

ICS4U7-3B Internal Assessment Criteria B

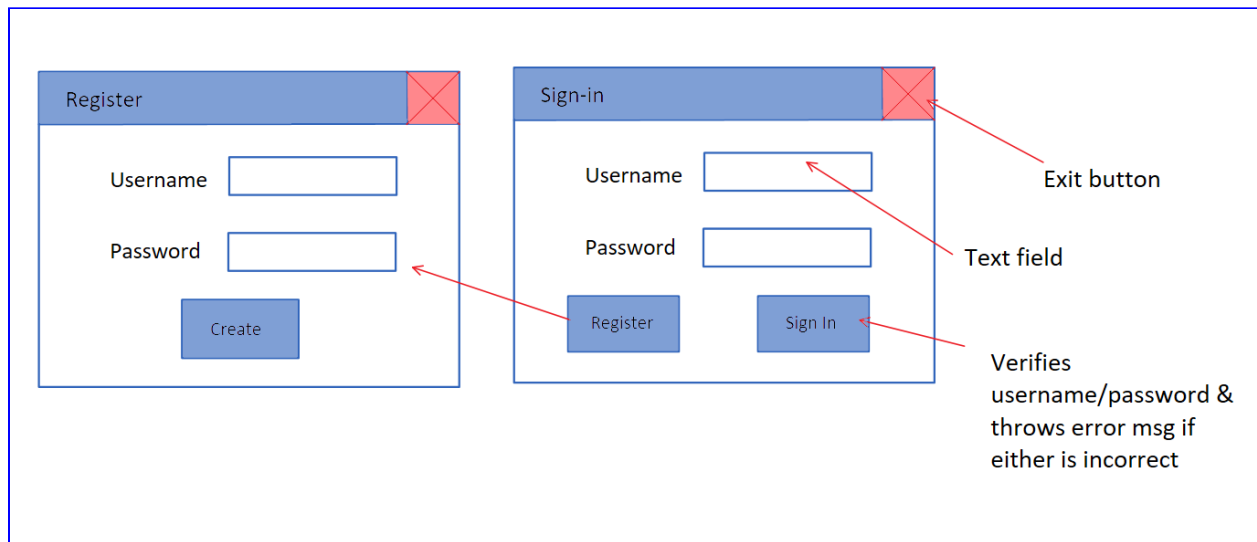
Software Design:

Figure 1: The diagram above shows the registration and sign-in screens planned for my product.

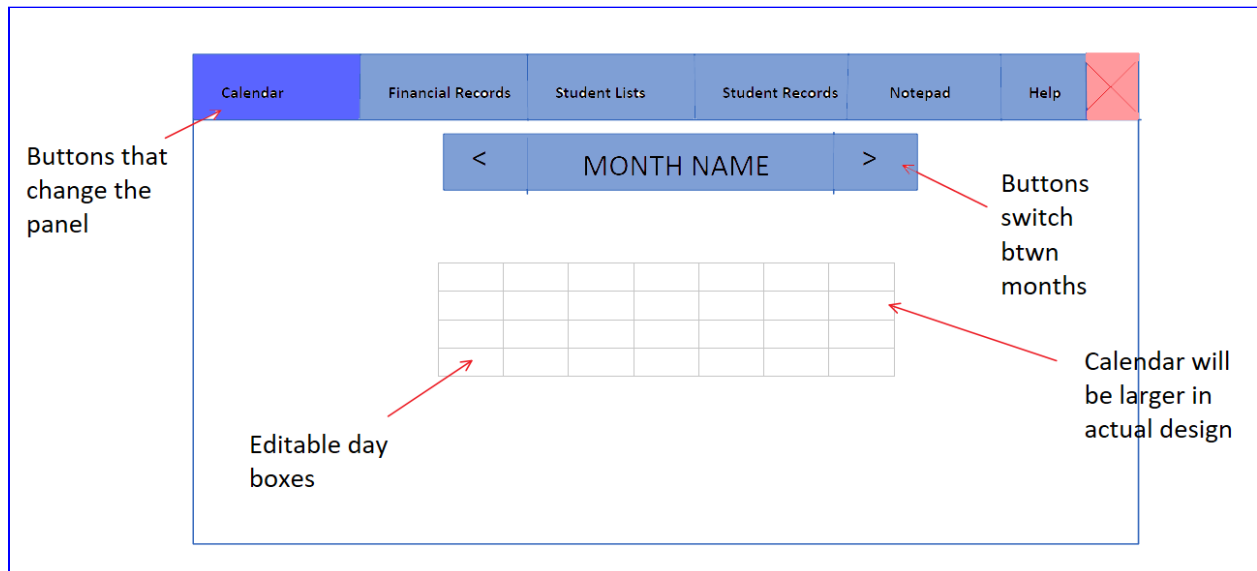


Figure 2: The diagram above shows the calendar screen planned for my product.

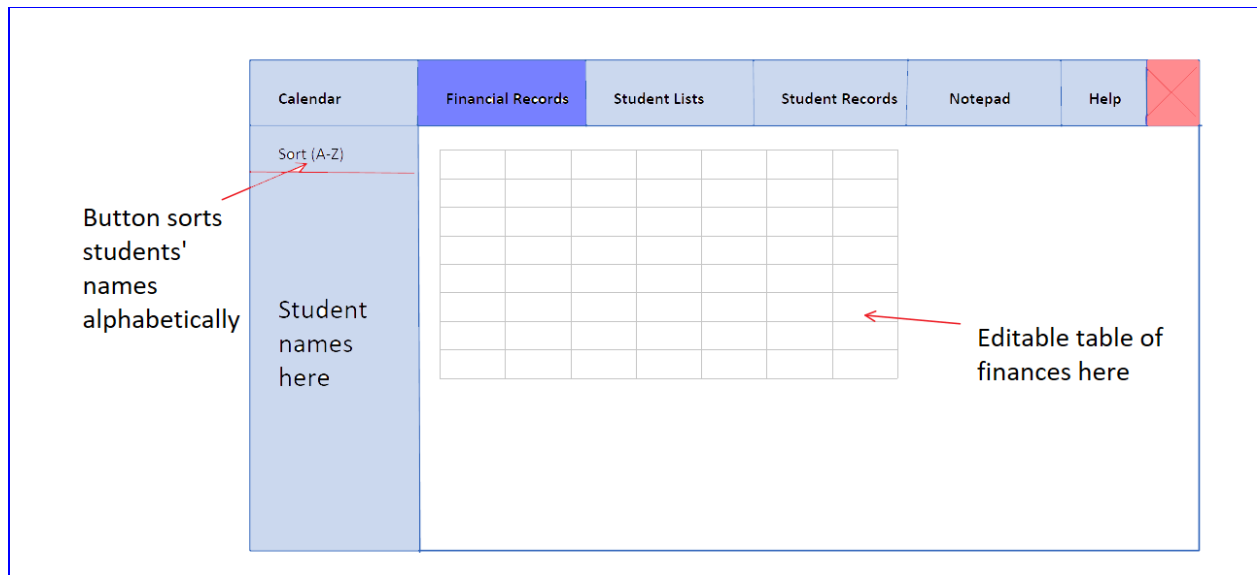


Figure 3: The diagram above shows the financial records screen planned for my product.

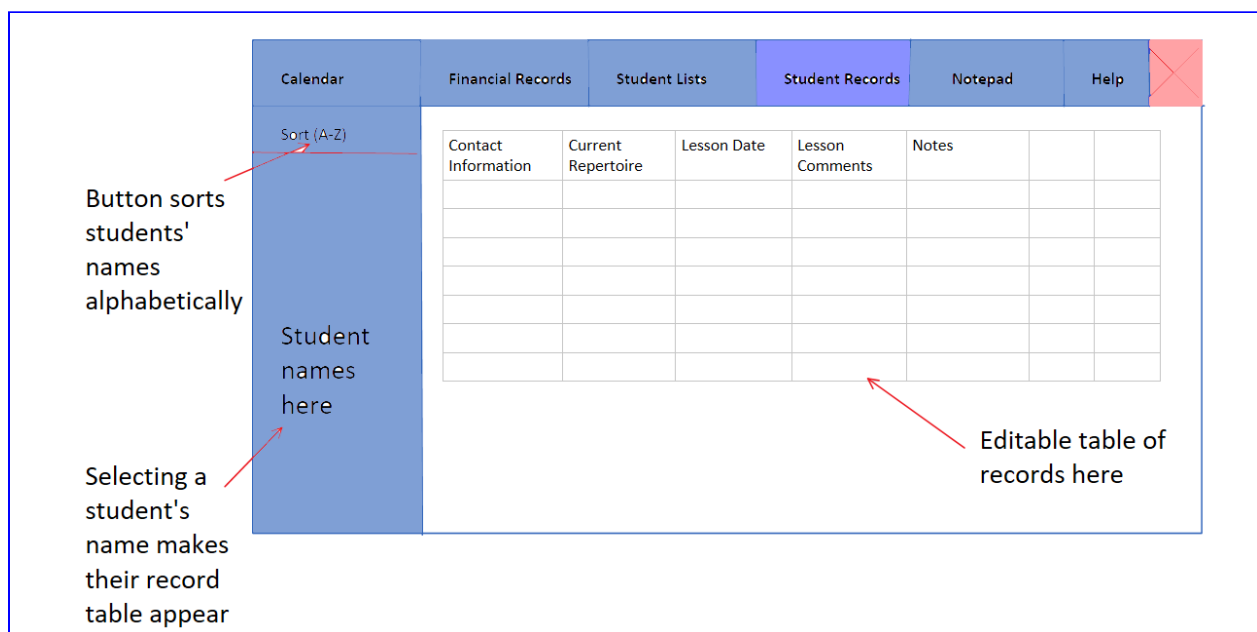


Figure 4: The diagram above shows the student records screen planned for my product.

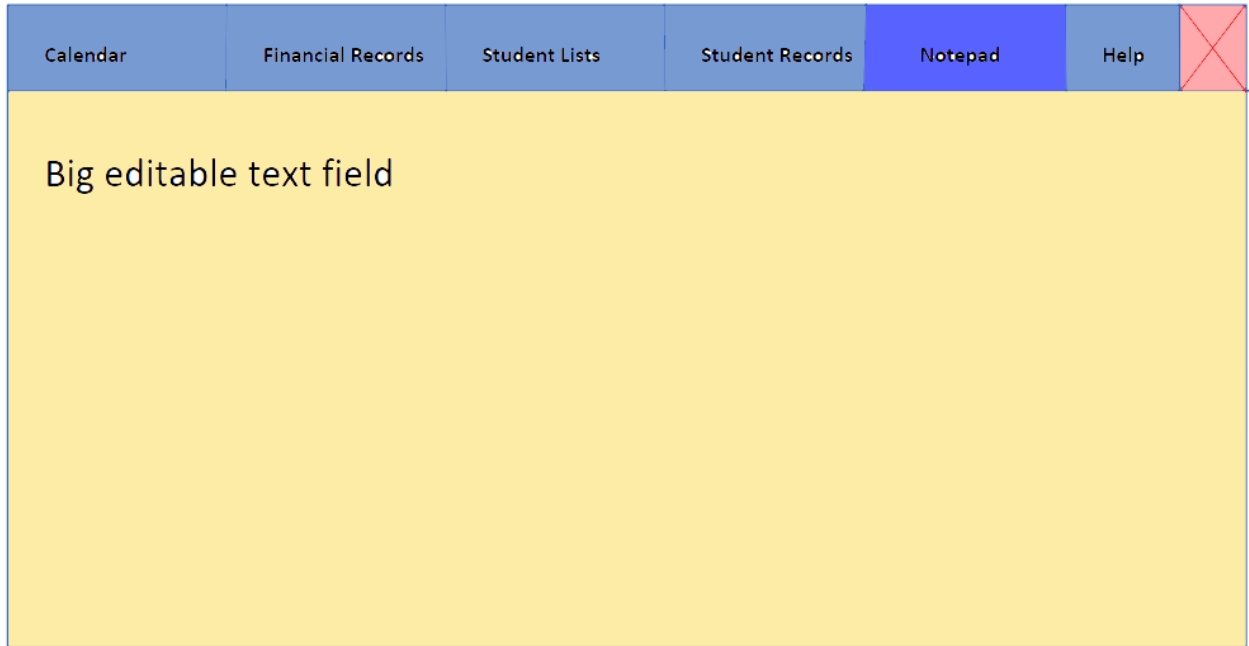
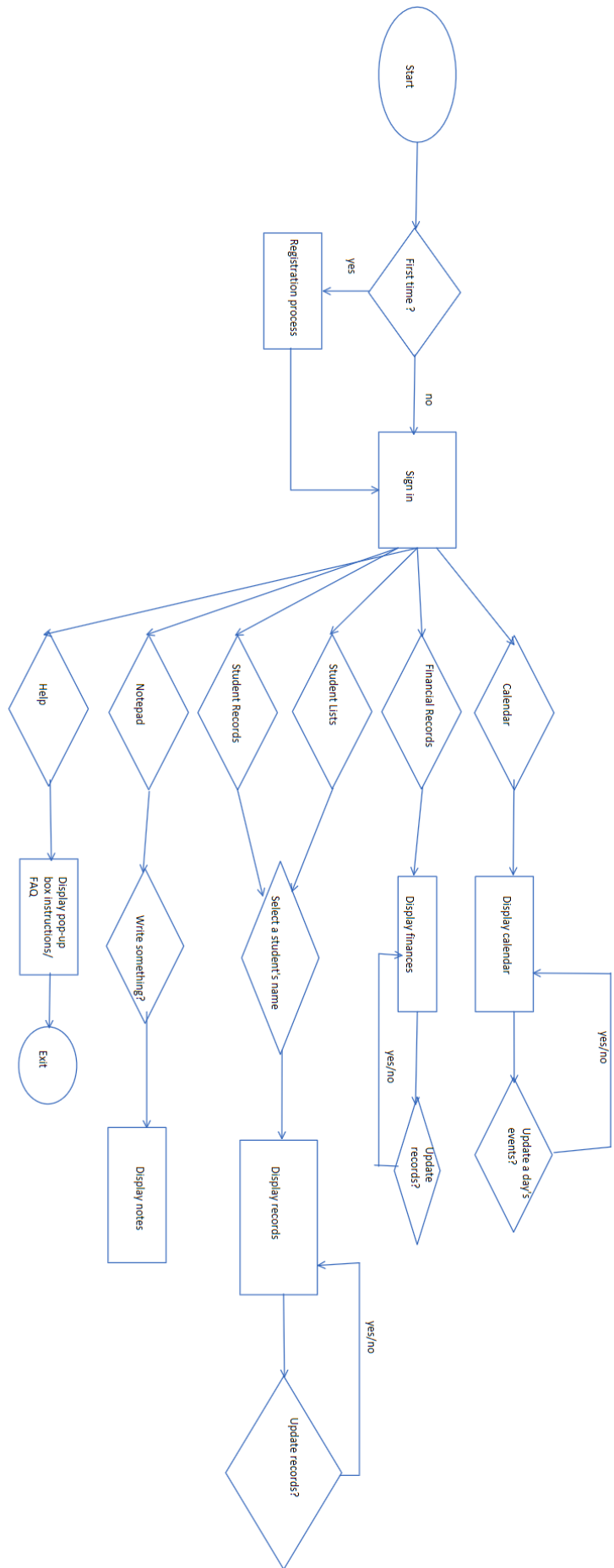


Figure 5: The diagram above shows the notepad screen planned for my product.

Flowchart:



Functions, Examples, and Comments:

Input Data	Location
<u>Login</u> Data 1 — Username & password Client registers with new login information	
<u>Example</u> musiclover100 password123	<u>Location</u> Login data will be stored in a text file that can be accessed to verify logins.
<u>Calendar</u> Data 1 — Adding a day's events Client inputs lesson information, including who will be attending lessons and the repetition frequency of each event. Client can also input other important dates.	
<u>Example</u> Date: Monday, January 16, 2020 Event: Insert information here Repeating: every week	<u>Location</u> A text file with all the inputted data will be saved.
<u>Financial Records</u> Data 1 — Student names Client can input/update the names of students.	

<u>Example</u> Jenna Marbles	<u>Location</u> A text file with all the inputted data will be saved (names.txt).
Data 2 — Records Client can input payments and dates in a row-column table.	
<u>Example</u> Monday Tuesday Wednesday \$40.00 \$25.00 No payment	<u>Location</u> A text file with all the inputted data will be saved.
<u>Student Lists</u> Data 1 — Student names Client can input/update the names of students.	
<u>Example</u> Jenna Marbles	<u>Location</u> The text file (names.txt) will be accessed and updated accordingly.
<u>Student Records</u> Data 1 — Contact Information Client can input/update contact information for students.	
<u>Example</u>	<u>Location</u>

Email: xx Phone Number: yy	A text file with all the inputted data will be saved.
Data 2 — Repertoire Client can input/update current repertoire students are learning.	
<u>Example</u> Bruch's Violin Concerto No. 1 in G minor Beethoven's Moonlight Sonata	<u>Location</u> A text file with all the inputted data will be saved.
Data 3 — Lesson Date Client can update/delete lessons. Dates should be automatically read and printed from the calendar file.	
<u>Example</u> Tuesday, January 19, 2020 — 9:00 PM – 11:00 PM	<u>Location</u> Information read here will not be stored nor read elsewhere.
Data 4 — Comments Client can write/update comments on lessons and progress.	
<u>Example</u> Jenna worked on her xx today.	<u>Location</u> A text file with all the inputted data will be

	saved.
<p>Data 5 — Notes</p> <p>Client can write/update extra notes here.</p>	
<p><u>Example</u></p> <p>Reminder: Ask Mr. Smith for recital information.</p>	<p><u>Location</u></p> <p>A text file with all the inputted data will be saved.</p>
<p><u>Notepad</u></p> <p>Data 1 — Notes</p> <p>Client can write/update as many notes as wanted here.</p>	<p><u>Location</u></p> <p>A text file with all the inputted data will be saved.</p>

Testing Plan:

Action Test	Result
Run the program and check if it works.	Should work without any errors.
Check registration process.	After creating a new login with the register button, check to see if the information is stored in the text file. Check to see if a new directory is made with the username.
Check the login validation process.	Check if proper error messages are outputted when a user inputs the incorrect username, password, or both. Check if, and only if, entering the correct login information allows the user to proceed to the homepage.
Check calendar accuracy.	Check to see if the calendar and dates display correctly align with a regular calendar.
Check for the addition and deletion of events.	Check if new events are correctly saved to a text file in the Calendar directory and if they can be displayed upon request. Check to see if events labelled “repeating” properly repeat and create corresponding text

	<p>files for future dates.</p> <p>Check to see if deleting the events properly erases the contents of the text files.</p>
<p>Check addition and deletion of students in student lists.</p>	<p>Check to see if the addition of a new student creates a new text file in the Student Lists directory with all the inputted contact information.</p> <p>Check to see if the addition of new students adds their names to the Master List text file.</p> <p>Check to see if the deletion of a student erases the contents of the text file.</p>
<p>Check sorting process of first and last names.</p>	<p>Check to see if the names of the students are properly sorted in alphabetical order, depending on the selection of sorting by first or last names, for Student Lists, Financial Records, and Student Records.</p>
<p>Check the saving and erasure of inputted financial data.</p>	<p>Check to see if the save function properly writes to the respective text files and erases the files' contents when the user deletes the information.</p>
<p>Check for proper display of existing students.</p>	<p>Check to see if all students recorded in the</p>

	Master List text file are properly retrieved and displayed in both the Financial Records screen and the Student Records one.
Check search function.	Check to ensure the proper position of the student's name is outputted when the user requests a search for them.
Check saving of notepad input.	Check to see that input in the Notepad screen is saved to a text file and can be retrieved.
Check for varying help boxes.	Check to see that different help boxes with screen-specific instructions correctly display when the user presses the help button on different screens.