Tyler Ardito

Implementation and description of code.

For the Student.m Class, I first created the class with classdef, then made the properties all of the things the students need. ID, name, age, GPA, and major. Then in the methods, I connected the currentOBJ. _____ = _____ where the blank is each one of the properties equal to itself. This means that when student 1 is called/created. The object is specific to that student. Ex. The name Tyler is only for student 2. Then it shows the student info by displaying each student's information but separated so all the students "objects" are all together. Ex. Student 1's id. Student 1's name, student 1's GPA. Etc. Then we add a function to find a student based on their id. When we type in the id into the main.m it will then show the information for that student. If that id # doesn't exist, then it will display an error message that says student not found. The reason we type it into the main.m is because it does not ask for inputs in the file, since we need to show how to create students. The final method in the Student.m is to update the student GPA. You need to add the new GPA and the id of the student whose GPA you want to change.

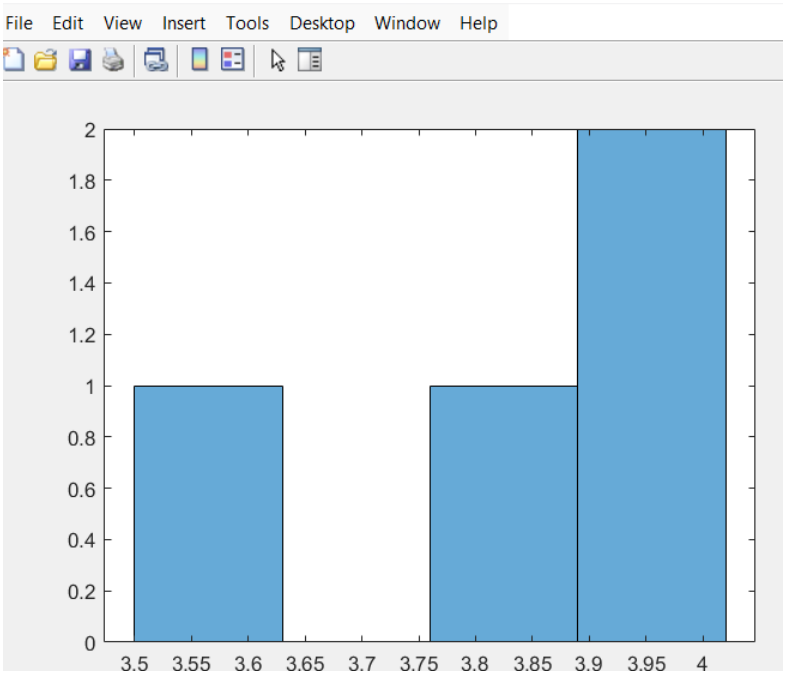
For the StudentDB.m, I first had to create the StudentDB class, then create set the size to 20, student list as an array, and number of users to start at 0. I used the methods we learned in class to create the initial database, and then I made a function to add students to the database. The create user actually create the students for the add students function to add them to the database. Show students will then display all of the student info, then will list them by *******Student x *****. We then have another way to find students, since it was in the practice we did in class, so if there were any important variables listed here, I did not want to miss typing them in on the Midterm code. Finally, I made a function to save the database to a new file.

Instructions for running the code:

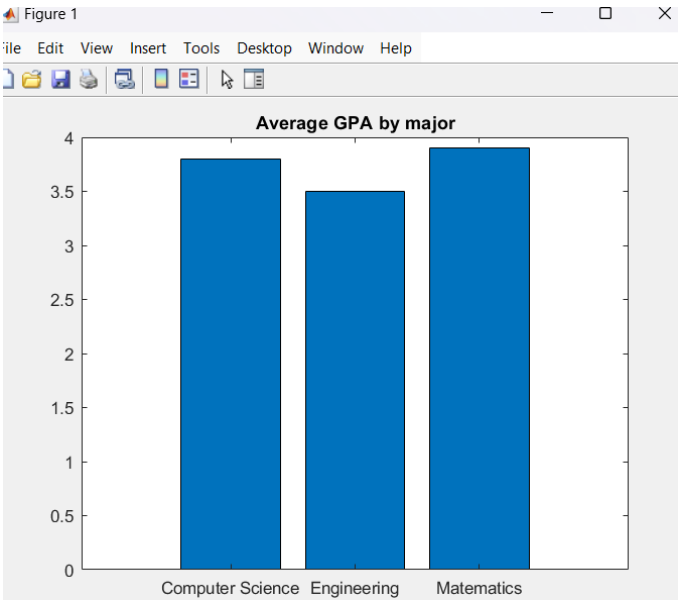
To run the code. You need to type in the student information into create user, and that will add it to the database. Then it will print the students, you then will type In the id of the student that you would like to search. The main.m will also create the 3 graphs shown below under visualizations. The reason the code does not ask for an input of information is because we need to show student creation and examples in the main.m. To run each individual visualization, you will need to comment out the other ones. I could not get MatLab to individually show each graph. In the main.m, I commented out the show-Students function since it was too much to put on the screen.

Visualizations:

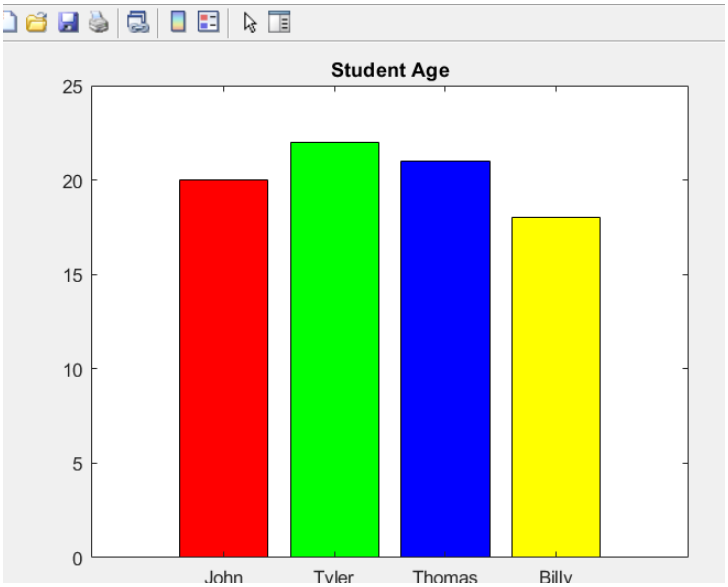
Histogram



GPA by major:



Distribution of student ages:


Student Age
25
20
15
10
5
0
John
Tyler
Thomas
Billy