Criterion E: Evaluation

Product Evaluation via Success Criteria:

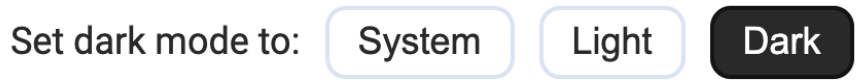
The program features:

1. The application detects if the user opened the application for the first time: **MET**

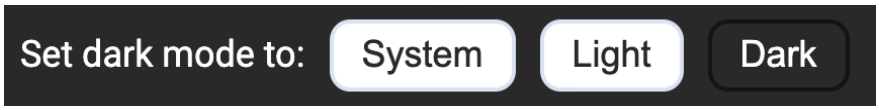
This criteria was easily met because the JSON user data has a variable named firstTimeOpened which has a default value of true but is instantly changed to false afterwards. Another way that validates this criteria for success is because the animation for loading the application happens once and never again no matter the amount of times the user closes and reopens the application. The variable for firstTimeOpened is also dependent on the existence of the file "userdata.json" which is stored in the user application data.

2. An operating system layout with a header and dark / light mode: **MET**

This criteria was met since the user is able to easily navigate to the system preference of the application from the header which allows the user for 3 options of different color modes such as default, dark, and light mode. Once the user clicks on any of these buttons, the software will save the color mode of the application and if the user quits the application and reopens it the color mode will remain the same from the last time they clicked.



Screenshot 1: Settings color mode clicking on light mode



Screenshot 2: Settings color mode clicking on dark mode

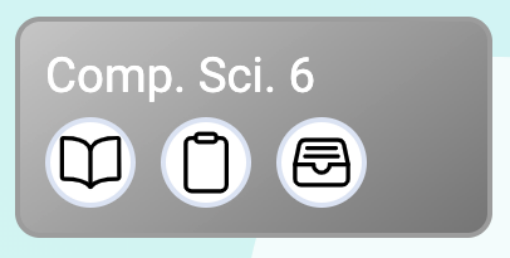
3. Ability to toggle the animation system: **MET**

Similarly to the changing the dark or light mode for the application, there is a button in the settings section that informs the user which animation style they would like, if they want a UI that acts quickly then they can disable the animation system, if not then they can have a smoother user experience.

4. A shortcut to edit each class from the home page: **MET**

This criteria was met because on the home page, each of the individual classes has 3 mini buttons that are shortcuts to edit the class, write report cards for the classes and access the repository for the individual classes. This allows the user to go to different applications and edit certain classes through one click of the home page.

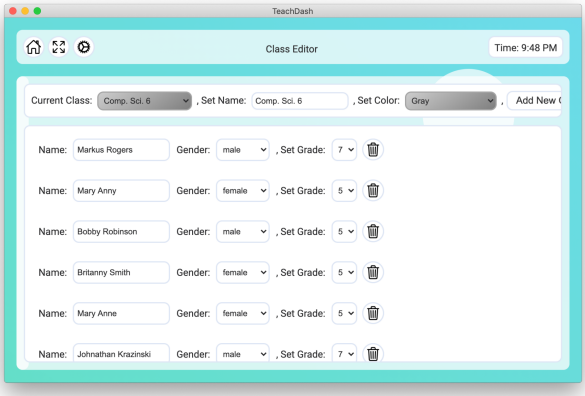
Screenshot:



Screenshot 3: A class from the home page with the 3 main buttons

5. Being able to create and edit different classes: **MET**

The sub-application class editor has a simple button that allows the user to create a new school class that is instantly saved to the user's computer once the user edits a class. The user can easily change the name, color and the student information on the application without needing to think about saving. The user is able to add students through the press of a button and is able to change the names, gender or grade of the student.



Screenshot 4: The class editor sub-application editing the first class

6. Ability to drag and drop files into a repository of each individual class with the ability to preview them: **MET**

In the repository application, the user is able to drag and drop files directly from their computer on to the application. The files are copied on to the user data folder and then are read by the application and loaded on in the repository. There is also a preview frame that the user can access by clicking on the eye icon for the individual files.

Screenshot 5: The repository sub-application previewing a text file

7. A scheduling system in a drag and drop format that is seperated in multiple weeks: **MET**

This criteria was met because when the scheduled sub-application is opened up, the user is able to click on the classes which then the user is able to drag it to the main body of the application. When the class is in range of a table box it gets dropped into the individual cell and it.

8. Having a text editor for writing report cards: **MET**

In the report card sub application, there is a middle section that contains a text area where the user is able to write the different report cards very easily. The application also detects changes to the text box and stores it onto the application if it's not there.

9. A data based for the different sentences to add them: **MET**

The report card sub-application stores the data from the different sentences that the TDStudentReport class stores to create report cards instantly for different classes. This data is then stored in the user data folder in a file named ReportCard.json.

10. The report card section can learn from the user's input and write: **MET**

The report card sub-application has a button with the label "Click To Learn" on the side of the application where a user can click on it for the application to collect the data from the text editor and convert it into sentences that can be reused by the application.

Client Feedback:

From contacting the client, during the interview they mentioned different elements of the product that they liked and disliked and had the ability to play around with it. The client was very satisfied with the features since I was able to implement all of their main ideas in the product.

Pros:

1. The first thing the client was pleased with was the design of the application stating that it is "clean and modern," with a lot of smooth animations that they were very pleased with. They also liked the freedom of having the ability to toggle the animation and the color mode for light and dark.
2. The client was happy with the simplicity of double clicking on sentences in the Report Card section and was "impressed" with the data analysis section of the report where the application is able to automatically generate a report based on the student.
3. They found the class editor to be intuitive and simple to use, which they mentioned would allow them to easily make changes or quickly add and remove new ones.
4. They also found the repository sub-application to be nice and simple for them to add and remove files from their desktop to the application to sort the different class materials.
5. The schedule system has a very intuitive and simple drag and drop user interface to create the time table matrix.

Cons:

1. An issue that the client mentioned was the fact that they felt that the widgets section was kind of empty and wasn't properly utilized.

Improvements & Further Recommendations:

The client had a few suggestions regarding what can be improved to make the quality of the product better both with small suggestions and larger sections of the application (Refer to appendix A3).

1. A small suggestion my client suggested was to add the title of the sub-application in the main menu of the application.
2. A larger section that the client suggested adding was to create a questions and answer sub-application that can create mini quizzes that the user can send off to other students through the web API.