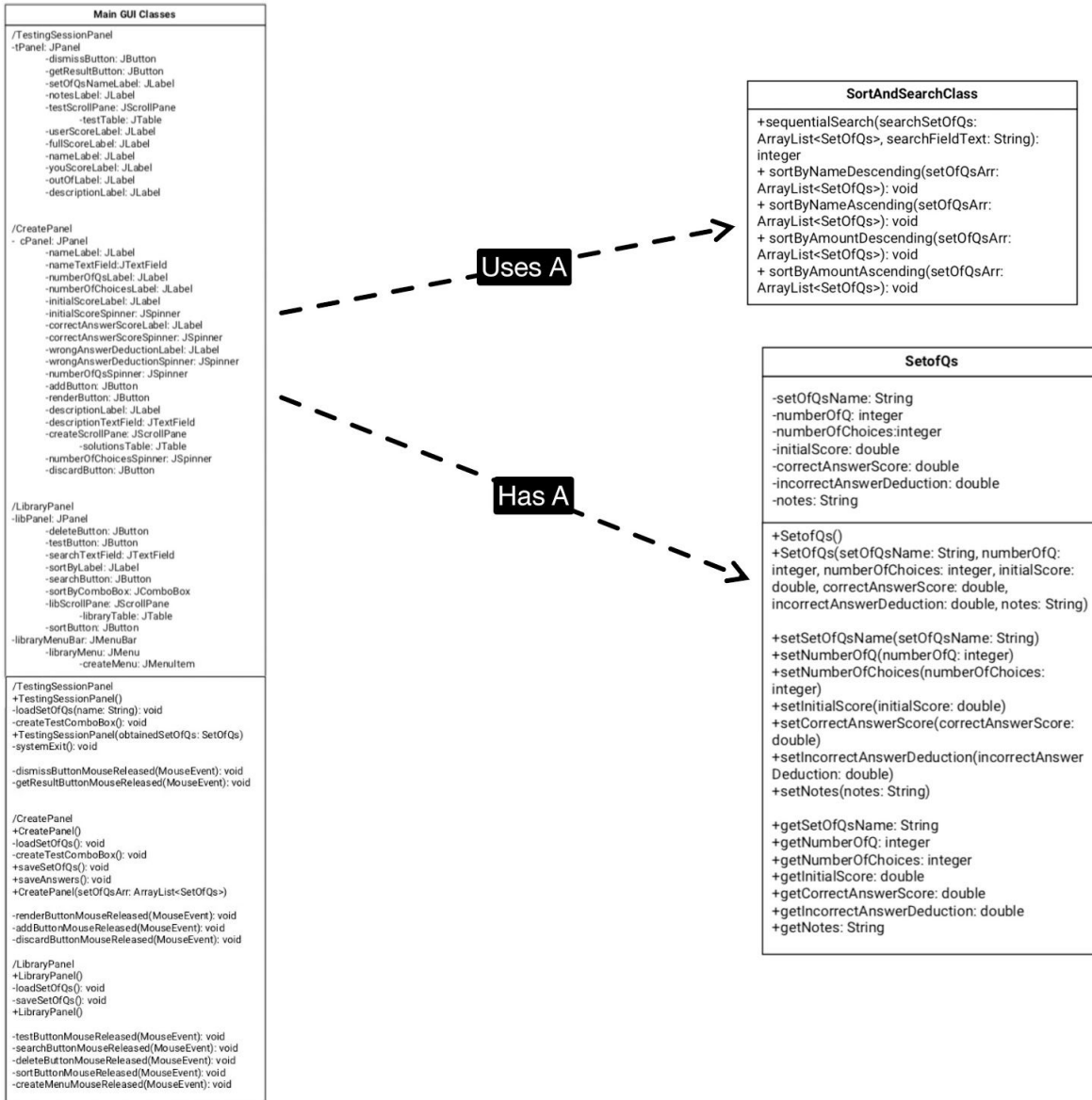


# Criterion B: Design

## Class Diagram:



## Inputs/Outputs:

Create Panel Input & Output:

<b>Input</b>	<b>Data Type</b>	<b>Example</b>
Number of questions	Integer	"15"
Number of choices	Integer	"4"
Score given per correct answer	Double	"1"
Score deducted per wrong answer	Double	"0.25"
Initial score	Double	"0"
Name	String	"Thermal Physics Paper 1 Questions"
Notes	String	"This is a practice test."
Solution	String	"A."
<b>Output</b>	<b>Data Type</b>	<b>Example</b>
Questions	Integer	"1"

Test Panel Input & Output:

<b>Input</b>	<b>Data Type</b>	<b>Example</b>
Your Answer	String	"A."
<b>Output</b>	<b>Data Type</b>	<b>Example</b>
Name	String	"March SAT 2018"
Notes	String	"Please give yourself 60 minutes to complete this test"
Questions	Integer	"2"

Your Final Score	Double	"35.0"
Full Score	Double	"40.0"
Answer Key	String	"C."

Library Panel Input & Output:

<b>Input</b>	<b>Data Type</b>	<b>Example</b>
Search field	String	"IB Physics"
<b>Output</b>	<b>Data Type</b>	<b>Example</b>
Question name	String	"2"
Amount of question	Integer	"0"

## Prototype Process:

### 1. Initial Prototype (with Client's Annotations)

File Edit Help

Library Open Create/Edit Result

Set of qs name

Question Name

Date Taken

Total Time Spent

Average Time Spent Per Each Question

Score

test taker

Session name

This screenshot shows a software prototype window with a menu bar (File, Edit, Help) and a toolbar (Library, Open, Create/Edit, Result). The main area contains a form with several text input fields. Handwritten annotations in black ink are present: "Set of qs name" above the first field, "test taker" above the sixth field, and "Session name" above the seventh field. Each field has a small rectangular box next to it, likely representing a data field or a placeholder.

File Edit Help

Library Open Create/Edit Result

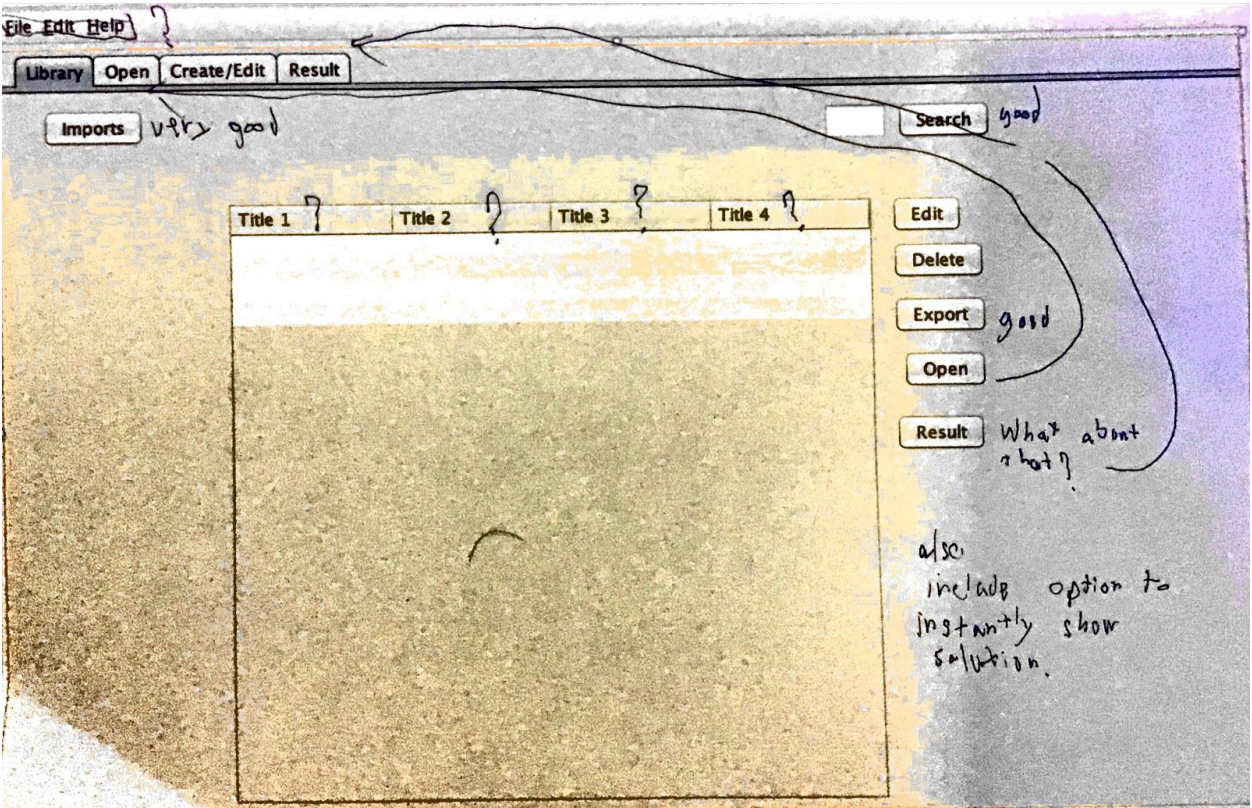
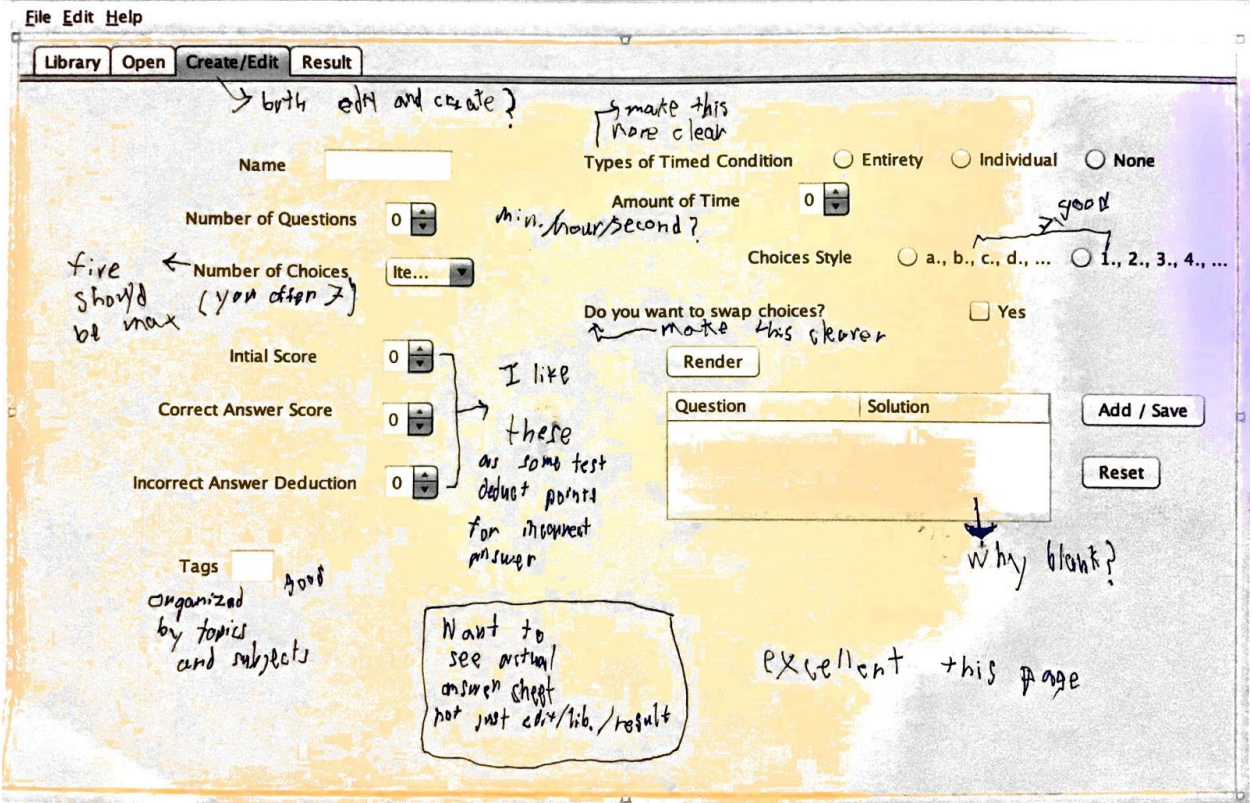
test taker

User

Exam condition  yes

This screenshot shows a software prototype window with a menu bar (File, Edit, Help) and a toolbar (Library, Open, Create/Edit, Result). The main area contains a form with a text input field and a checkbox. Handwritten annotations in black ink are present: "test taker" above the text input field, with an arrow pointing to the field, and "Exam condition" followed by a checkbox and the word "yes" below it.





## 2. Changed Prototype

Design Preview [createPanel]

Name

Number of Questions  Initial Score  Correct Answer Score

Time (minute(s))  Number of Choices  Answer Deduction

Tag(s):

Review Answer    Choice Style:  1., 2., 3., 4,....  
 Swap Choices     a., b., c., d,....     n., B., R., 4,....

Notes:

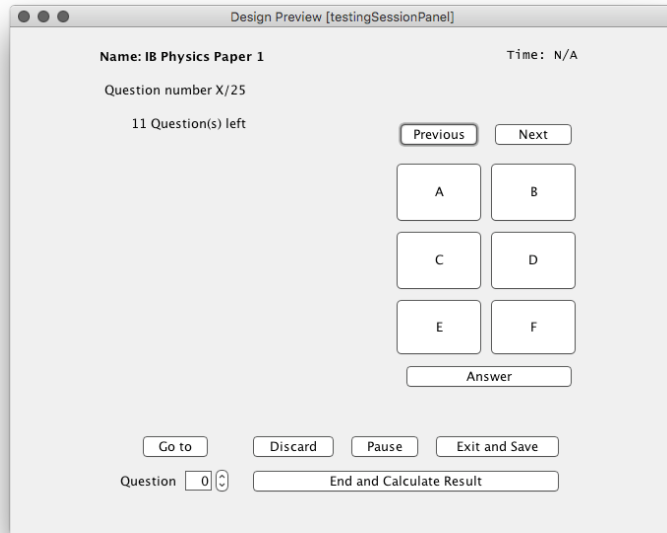
Question(s)	Solution(s)
-------------	-------------

Design Preview [libraryPanel]

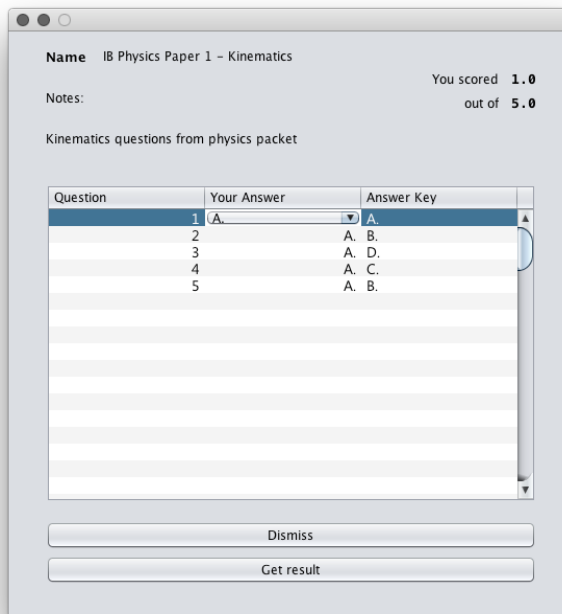
Sort by:

Tag(s):

Name	Amount of Que...   Tag(s)	Duration	Date Created
------	---------------------------	----------	--------------



### 3. Final Prototype







## **Chronological Development:**

### **1. GUI Coding and Interviews**

- a. Initial interview & initial prototype
  - i. Create a tabbed pane to allow for the creation of sets of questions
  - ii. Create a 2nd tabbed pane to manage the created sets of questions
  - iii. Create a 3rd tabbed pane to be a test panel that can be used to launch sets of questions
  - iv. Create a 4th tabbed pane to display test results for each set of questions
- b. Second Interview & Changed Prototype
  - i. Rework, improve, and simplify the initial GUI by using multiple windows instead of tabs, creating a testing session panel, and abandoning a button for timing each question.
  - ii. Create confirmation pop-up windows for delete and 'continue'.
  - iii. Assign a proper name to each variable
  - iv. Add in a basic MouseReleased event for each button
  - v. Make windows non-resizable

### **2. Create template classes and methods**

- a. Create set of questions template class
  - i. Come up with and add attributes for each sets of questions
- b. Create a sorting and searching class
  - i. Implement the selection sort algorithm to sort sets of questions in the library panel, sort results in the results panel
  - ii. Implement sequential search for search perform via the searchTextField
- c. Create a calculation class
  - i. Create a class that compares user's answers to solutions to calculate user's score
  - ii. Calculation class also returns the calculated 'results' to the result class, which extends the sets of question class.
- d. Create an import method and an export method
  - i. Design and code the export method to output .txt file
  - ii. Design and code how the import method should work with exported .txt file
- e. Create saving and loading methods for writing and reading sets of questions and their solutions
- f. Implement those classes and methods into the program

### 3. Come up with a testing plan

- a. Identify errors
- b. Design and program handlings of error
- c. Test for errors
- d. Further fix error handling issues

### 4. Produce a demo video of the program

### 5. Final interview

- a. Give client a walkthrough and a live demo of the program
- b. Recieve general feedbacks
- c. Fix reported error handling issues

word count: 0 (Bulleted Lists)

## Testing Plan:

- Testing Plan
  - Check the library panel to see if reading and writing of saved sets of questions is working properly.
    - Test the program to see if there are errors that cause the program to stop running.
  - Check the entries in the create panel to make sure that they work as they should and check to make sure that the spinners' minimum and maximum value prevent each parameter's inputs from exceeding each parameter's limit.
    - Number of choices should not be more than 6 or less than 2, the number of questions should not be more than 100.
    - Other parameters shouldn't accept any negative number
  - Check the outputs in the test panel to make sure that they output what was input in the create panel.
  - Check the library panel to make sure that the search and sorting works.
    - Try searching
    - make sure that the sorting works.
  - Check the save function
    - Click add in the create panel and go to the library panel to make sure that the newly created set of questions is displayed.
  - Check the read function
    - Create a set of questions, relaunch the program and make sure that it's still in the libraryTable.
- Error Checking
  - //Create panel
  - Leaving the name textfield blank
    - Error message: Please name your set of questions.
  - Entering a name that has already been used

- Error message: Please choose another name for your set of questions.
- Leaving one or more of the solutions ComboBoxes blank
  - Error message: Please fill in the remaining solution(s).
- //Library panel
- Interacting with an empty row
  - Error message: This is an empty row. Please select another row.

word count: 265

criterion A + B word count: 712