartTes

Program Screenshots

Text

Description automatically generatedGraphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, table

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Calendar

Description automatically generated

Table

Description automatically generated

Graphical user interface, application, Word

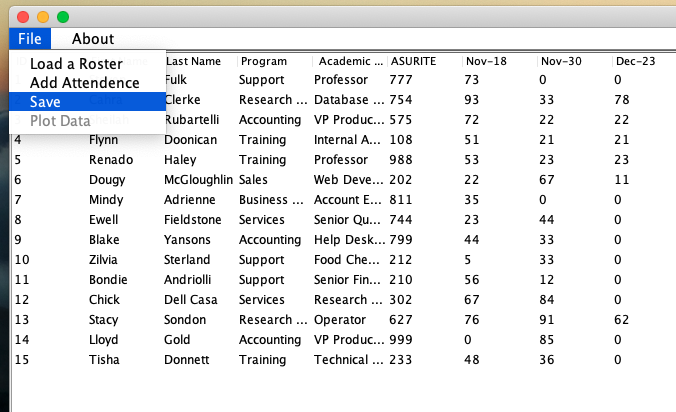
Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

Table

Description automatically generated



Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

Chart, scatter chart

Description automatically generated

Process

Since I completed the project by myself I choose the Agile software method since it meant I would prioritize getting a working program completed with flexibility. When starting out, I first facilized myself with the Maven dependencies by creating building a “dummy” program which had a split panel of a JTable with arbitrary data sets and a scatter plot of random coordinates. Next begun designing the program starting with the Main Application class to create a blank Split pane with a menu bar. I then created a decorative pattern with the table and scatter plot objects. I created instances of JFree Scatter Plot and JTable within each decorative child class of DataDecorative type. The DataDecorative type inherts from the DisplayableData class which provides access to the array of data displayed by the table and scatter plot. At this point when creating the individual classes I would leave them mostly un-implemented until they were needed to add the next level of functionality to the program. Given the skeleton of the program, I could now begin building functionality starting with adding action listeners to the Load Attendance menu item. This then lead to completing the Observer pattern by creating a function in the StudentListSource so the Observer can update the dataTable array with input read from the CSV file. Once I could print the dataTable, I begun adding functionality to the JTable which I felt was the most challenging. The JTable which uses the DefaultTableModel worked well with basic input but became more challenging when adding additional columns of attendance time data. If I was to go back and redesign this part I would have build my own table model so that searching for duplicate item would not require so much work for just a few methods in the DataTable class. Once the table could display both student information and attendance properly I went back to the mainApplication class to add functionality for Saving and building the Scatter Plot. Since the scatterplot uses XYCollections it required a few extra functions for correctly displaying the data. Apart from a few bugs where the JDatePicker hides behind the new attendance dialog box and failing to display data when an attendance file is used for loading a roster, I think the program runs smoothly despite the complexity of the model. One disadvantage with the agile process is that it can be hard to organize functionality properly since design elements tend to change as the model is being developed.