Maribel Rando

Bryce Konst

Caleb Jaramillo

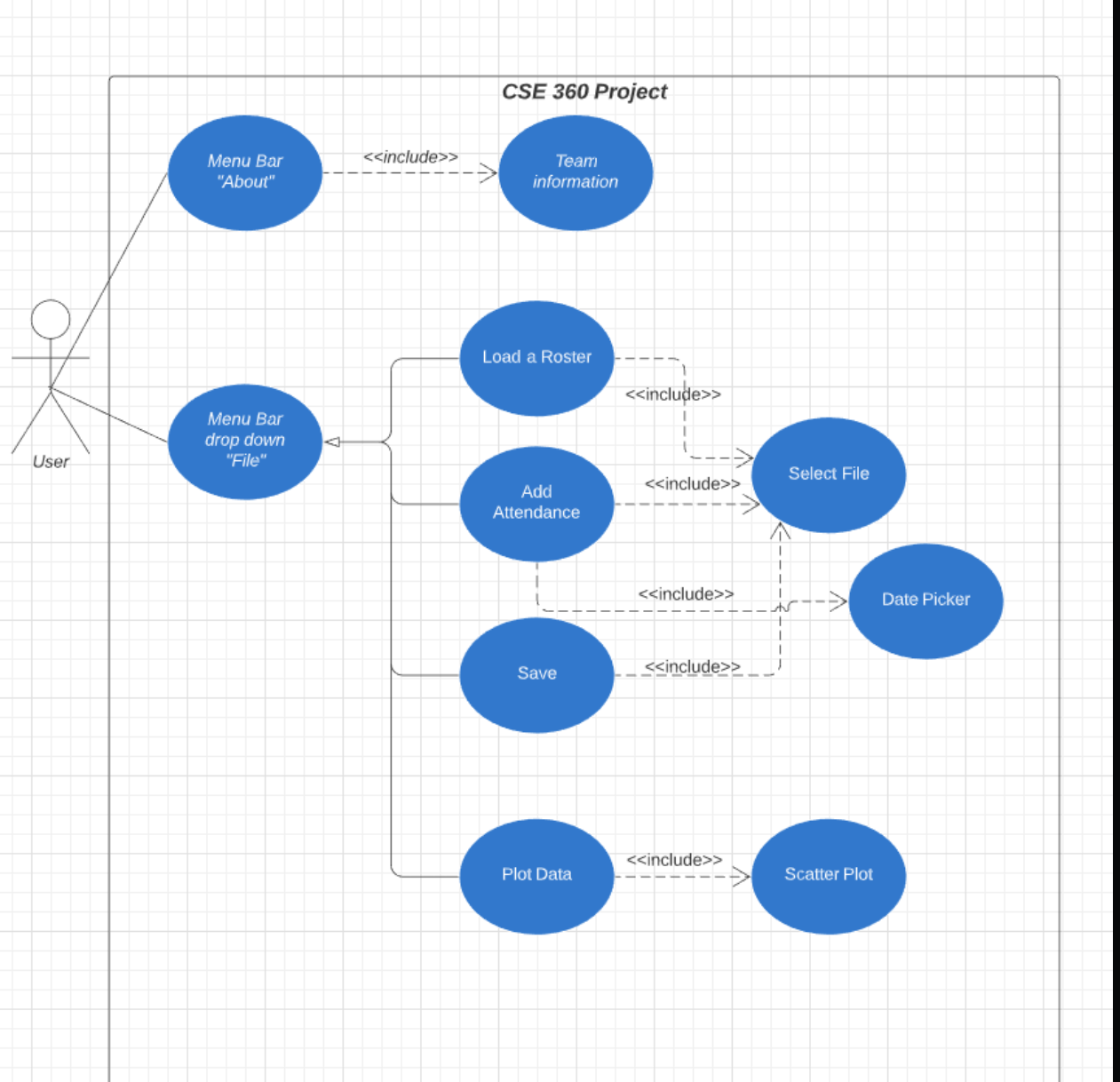
Samuel Trujillo

Yisha Fan

CSE360 - Javier Gonzalez-Sanchez

Semester Project Report

**Requirements:**



The user, through interacting with the menu drop down “About” selection should be able to read team information.

Through interaction with the menu drop down “File”, the user should be able to perform one of four actions at a time.

By choosing “Load a Roster” the user should be prompted with a File Chooser to choose a CSV file to load roster information from. The information loaded from the File Chooser is then used to create a table with 6 columns initially, and a row for every line in the CSV file.

Should the “Add Attendance” option be selected, the user will again be prompted to pick a file from the FIle Chooser. The file selected will be read, and its contents will be used to fill in a new column added at the end of the table. This function of the application will allow users to show the minutes a student attended a class on a certain date. The date should be obtained through user input by way of a Date Picker. If students not in the roster would appear in the attendance file, they will be displayed in a separate dialogue box.

Selecting the “Save” option will save the information that the user has compiled in the application into a CSV file. This option will also prompt the user to select a file from File Chooser to save the table information into.  
 Finally, the “Plot Data” option will read the table information and create a scatter plot based on attendance statistics. The X-axis is the percentage of attendance per day, with 75 minutes being the threshold for 100% attendance. Any excess minutes are disregarded. The Y-axis is the number of occurrences of each length of attendance for each day, with the graph being able to plot information for n number of days. This function of the application will utilize Scatter Plot from JFreeChart.

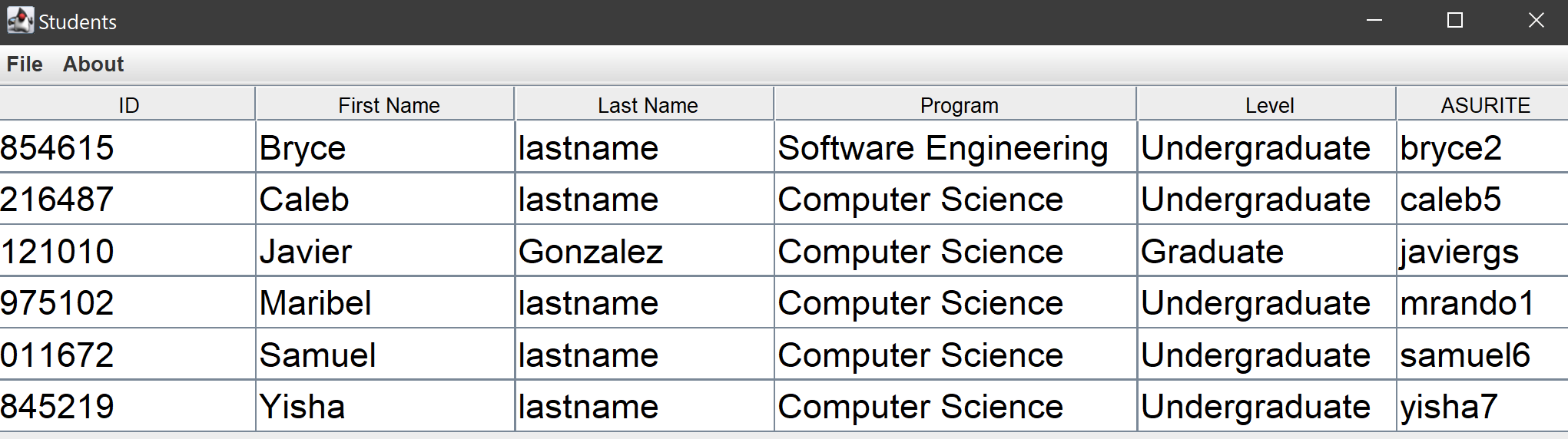
**Testing:**

Unit testing using JUnit was not used in the development of this application. Because this application is centered around a user interacting with a Graphical User Interface (GUI), the outputs are very easy to see and decipher. Furthermore, while a JUnit test could very easily check data outputs from certain computations, this application does not require any computations more complex than simple addition and division. A JUnit test also could not tell how the application looked visually, and presentation through a GUI is very important.

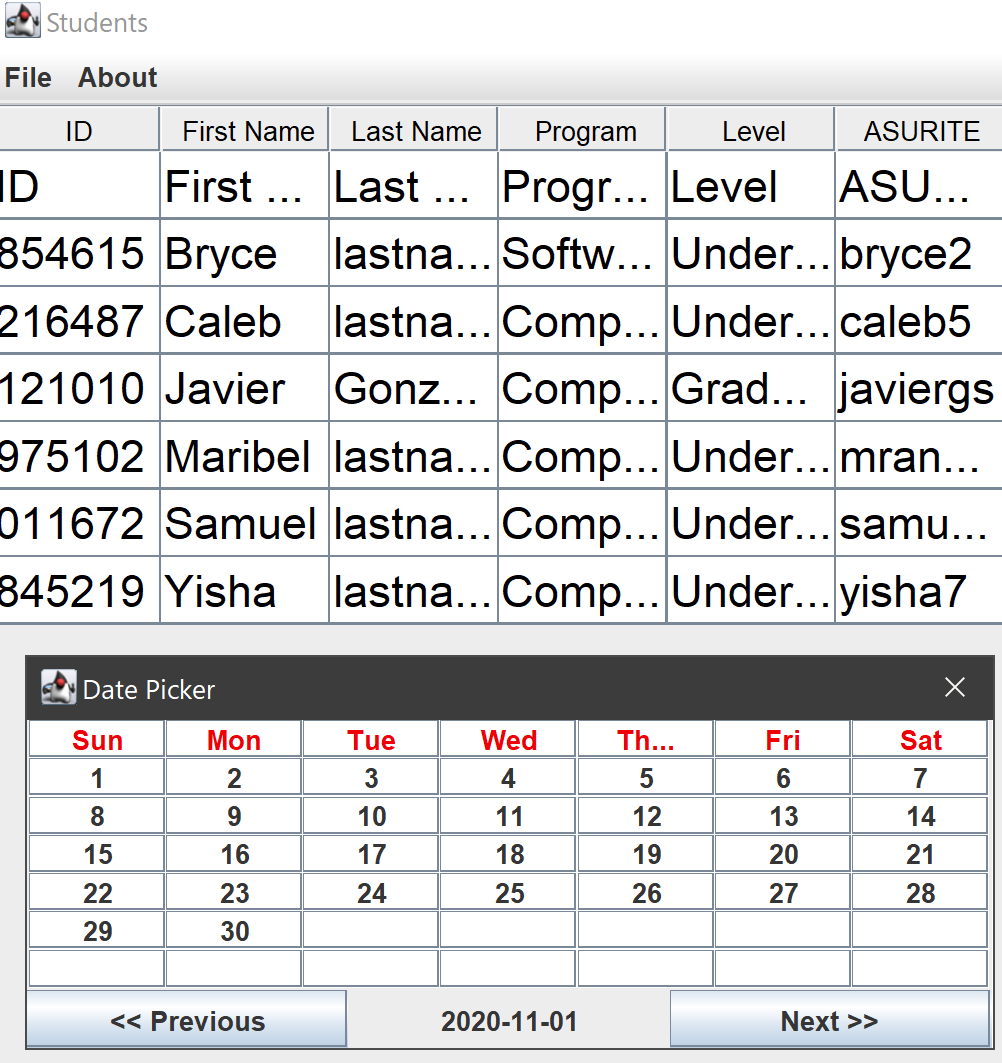
Instead testing was composed of making sample CSV roster and attendance files and loading them into the application manually to see if the program did what it was intended to do. While this process may seem more tedious, it allowed for testers to judge and validate the application based on its visual aspects alongside the expected data outputs. This allowed for insight to be gained on the sizing and style of many elements of the GUI, and eventually changes to be made so that the GUI was presented in a more pleasing and professional manner. Through this along with verification testing, we discovered potential errors or defects that would occur.

The following figures below show the various testing results.

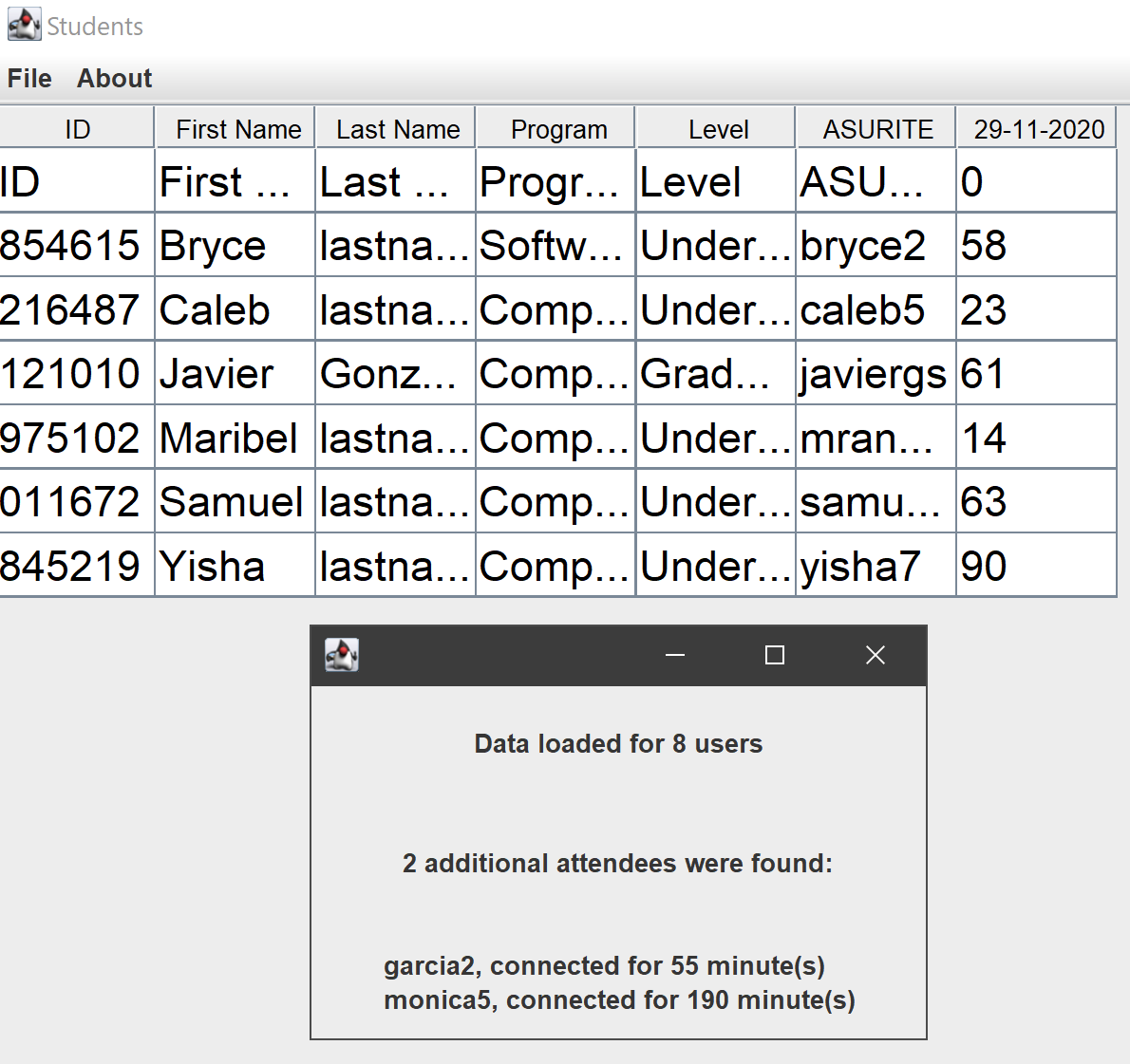
**Fig (1): Roster Test**



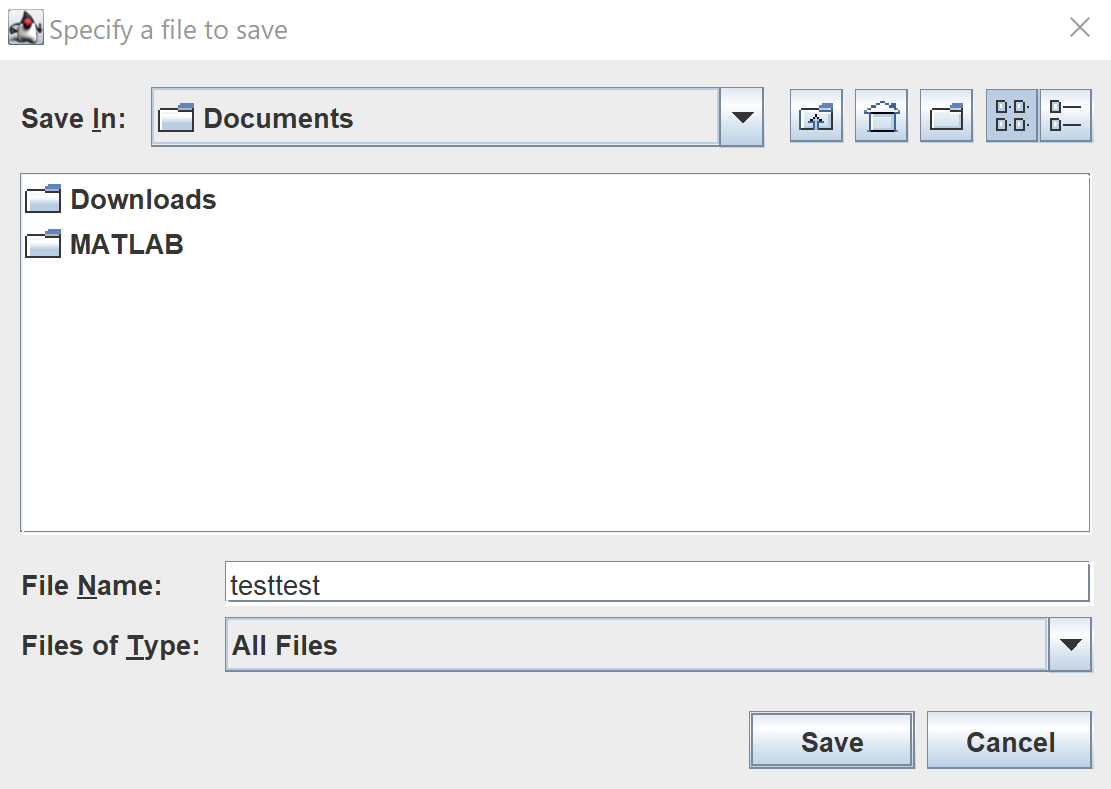
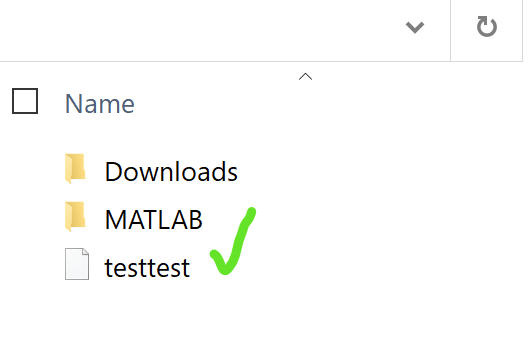
**Figure (2a): Roster + Attendance Test**



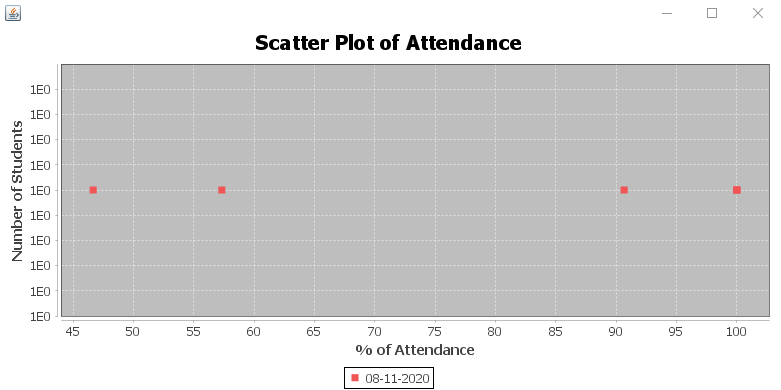
**Figure (2b): Roster + Attendance Test**



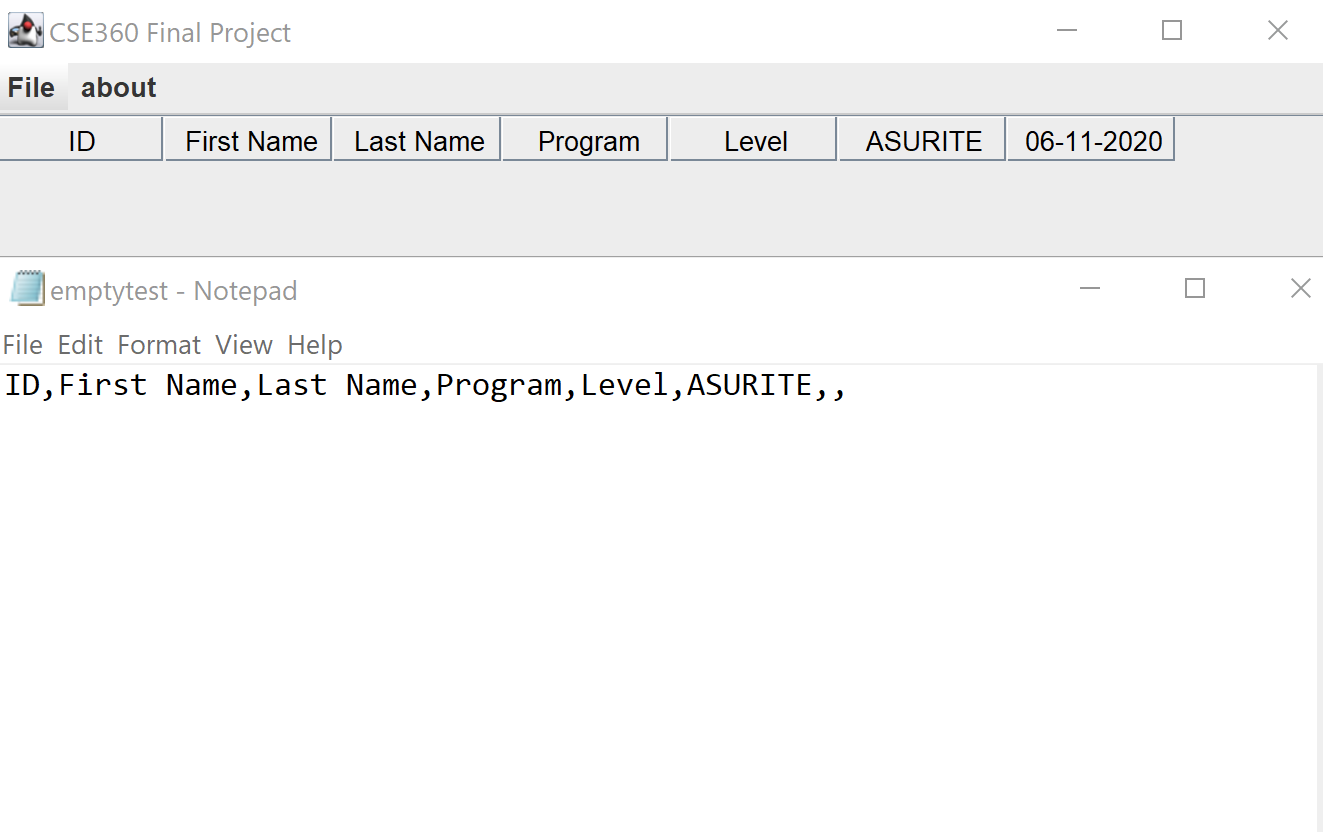
**Figure (3): Add Attendance + Save**

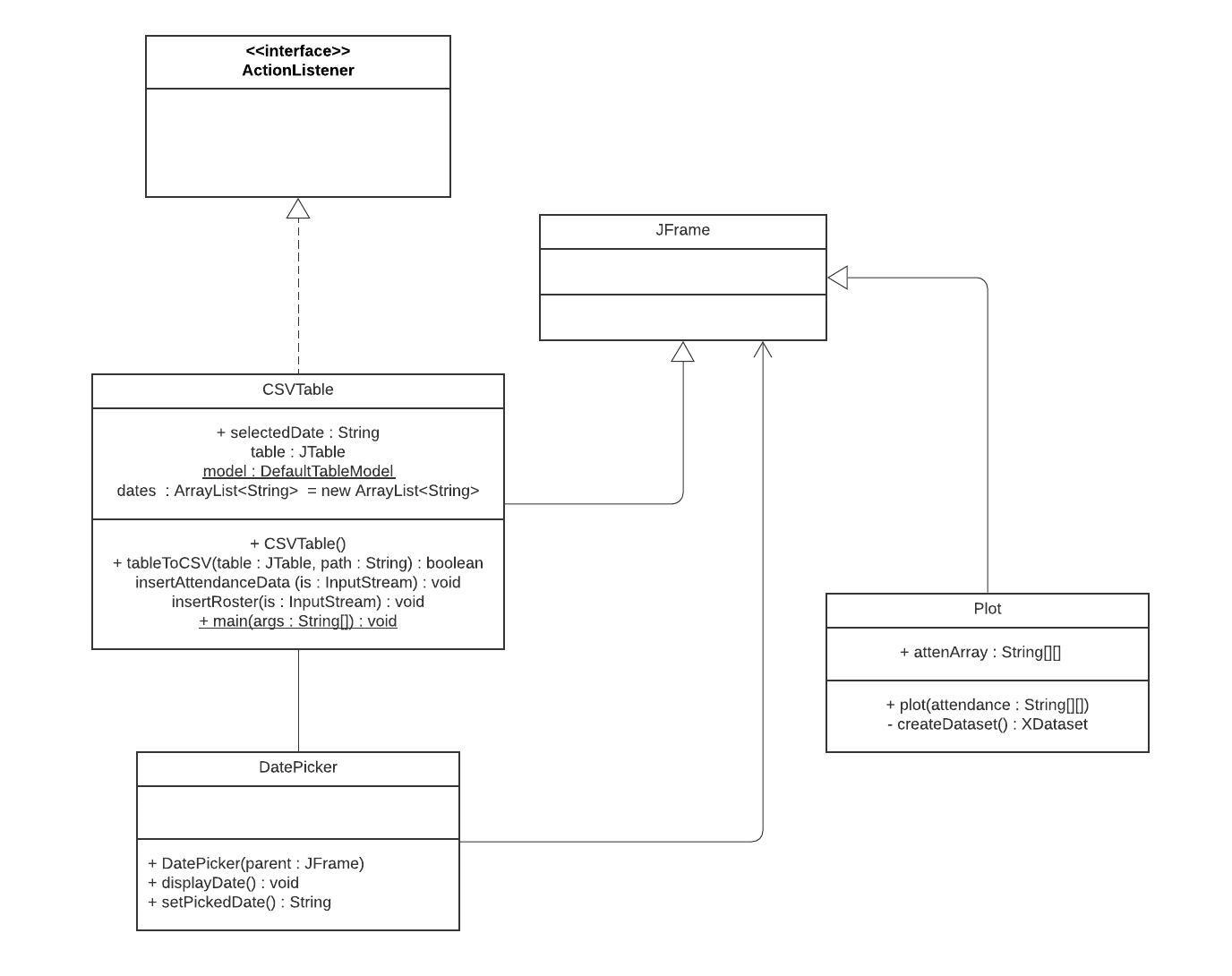
**Figure (4): Scatter Plot**



**Figure (5): Empty File**

****

**UML Class Diagram**



**Project Management:**

The process model utilized for the development of this application was the Waterfall Model. The Waterfall Model was chosen for a few reasons: there was a hard deadline that had to be met, the process of communication, planning and modeling before construction, and finally the fact that there was unlikely to be any changes to specification after the process was underway.

Through the development of the project, the team met twice a week, once on Tuesdays, once on Fridays, to discuss progress made and further responsibilities that needed to be taken. The general process of splitting up the responsibilities was gaining a comprehensive understanding of what needed to be done, then members would volunteer to take on responsibilities. The twice weekly meetings also allowed for all group members to monitor each others’ progress and modify responsibilities accordingly. Below is a list of all the responsibilities each member took:

Caleb Jaramillo: Creating the group, drawing PERT Chart, building a list of what needed to be done each week and facilitating the splitting of responsibilities, making sure the GUI looked pleasing and professional (sizing and menu drop downs), bug fixes, writing the text for the report.

Bryce Konst: Creating the CSVTable, creating the “Save” and “Load Roster” functionalities, helping other group members understand how to use the Table for other functionalities, creating the use case diagram.

Maribel Rando: Created sample CSV files for “Load Roster” and “Add Attendance”, testing, creating the class diagram, gathering screenshots of program running.

Samuel Trujillo: Creating the “Add Attendance” functionality, implementing Date Picker.

Yisha Fan: Helped recruit team members, creating the plot data functionality, figuring out JFreeChart.

**PERT Chart:**

