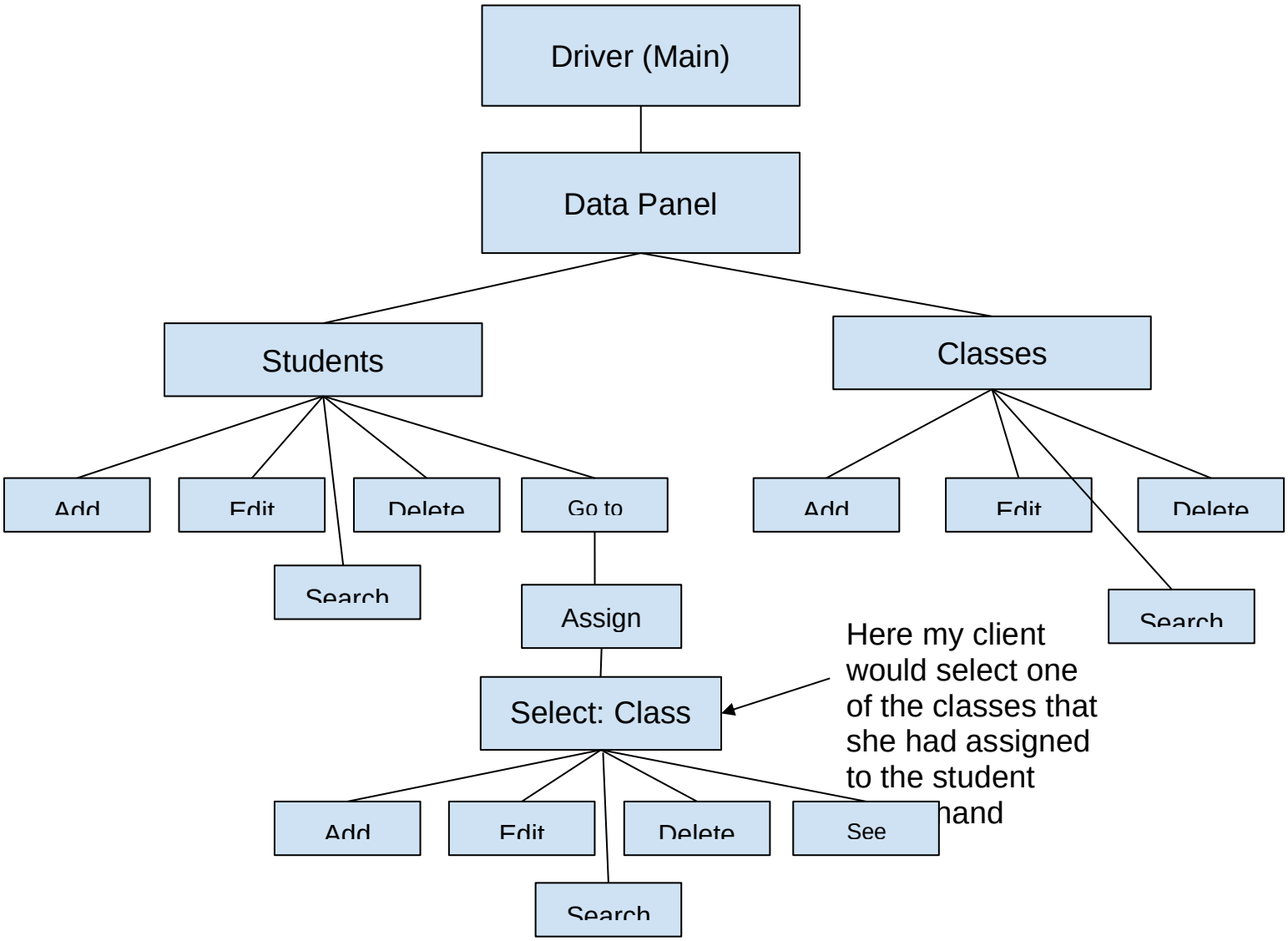
Criterion B: Design

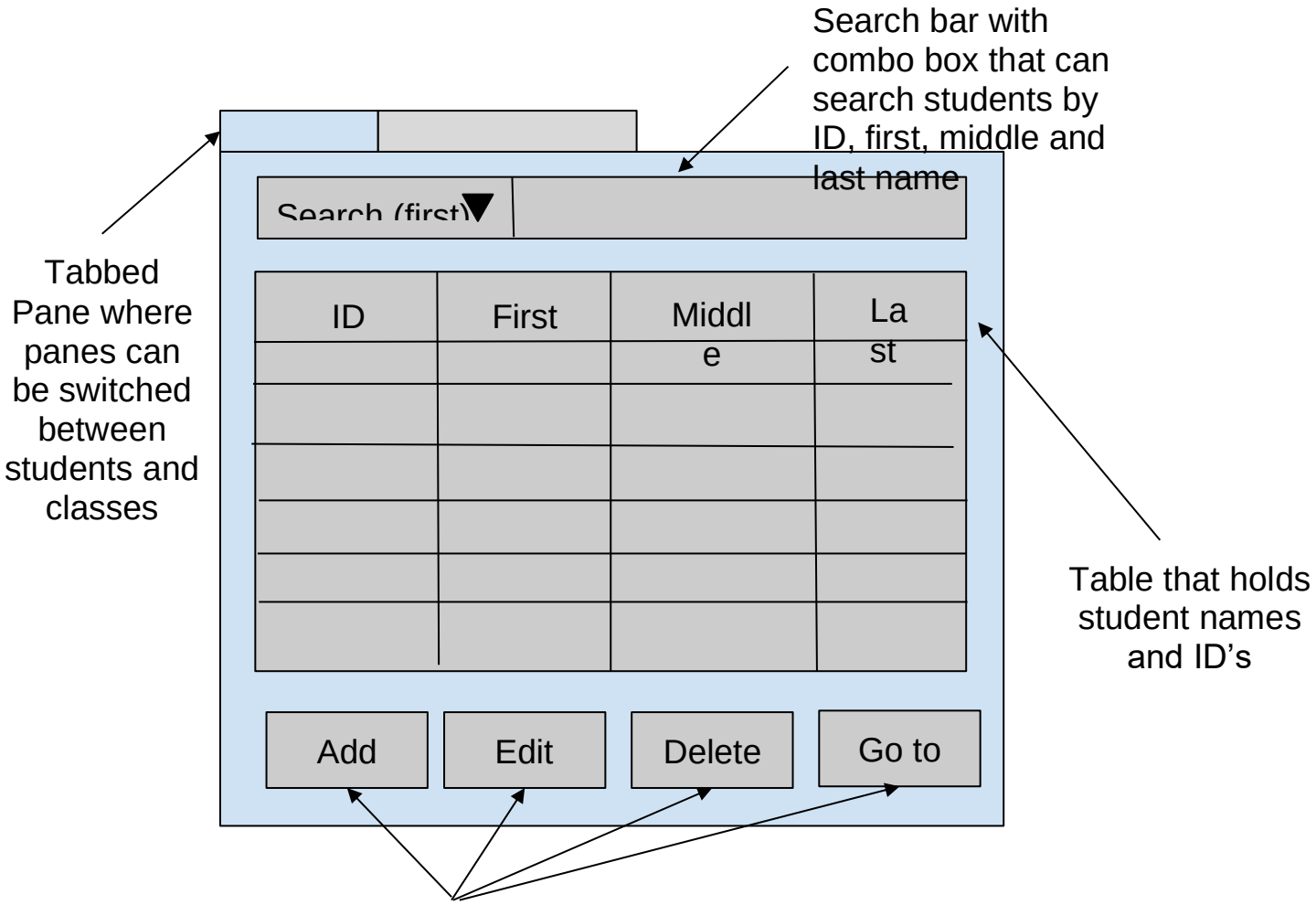
Program Outline:



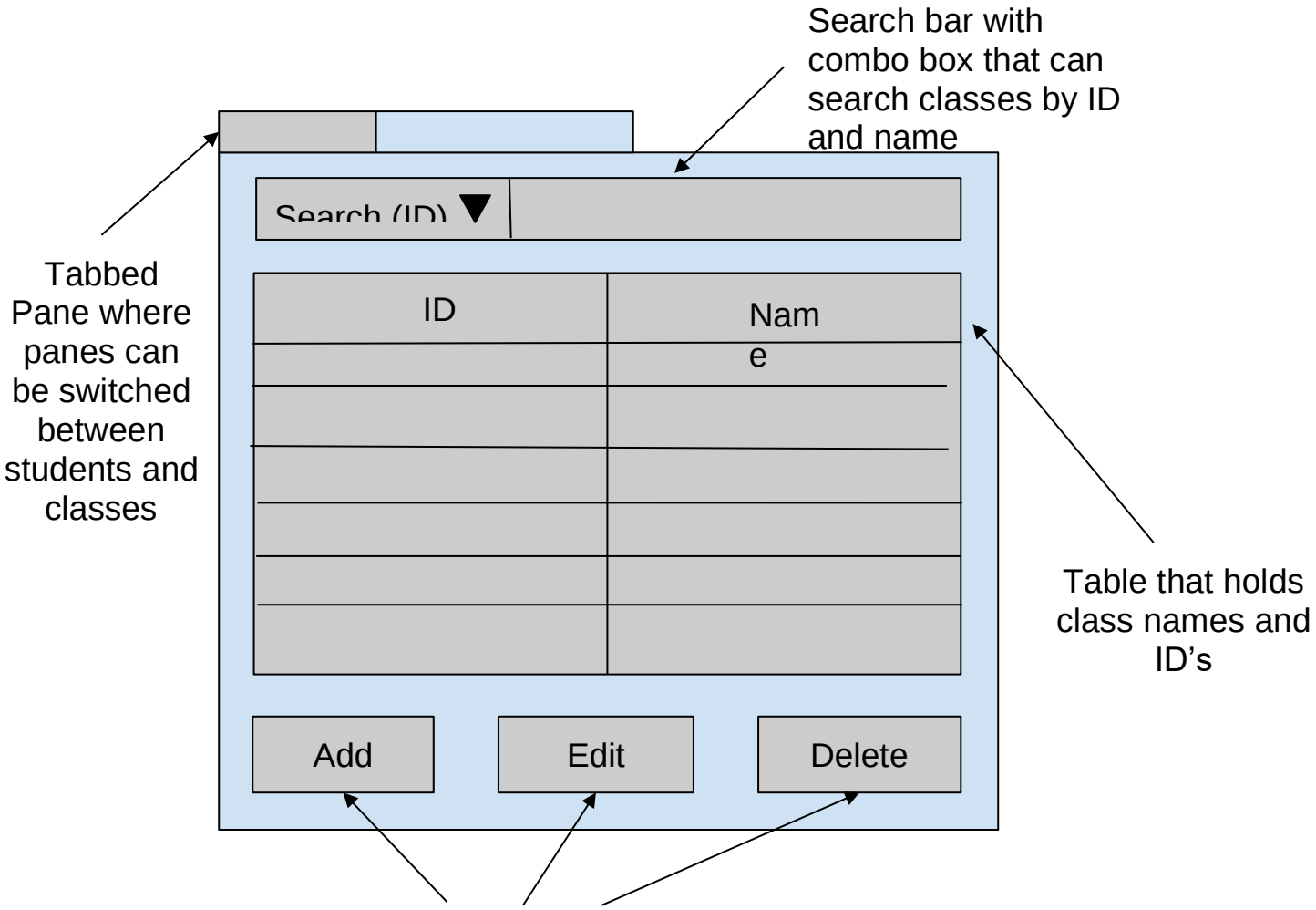
Criterion B: Design

Panel Designs:

Main Menu for students:

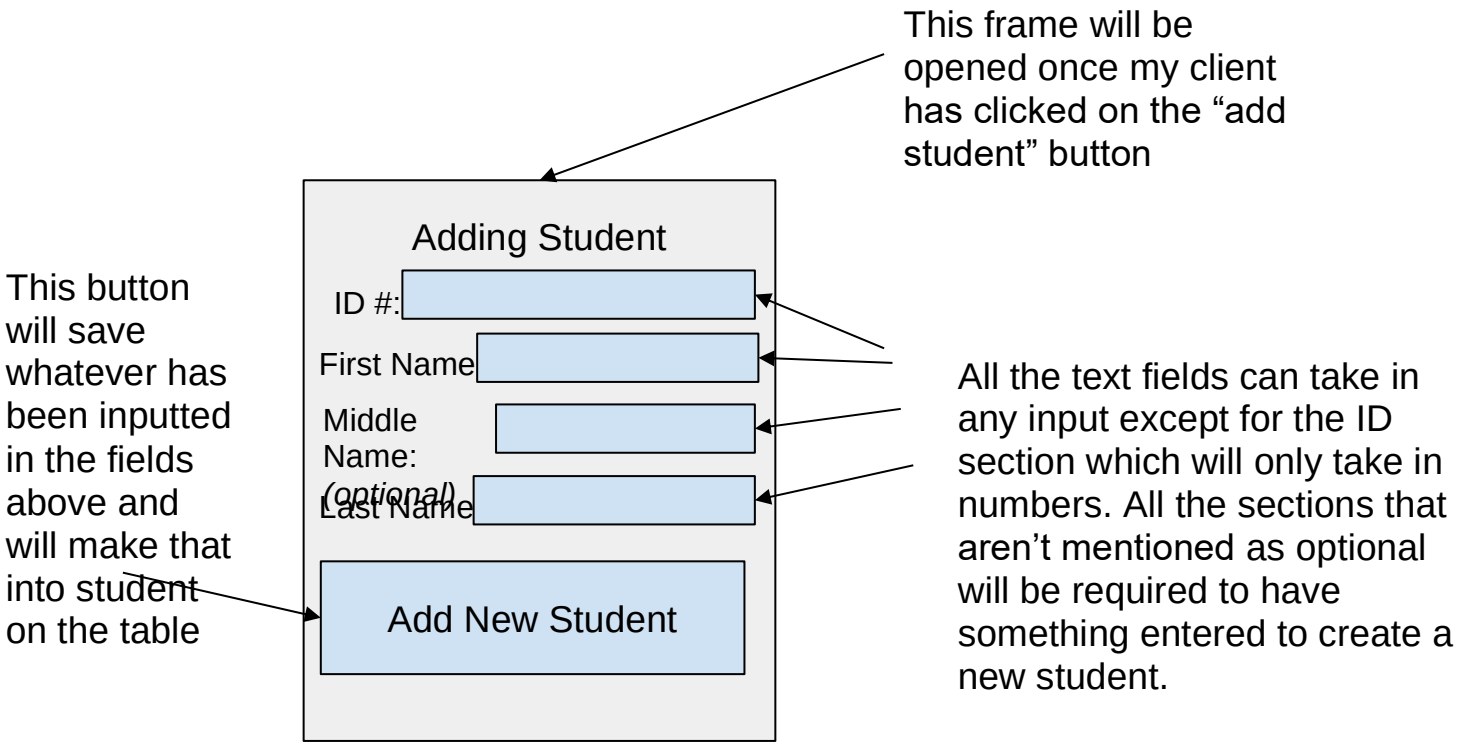


Main menu for Classes:



Criterion B: Design

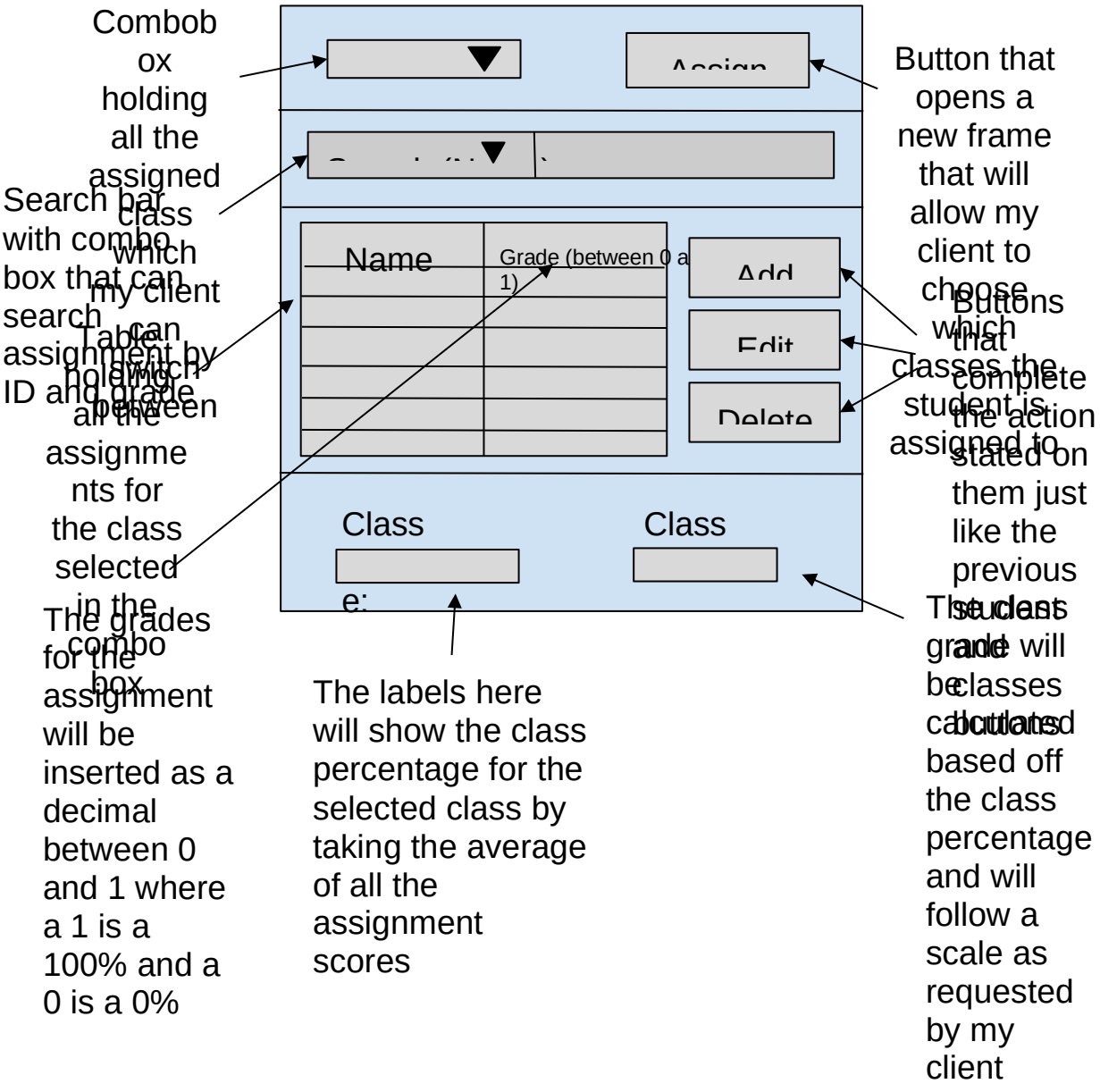
Adding/Editing Menu (example of adding student menu below):



My program will also include an editing students menu, which will have a panel similar to the one above except that it will pre-fill the text fields with the student’s information, and the button name will be changed to “Save Changes.” Just as there’s an adding and editing students menu, those menus will also be made for classes and assignments as well which will follow the same format, but accommodate for the columns that are present in each.

Criterion B: Design

Student Page Menu:



Scale of class grades for student page menu:

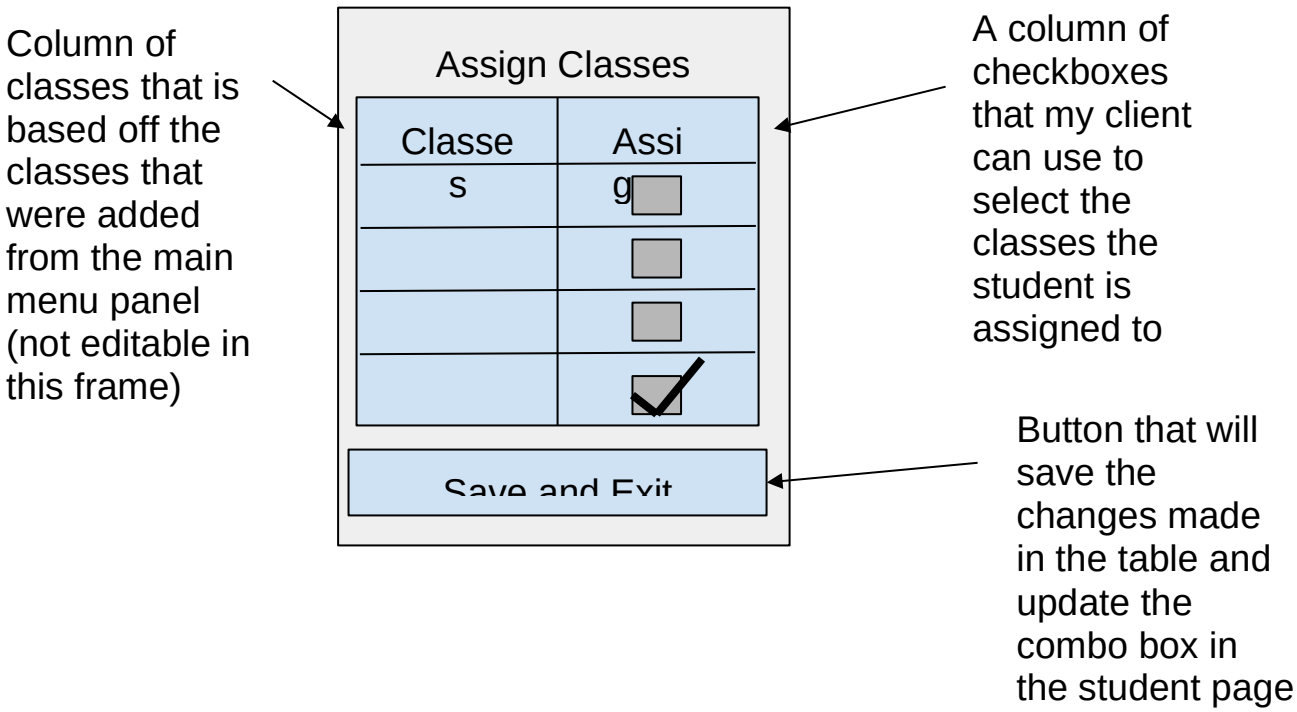
Percentage Range	Grade
100% - 90%	A
89% - 80%	B
79% - 70%	C
69 - 60%	D

Criterion B: Design

59% or below	F

Each student that is inserted into the program will be able to have their own student page in which my client can customize the classes that they are assigned along with adding the assignments and grades they've received.

Assign Classes Menu:



This frame is accessed by clicking on the "assign classes to student button" from the student page. Here my client will be able to select the classes she wants to assign to the student or if she had previously assigned students before, will show the classes that have already been assigned.

Criterion B: Design

UML of Product:

Criterion B: Design

Overview of classes:

SQLite:
Here I will have most of my sqlite commands in a single class that I can reference through my program.

Driver:
This is where the DataPanel will be made which is the main panel of my program with the class including the main method.

DataPanel:
This class will make the tabbed pane from the first two panel designs with everything inside.

All “Adding” or “Editing” Frame classes:
The purpose of all these classes is to make their respective frames and save the changes they made to the database from what was entered in the text fields and update the respective tables. These classes generally have the same type of instance variables but are just dependent on what JTables they are working for. And the main difference between the adding and editing is the inclusion of getters and setters for the editing class.

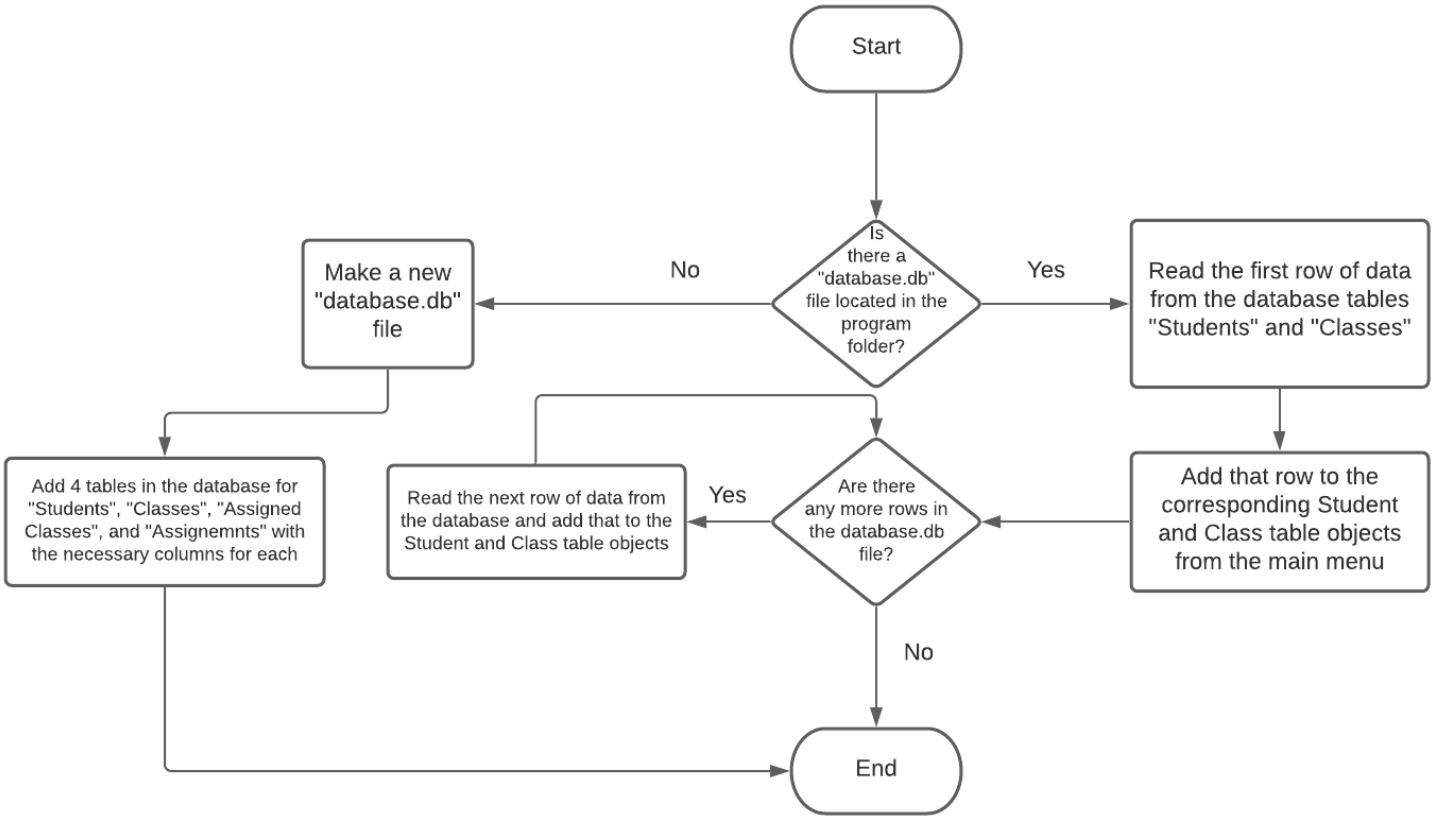
StudentPageFrame:
This class will make a fully functioning student page menu where my client can see that individual’s assignments and grades.

AssigningClassesFrame:
This class will make the assigned classes menu that my client can interact with to assign or remove classes from a student.

Criterion B: Design

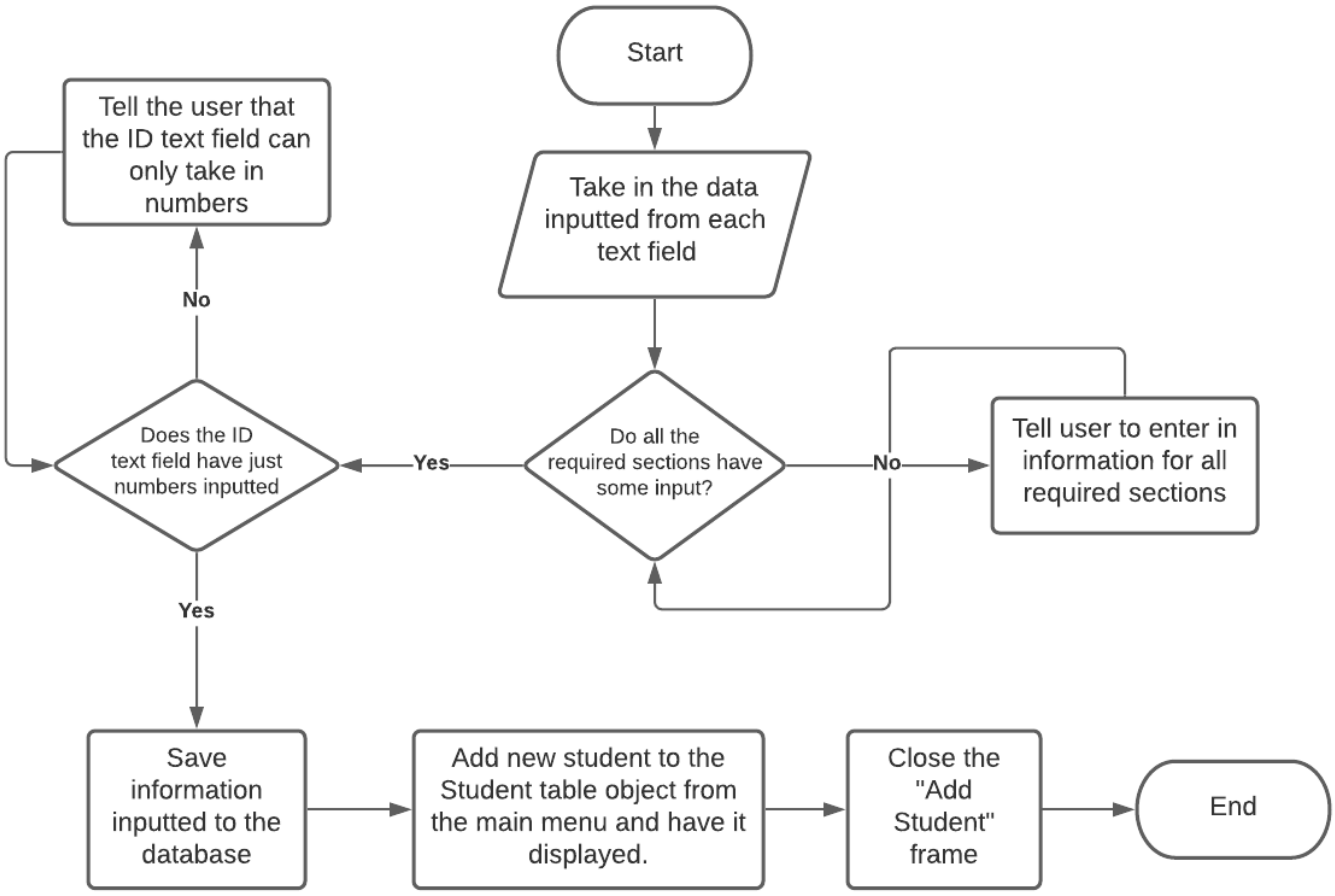
Flowcharts:

The program first starting up (database initialization):



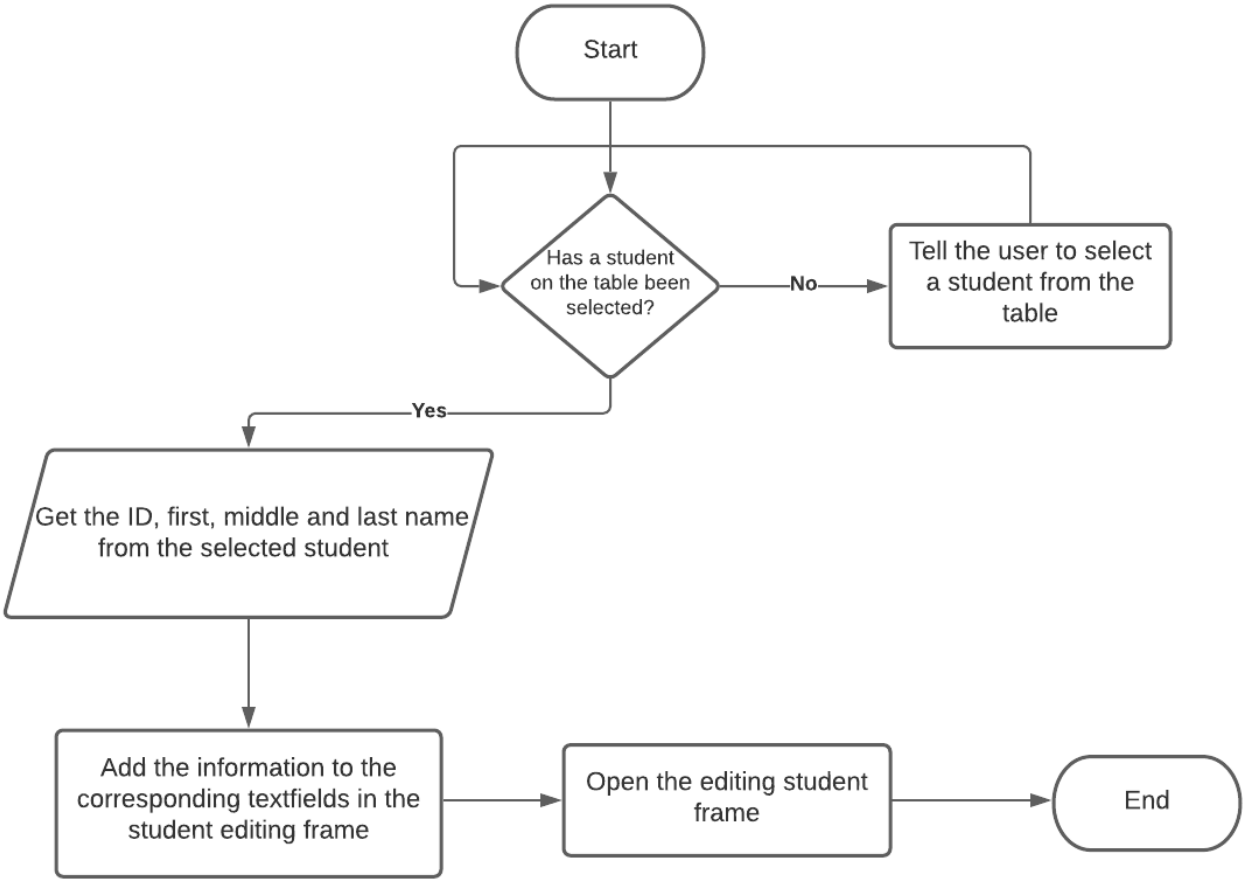
Criterion B: Design

Adding students to program once the “Add New Student” button has been clicked:



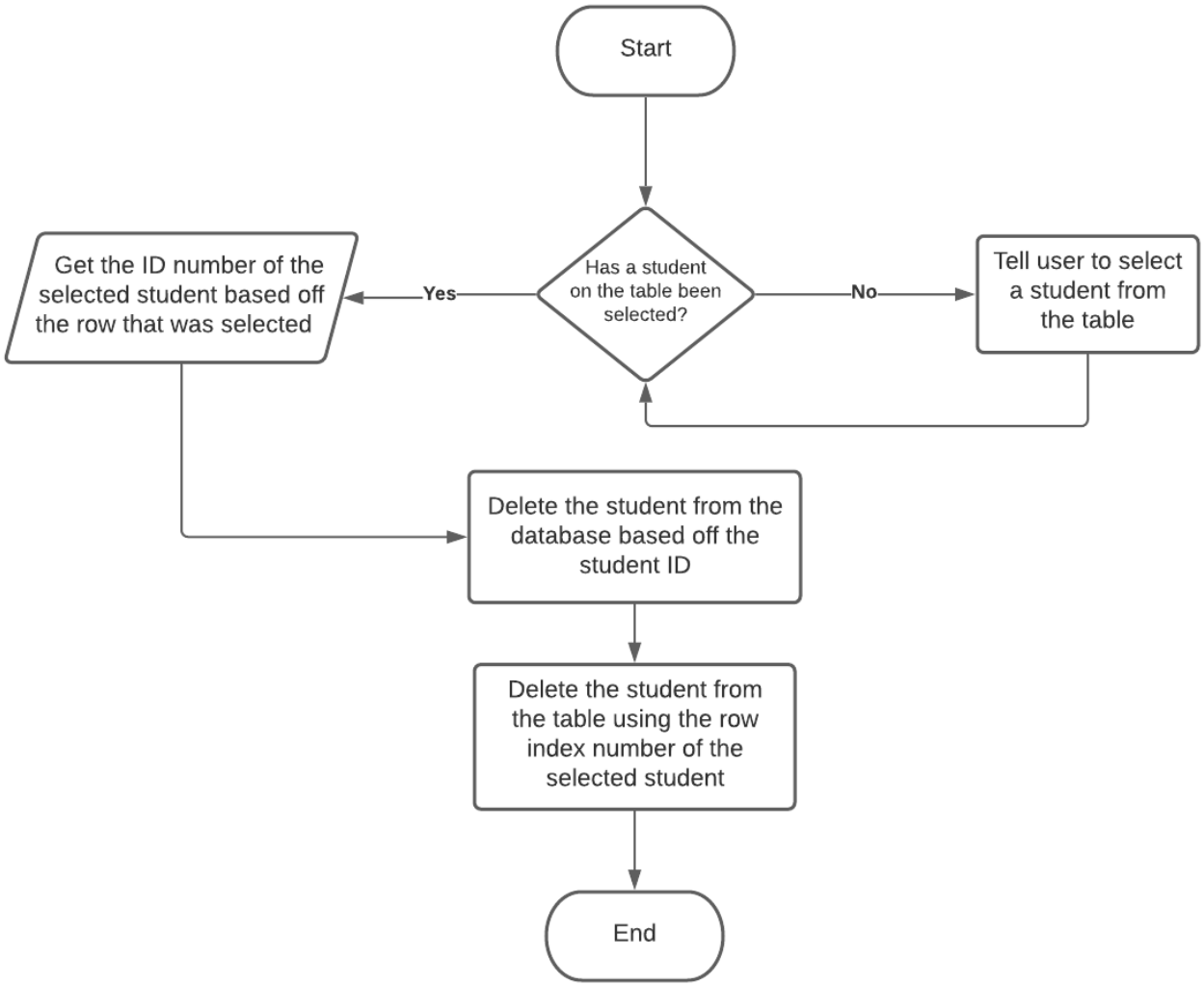
Criterion B: Design

Opening the editing student frame when the “Edit Student” button has been pressed:



Criterion B: Design

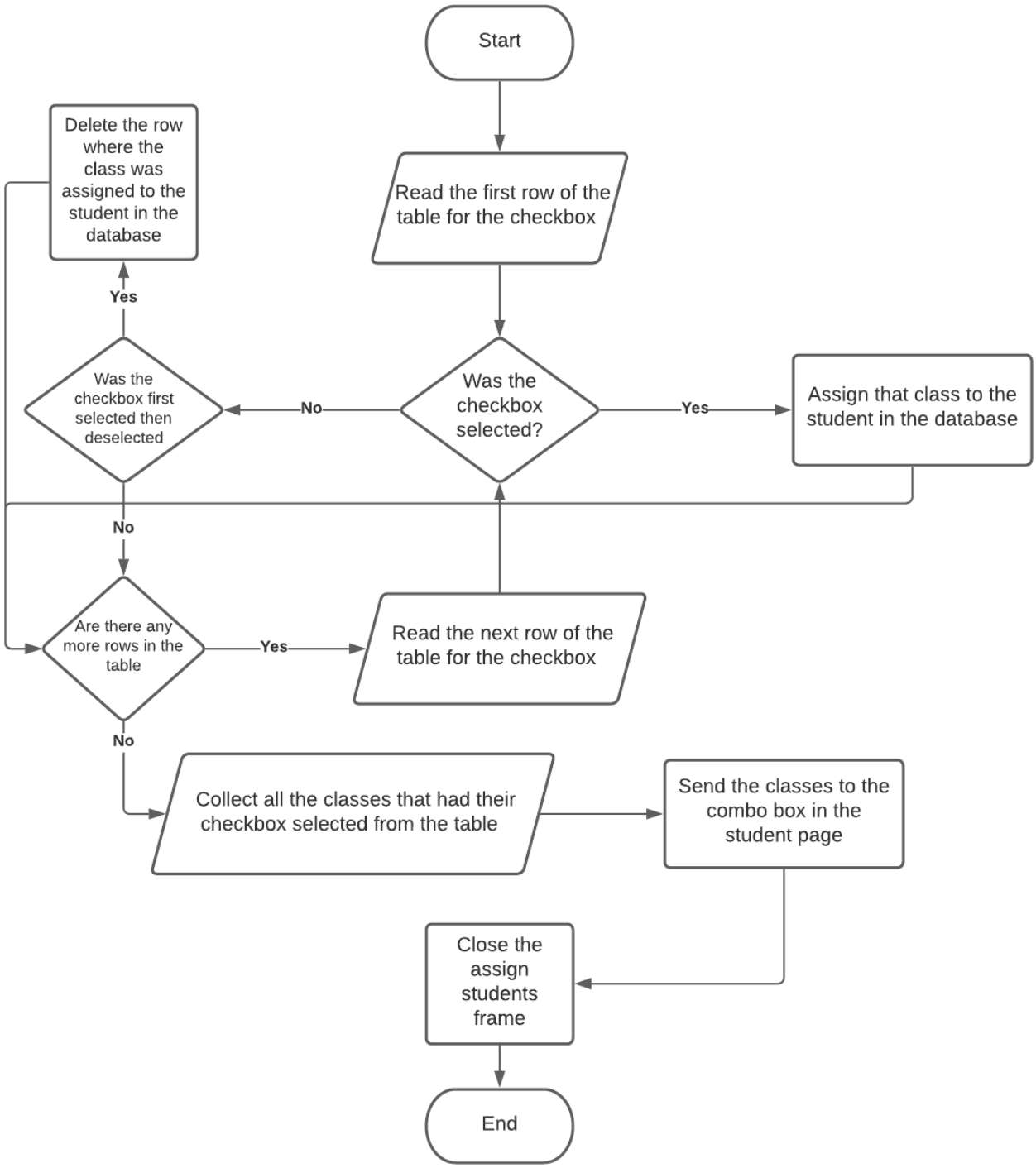
Deleting students from the program once the “Delete Student” button has been clicked:



Although the last three flow charts only show some of the adding, editing, and deleting for students, the classes and assignments follow a similar pattern, accommodating once again for the columns that are present. The assignments will also have to keep track of what student and class they are a part of when adding, editing, or deleting it from the database.

Criterion B: Design

Assigning classes to the student once the “Save and Exit” button is clicked:



Criterion B: Design

Testing Plan:

Action to Test	Test Method
Program can successfully create database	Check the files of the program to make sure that a "database.db" file is created
A graphical user interface that properly represents all the information	Check all the information from the database if being presented in the GUI. Test all the buttons and features in the GUI to make sure that they are successfully working.
Program can add, edit and delete students	Create a dummy student and see if it appears in the student table of the program. Edit dummy students to see if changes are made in the table. Delete dummy student to check if it was deleted from the table. Use an external program (DB Browser for SQLite) that can open database files to check if it was successfully added, edited, and deleted.
Program can add, edit and delete classes	Create a dummy class and see if it appears in the class table of the program. Edit dummy class to see if changes are made in the table. Delete dummy class to check if it was deleted from the table. Use an external program (DB Browser for SQLite) that can open database files to check if it was successfully added, edited, and deleted.
Program can assign classes for each individual student	Add a few classes and students to the program. Then go to the student's page to add a few classes. Check the combo box in the student page frame if it was successfully added. Remove some classes then see if combo box updates. Use an external program (DB Browser for SQLite) that can open database files to check if assigned classes were successfully added or removed from the database.
Client will be able to add edit and delete assignments	Select a class in the student page frame then create a dummy assignment and see if it appears in the assignment table of the program. Edit dummy assignment to see if changes are made in the table. Delete dummy assignment to check if it was deleted from the table. Use an external program (DB Browser for SQLite) that can open database files to check if it was successfully added, edited, and deleted.
Grade Calculator that will calculator a student's percentage in the class and grade	Add a few assignments for a selected class, then see if class percentage is successfully calculated and class grade falls in the proper range.
Database is used to present information back to the program when it is restarted	Add a few students, classes, and assignments to the program. Close the program then run the program again to check all the tables to see if all the rows are still there.
Program a search tool to find a student/class/parent	Add multiple students, classes, and assignments to the database. Use the search tool in each to see if the correct student comes up under the respective search conditions.