05/2023

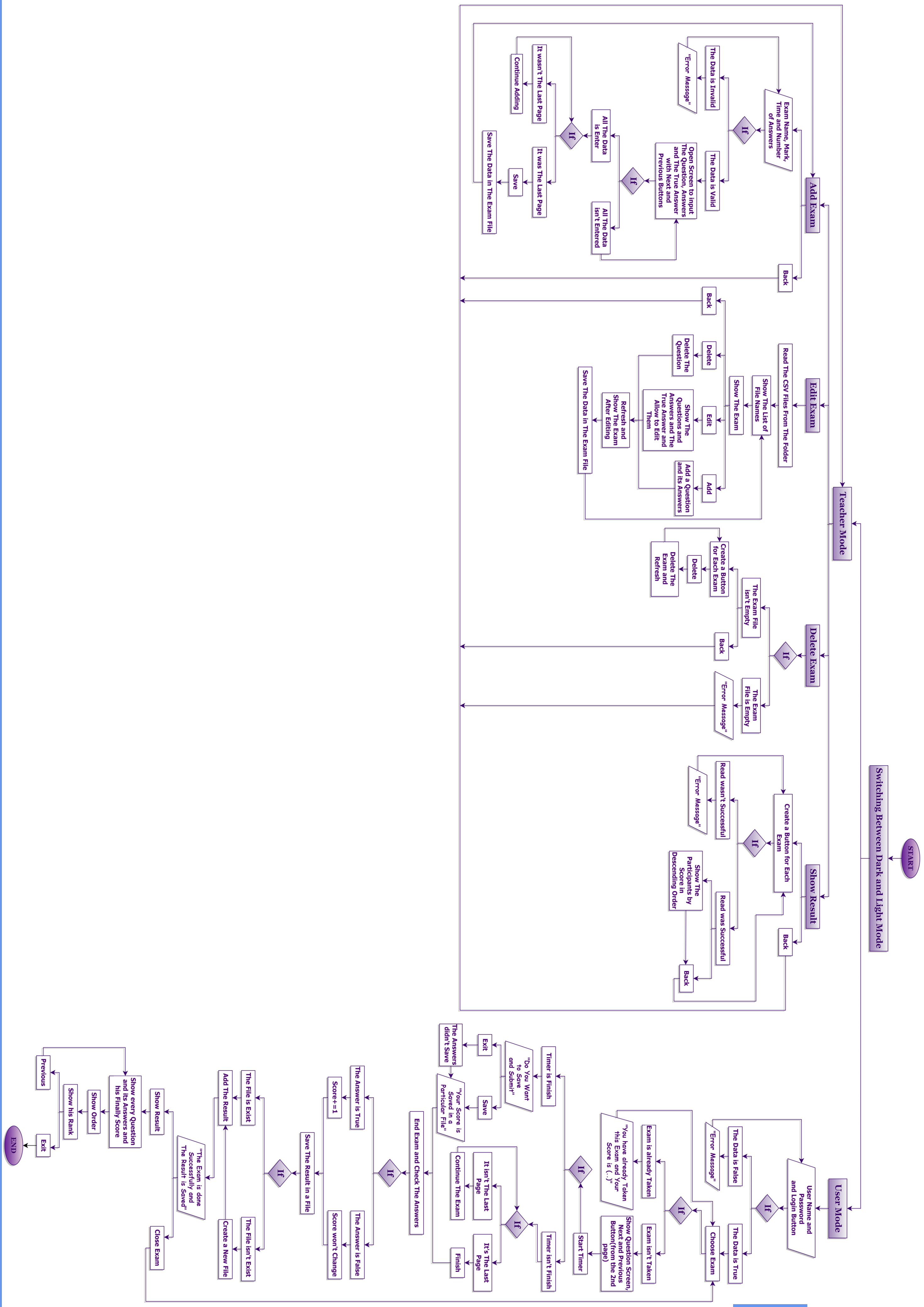
Creating Exams Project

Java mini project

Submitted by : Group (29)

Student Names

• Ahmed Riyad Mohamed Riyad	1
• Ahmed Maher Tohamy Elhady	2
• Ahmed Maher Ali Ibrahim	2
• Fayrouz Abdelhalim Taha Abdelhalim	11
• Mohamed Mamdouh Mohamed Abdelhamid	15
• Mahmoud Aboelazm Mohamed Salah	16
• Noha Ahmed Mohamed Elsayed	19
• Nourhan Hesham Abdelaziz Abonar	19
• Heba Tallah Ezzat Abdelrahman Osman	19
• Waad Elsayed Mohamed Ahmed	19
• Yasmin Emad Ibrahim Eldesoky	20

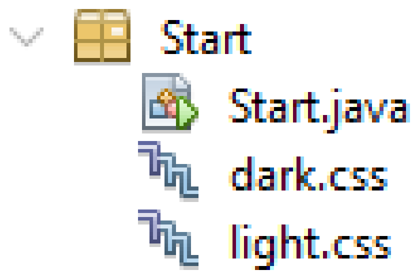

START
Switching Between Dark and Light Mode
Teacher Mode
User Mode
Add Exam
Exam Name, Mark, Time and Number of Answers
If
The Data is Invalid
"Error Message"
The Data is Valid
Open Screen to input The Question, Answers and The True Answer with Next and Previous Buttons
If
All The Data is Enter
All The Data isn't Entered
If
It wasn't The Last Page
Continue Adding
It was The Last Page
Save
Save The Data in The Exam File
Back
Edit Exam
Read The CSV Files From The Folder
Show The List of File Names
Show The Exam
Delete
Delete The Question
Edit
Show The Questions and Answers and The True Answer and Allow to Edit Them
Add
Add a Question and its Answers
Refresh and Show The Exam After Editing
Save The Data in The Exam File
Back
Delete Exam
If
The Exam File isn't Empty
Create a Button for Each Exam
Delete
Delete The Exam and Refresh
Back
The Exam File is Empty
"Error Message"
Show Result
Create a Button for Each Exam
If
Read wasn't Successful
"Error Message"
Read was Successful
Show The Participants by Score in Descending Order
Back
Back
User Name and Password and Login Button
If
The Data is False
"Error Message"
The Data is True
Choose Exam
If
Exam is already Taken
"You have already Taken this Exam and Your Score is (. .)"
Exam isn't Taken
Show Question Screen, Next and Previous Button(from the 2nd page)
Start Timer
If
Timer is Finish
"Do You Want to Save and Submit"
Exit
The Answers didn't Save
Save
"Your Score is Saved in a Particular File"
Timer isn't Finish
If
It isn't The Last Page
Continue The Exam
It's The Last Page
Finish
End Exam and Check The Answers
If
The Answer is True
Score+=1
The Answer is False
Score won't Change
Save The Result in a File
If
The File is Exist
Add The Result
The File isn't Exist
Create a New File
"The Exam is done Successfully and The Result is Saved"
Show Result
Show every Question and its Answers and his Finally Score
Show Order
Show his Rank
Previous
Exit
Close Exam
END

Packages :

The code consists of 3 pakages as follows :

First : the Start package

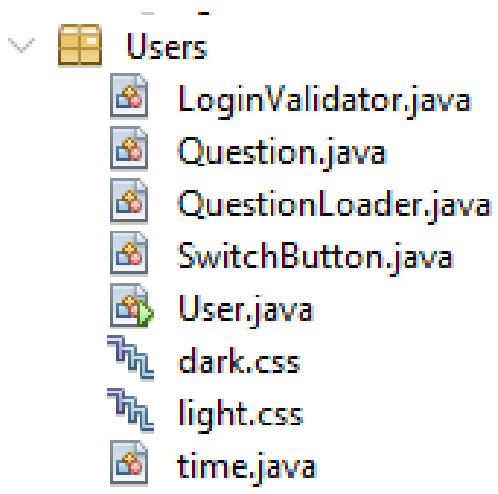
It contains 3 files , the Start file and the 2 styling css files (dark & light) for switching between two modes .



Start
- Start.java
- dark.css
- light.css

Second : the Users package

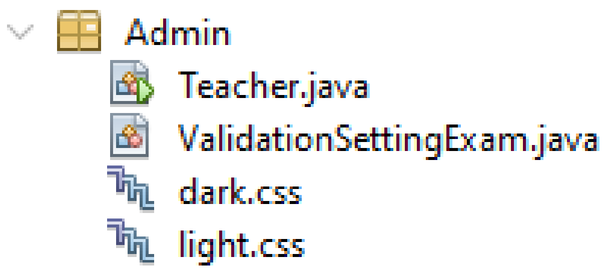
It contains 8 files : LoginValidator,Question,QuestionLoader, SwitchButton,User,time files and the 2 styling css files (dark & light) for switching between two modes .



Users
- LoginValidator.java
- Question.java
- QuestionLoader.java
- SwitchButton.java
- User.java
- dark.css
- light.css
- time.java

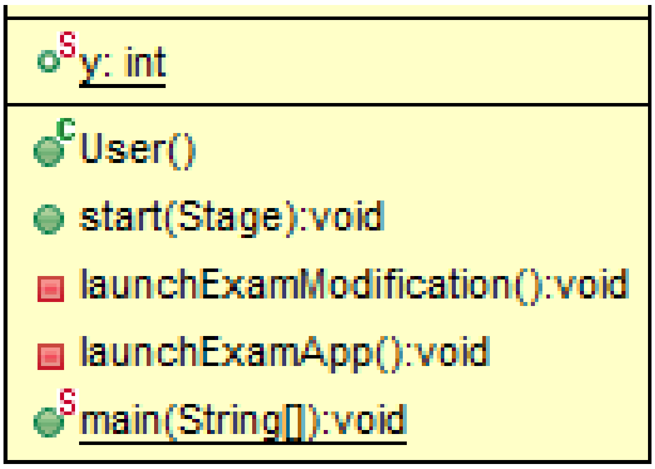
Third: the Admin package

It contains 4 files : Teacher,ValidationSettingExam files and the 2 styling css files (dark & light) for switching between two modes .



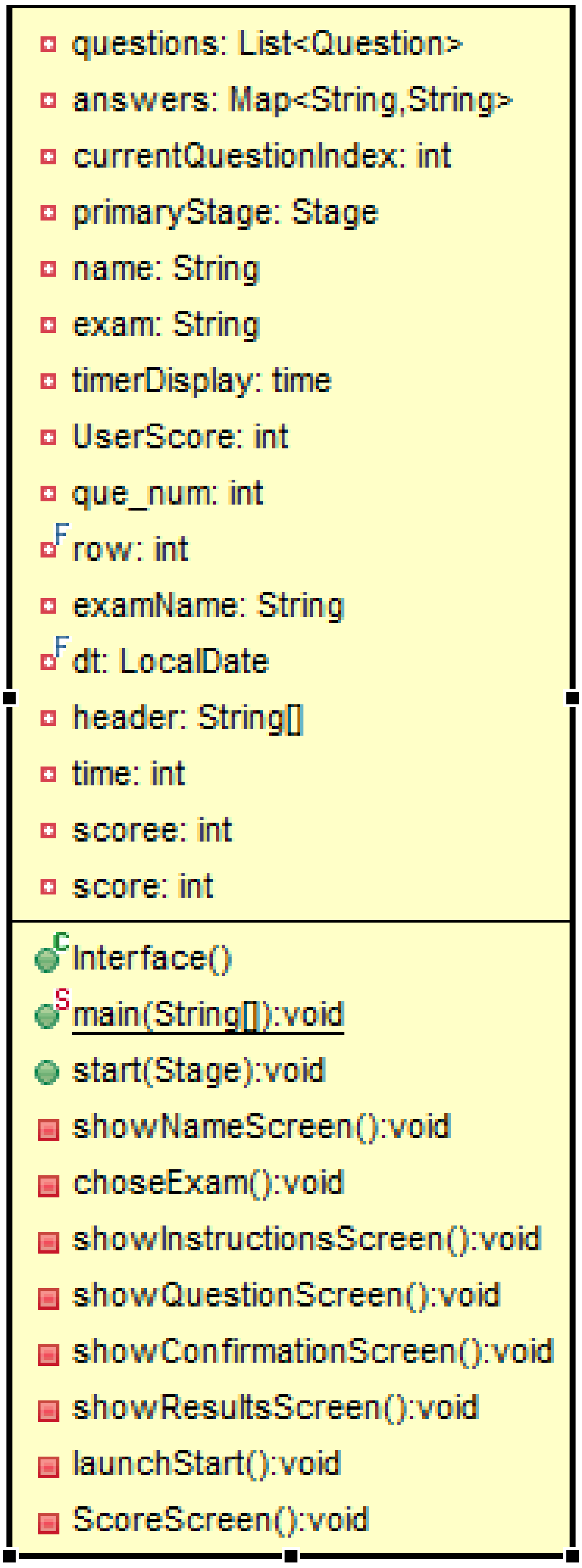
Admin
- Teacher.java
- ValidationSettingExam.java
- dark.css
- light.css

UML of User Part



y: int

User()
start(Stage):void
launchExamModification():void
launchExamApp():void
main(String[]):void

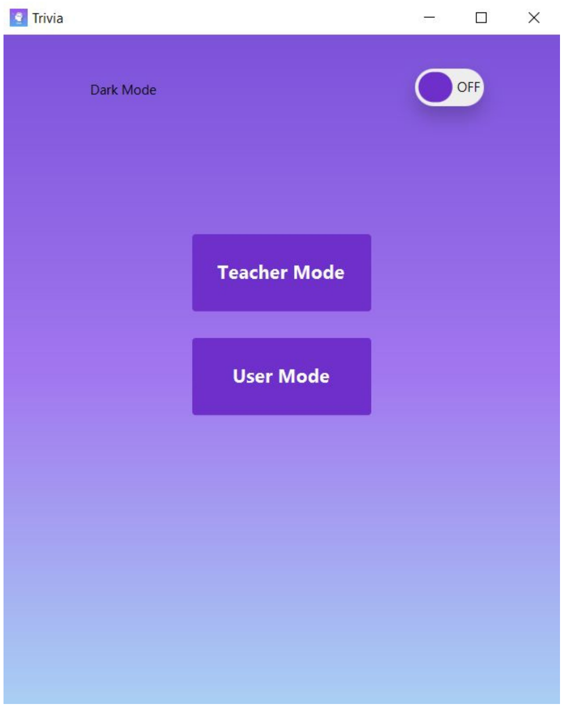


questions: List<Question>
answers: Map<String,String>
currentQuestionIndex: int
primaryStage: Stage
name: String
exam: String
timerDisplay: time
UserScore: int
que_num: int
row: int
examName: String
dt: LocalDate
header: String[]
time: int
scoree: int
score: int

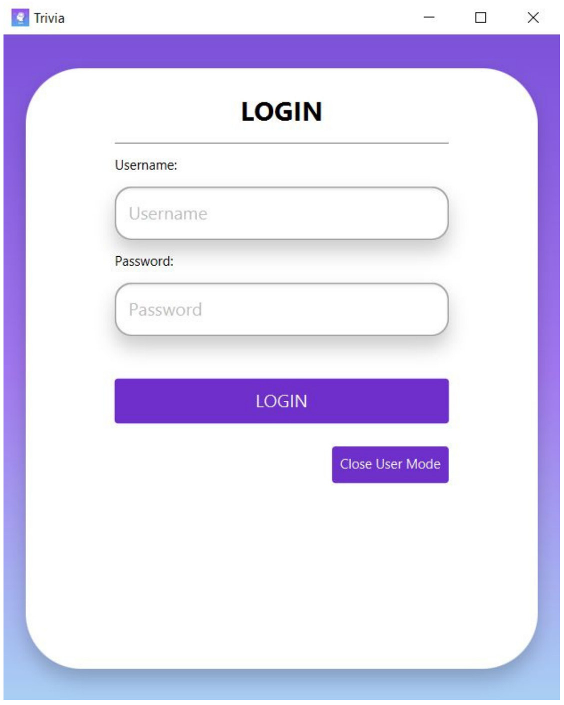
Interface()
main(String[]):void
start(Stage):void
showNameScreen():void
choseExam():void
showInstructionsScreen():void
showQuestionScreen():void
showConfirmationScreen():void
showResultsScreen():void
launchStart():void
ScoreScreen():void

Explanation of User Part

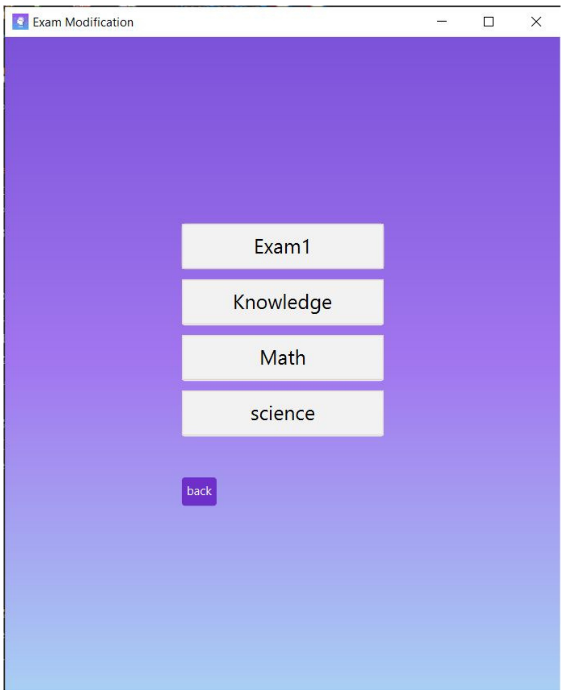
- Create two buttons: one for Teacher Mode and one for User Mode.
- Set the actions for the buttons.
- Create a Grid Pane layout and set its properties.
- Add a label for Dark Mode to the layout.
- Add a Switch Button to the layout to toggle between Dark Mode and Light Mode.
- Add the buttons to the layout.
- Create and show the scene.



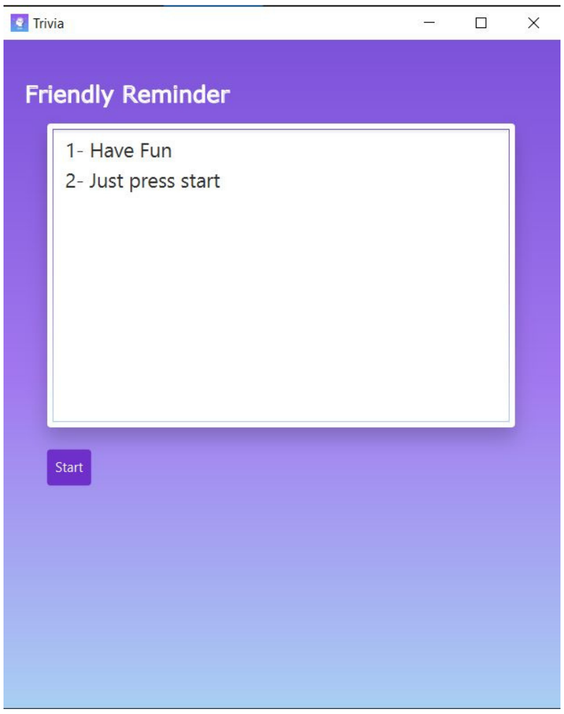
- The show Name Screen() method creates a new Grid Pane to layout the name screen.
- The Grid Pane is then populated with a label and text field for the user's name, a label and password field for the user's password, and a button to start the exam.
- When the user clicks the start button, the method validates the user's login credentials.
- If the credentials are valid, the method saves the user's name and moves on to the exam selection screen.
- If the credentials are invalid, the method displays an error message.
- The method also includes a button to close the user mode. When the user clicks this button, the method closes the primary stage and launches the start method.



- Create a Grid Pane to display the exam selection screen.
- Add a label for the exam selection prompt to the Grid Pane.
- Add a list of radio buttons to the Grid Pane for each available exam.
- Add a button to the Grid Pane to start the selected exam.
- When the user clicks a radio button, get the name of the selected exam.
- Check if the user has previously taken the exam.
- If the user has not previously taken the exam, load the selected exam's questions and shuffle them.
- Set the current question index to 0.
- Display the instructions screen. If the user has previously taken the exam, inform the user that they have already taken the exam and display their score.



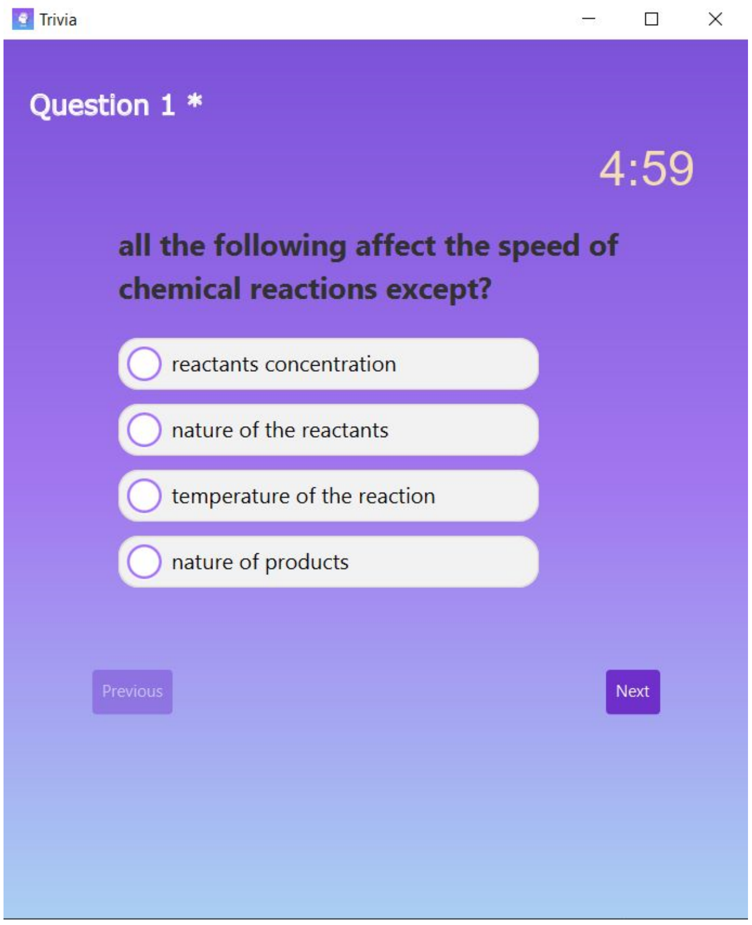
- Create a new Grid Pane to layout the instructions screen.
- Add a Text Area to display the exam instructions.
- Set the Text Area to be non-editable and to wrap text.
- Load the exam instructions from a file and display them in the Text Area.
- Add a "Start" button to the Grid Pane.
- Set the action for the "Start" button to show the questions screen.
- Create a new scene with the Grid Pane and set it as the primary stage's scene.
- Set the scene's stylesheet based on the current dark mode setting.



1- Have Fun
2- Just press start

- Get the current question from the questions list.
- Create a GridPane to display the question and choices.
- Add the question label to the GridPane in the first row, first column.

- Add the timer text to the GridPane in the second row, first column.
- Add the toggle group for the choices to the GridPane in the first row, starting from the second column.
- Add the previous button to the GridPane in the second row, second column.
- Add the next or finish button to the GridPane in the second row, fourth column.
- Set the event handlers for the previous button, next or finish button, and radio buttons.
- Set the scene and show the stage.

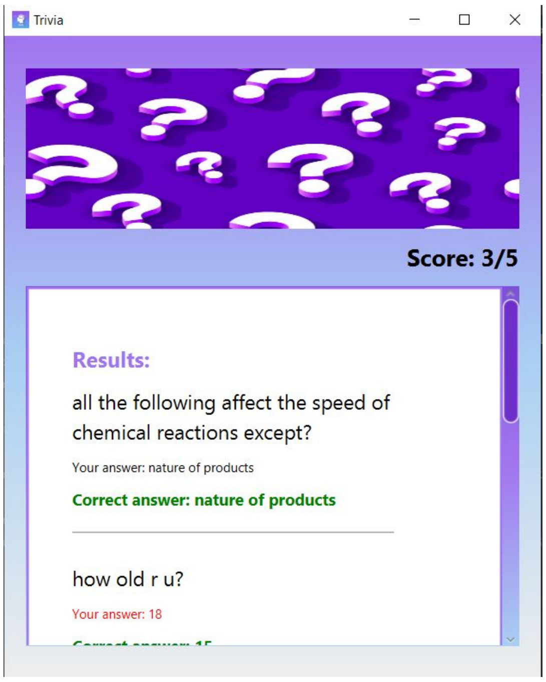


all the following affect the speed of chemical reactions except?

- reactants concentration
- nature of the reactants
- temperature of the reaction
- nature of products

- Remove the close request handler from the primary stage.
- Create a new VBox to hold the buttons.
- Loop through the questions list and update the score if the user's answer is equal to the correct answer.
- Create a new file named ExamResults/ResultsOF + exam.
- If the file already exists, check if the file has a header. If it does not, print the header to the file.
- Append the participant's name, score, and date to the file.
- Create two buttons: a "Close Exam" button and a "Show Results" button.
- Create a new scene with the VBox and set it as the primary stage's scene.
- Show the primary stage.

- Create a new GridPane to hold the results.
- Create a label for the title and add it to the GridPane.
- Create a new ScrollPane with the GridPane as its content.
- Loop through each question and add the question text, the user's answer (if they answered the question), and the correct answer to the GridPane.
- Add a label for the score to the GridPane.
- Add a button to save the results to a CSV file and handle the button's action.
- Create a new scene with the ScrollPane and set it as the primary stage's scene.
- Show the primary stage.



Score: 3/5

Results:

all the following affect the speed of chemical reactions except?

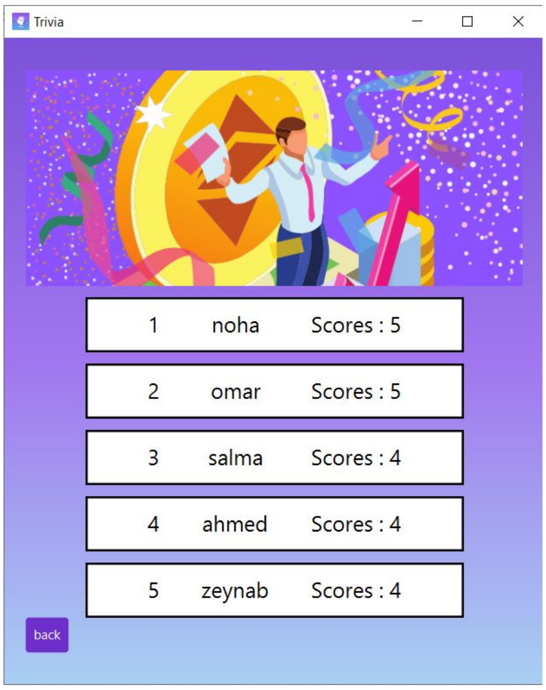
Your answer: nature of products

Correct answer: nature of products

how old r u?

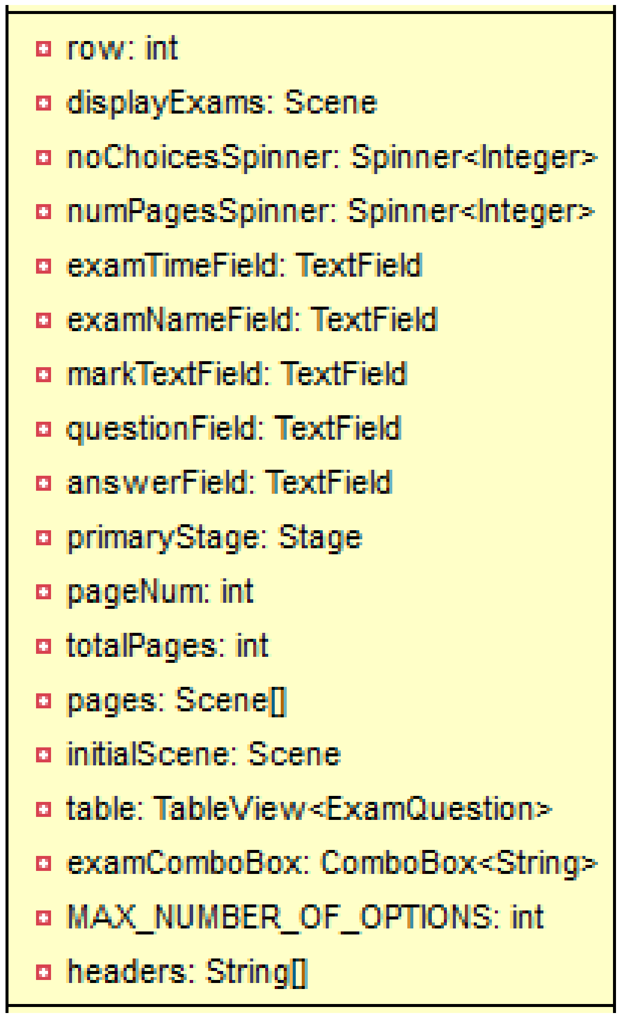
Your answer: 18

- Create a new GridPane to hold the scores.
- Read the participants' scores from the CSV files.
- Sort the participants' scores in descending order.
- Create a StringBuilder to store the participants' names and scores.
- Create a TextArea and set its text to the contents of the StringBuilder.
- Add the TextArea to the GridPane.
- Create a button and set its text to "back".
- Add the button to the GridPane.
- Create a new scene with the GridPane and set it as the primary stage's scene.
- Show the primary stage.

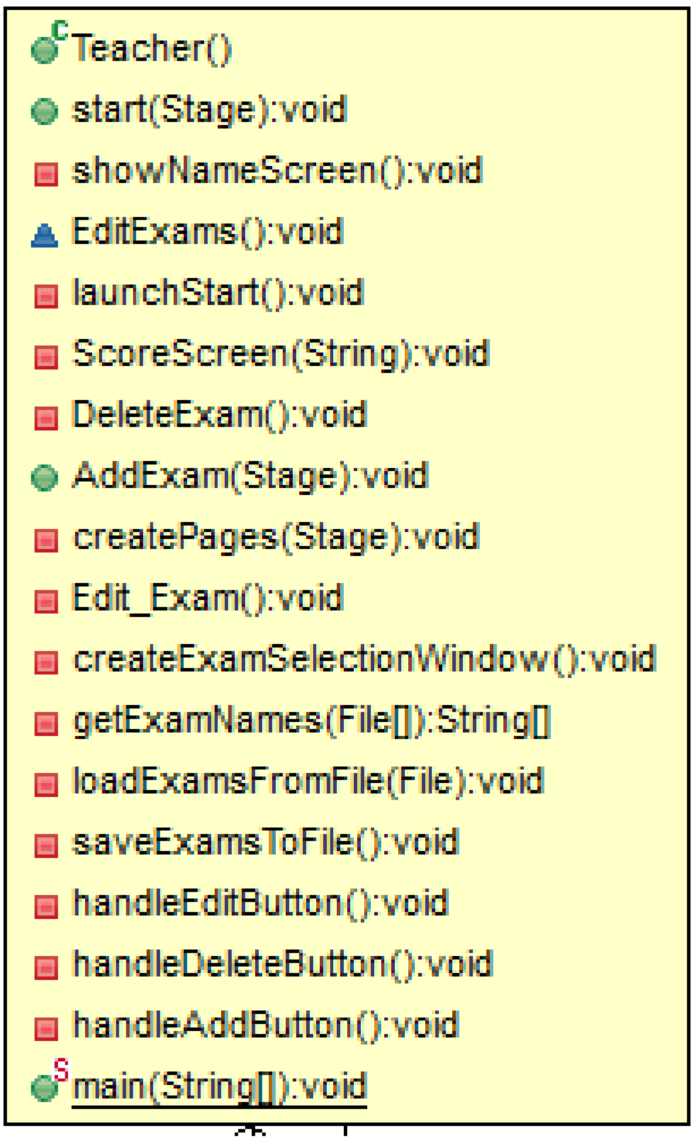


1	noha	Scores : 5
2	omar	Scores : 5
3	salma	Scores : 4
4	ahmed	Scores : 4
5	zeynab	Scores : 4

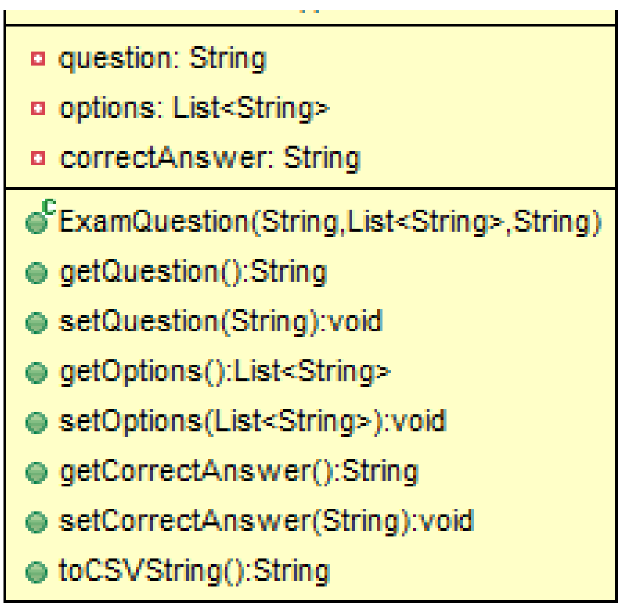
UML of Teacher Part



row: int
displayExams: Scene
noChoicesSpinner: Spinner<Integer>
numPagesSpinner: Spinner<Integer>
examTimeField: TextField
examNameField: TextField
markTextField: TextField
questionField: TextField
answerField: TextField
primaryStage: Stage
pageNum: int
totalPages: int
pages: Scene[]
initialScene: Scene
table: TableView<ExamQuestion>
examComboBox: ComboBox<String>
MAX_NUMBER_OF_OPTIONS: int
headers: String[]



Teacher()
start(Stage):void
showNameScreen():void
EditExams():void
launchStart():void
ScoreScreen(String):void
DeleteExam():void
AddExam(Stage):void
createPages(Stage):void
Edit_Exam():void
createExamSelectionWindow():void
getExamNames(File[]):String[]
loadExamsFromFile(File):void
saveExamsToFile():void
handleEditButton():void
handleDeleteButton():void
handleAddButton():void
main(String[]):void

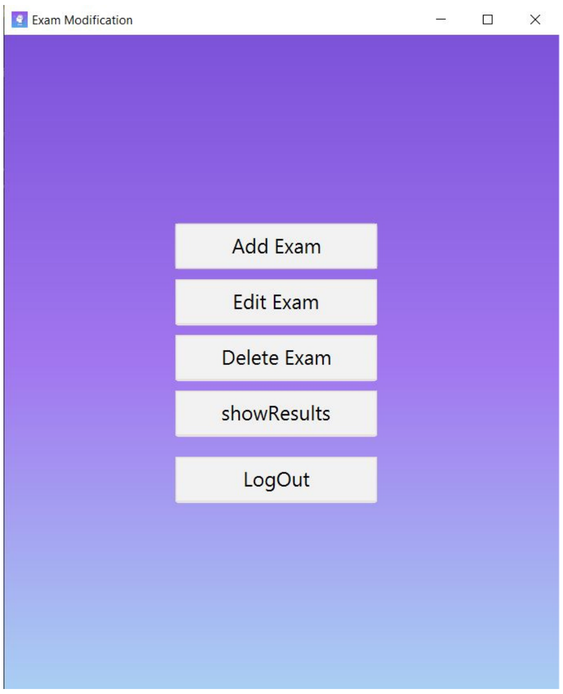


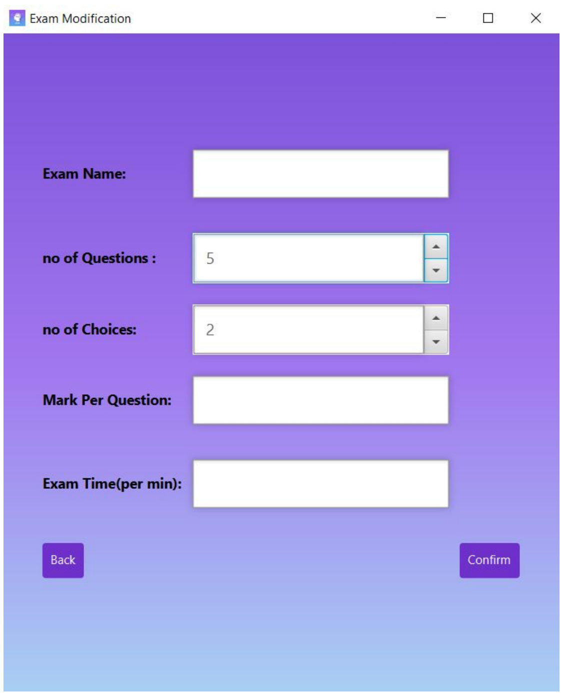
question: String
options: List<String>
correctAnswer: String
ExamQuestion(String,List<String>,String)
getQuestion():String
setQuestion(String):void
getOptions():List<String>
setOptions(List<String>):void
getCorrectAnswer():String
setCorrectAnswer(String):void
toCSVString():String

Explanation of Teacher Part

1. Get the number of choices from the `noChoicesSpinner`.
2. Create an array of scenes to store the pages.
3. For each page in the array:
* Create a vertical layout.
* Add a text node to display the error message.
* Add a text node with the text "Question " + (i + 1).
* Add a text field for the question.
* Add a text node with the text "Choices ".
* For each choice in the number of choices:
* Add a text field for the choice.
* Add a text node with the text "Correct Answer".
* Add a text field for the correct answer.
* Add a horizontal layout to navigate to the next and previous pages.
* Add a button to go to the previous page.
* If this is not the first page, add a button to go to the next page.
* If this is not the last page, add a button to save the page.
* Add the horizontal layout to the vertical layout.
* Create a scene from the vertical layout.
* Add the scene to the array of scenes.
4. Return the array of scenes.

The `createPages()` method creates a new scene for each page of the exam. The scene contains a text field for the question, text fields for the choices, and a text field for the correct answer. The scene also contains buttons to navigate to the previous and next pages, and a button to save the page.





Exam Name:
no of Questions : 5
no of Choices: 2
Mark Per Question:
Exam Time(per min):

1. Create the Exams directory if it does not already exist.
2. Create a new file in the Exams directory with the name of the exam.
3. Write the header of the file to the first line.

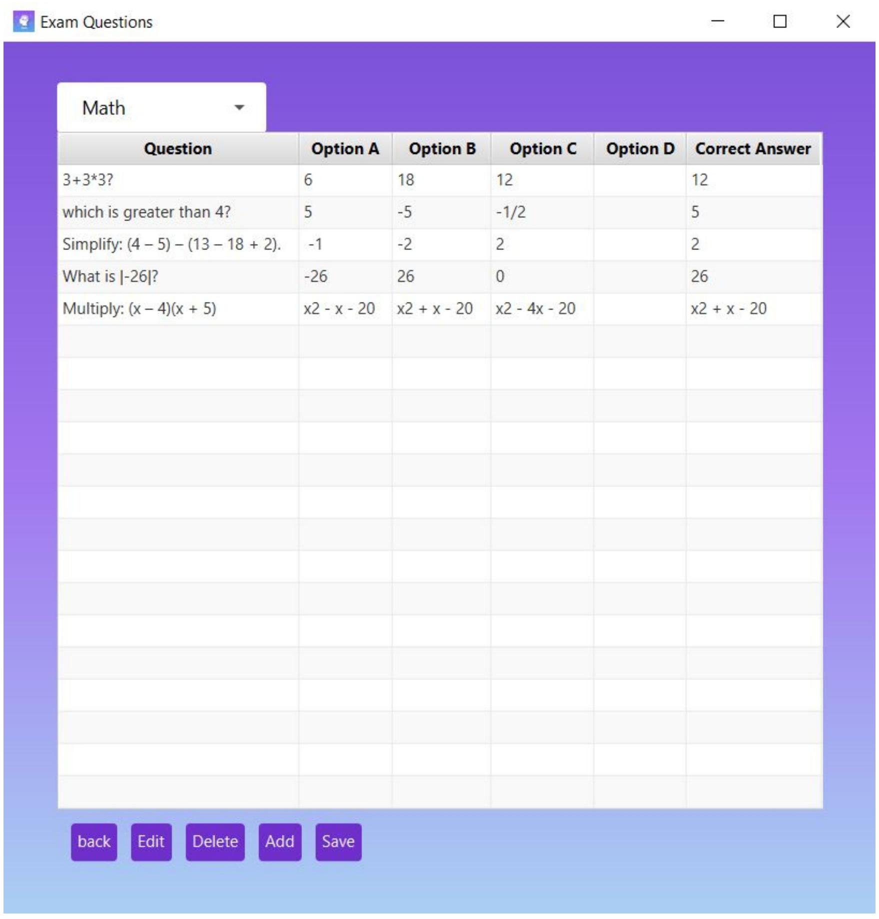
The header consists of the following columns:

* Question
* Option A
* Option B
* Option C
* Option D
* Correct Answer

4. Write each exam question to the file, one per line. The format of each line is as follows:

* Question
* Option A
* Option B
* Option C
* Option D
* Correct Answer

5. Close the file.



Question	Option A	Option B	Option C	Option D	Correct Answer
3+3*3?	6	18	12		12
which is greater than 4?	5	-5	-1/2		5
Simplify: (4 – 5) – (13 – 18 + 2).	-1	-2	2		2
What is |-26|?	-26	26	0		26
Multiply: (x – 4)(x + 5)	x2 - x - 20	x2 + x - 20	x2 - 4x - 20		x2 + x - 20

The `saveExamsToFile()` method is a useful method for saving the exam data to a file. The method takes care of all the details of saving the data, such as creating the directory, opening the file, and writing the data to the file. This allows the user to focus on creating the exam questions.

* The `handleEditButton()` method is called when the user clicks the "Edit" button.
* The method first gets the selected question from the table.
* If the selected question is not null, the method creates a dialog for editing the question.
* The dialog displays the current question and the answer options.
* The user can edit the question and the answer options in the dialog.
* If the user clicks OK, the method updates the question with the new values.
* The method then refreshes the table to show the updated question.

* The `handleDeleteButton()` method is called when the user clicks the "Delete" button.
* The method first gets the selected question from the table.
* If the selected question is not null, the method removes the question from the list of questions.
* The method then refreshes the table to show the updated list of questions.

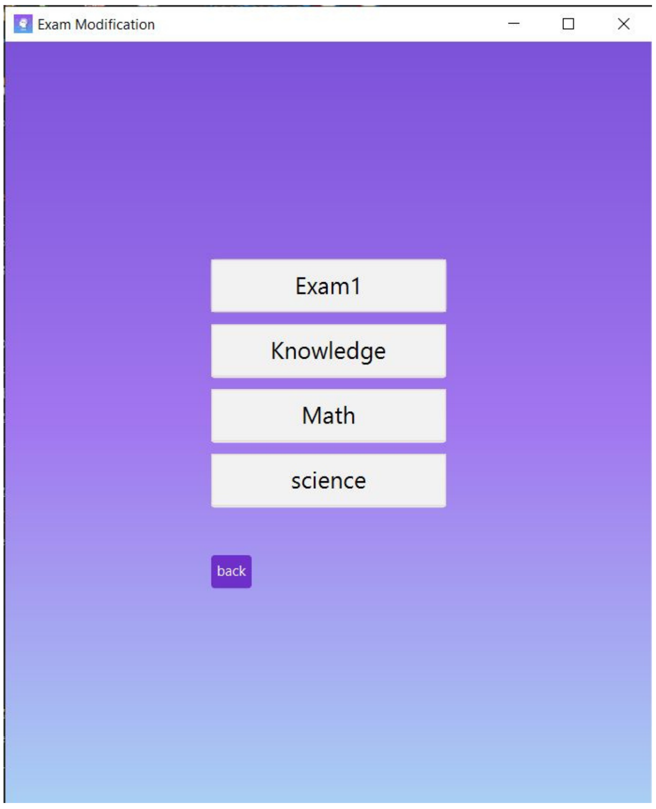
* The `handleAddButton()` method is called when the user clicks the "Add" button.
* The method first creates a dialog for adding a new question.
* The dialog displays a text field for the question and four text fields for the answer options.
* The user can enter a question and four answer options in the dialog.
* If the user clicks OK, the method creates a new question with the values entered by the user.
* The method then adds the new question to the list of questions.
* The method then refreshes the table to show the updated list of questions.

* The `ExamQuestion` class represents an exam question.
* The class has three properties:
  * question: The question of the exam question.
  * options: The answer options of the exam question.
  * correctAnswer: The correct answer of the exam question.
* The class has getters and setters for all three properties.
* The class also has a method called `toCSVString()`.
* The `toCSVString()` method converts the exam question to a CSV string.
* The CSV string can be easily saved to a file or sent to a database.

* The `main()` method is the entry point of the program.
* The `main()` method creates a new stage and loads the `ExamEditor` scene into the stage.
* The `ExamEditor` scene is the main scene of the program.
* The `ExamEditor` scene contains a table for displaying the exam questions, a toolbar for adding, editing, and deleting exam questions, and a button for saving the exam questions to a file.

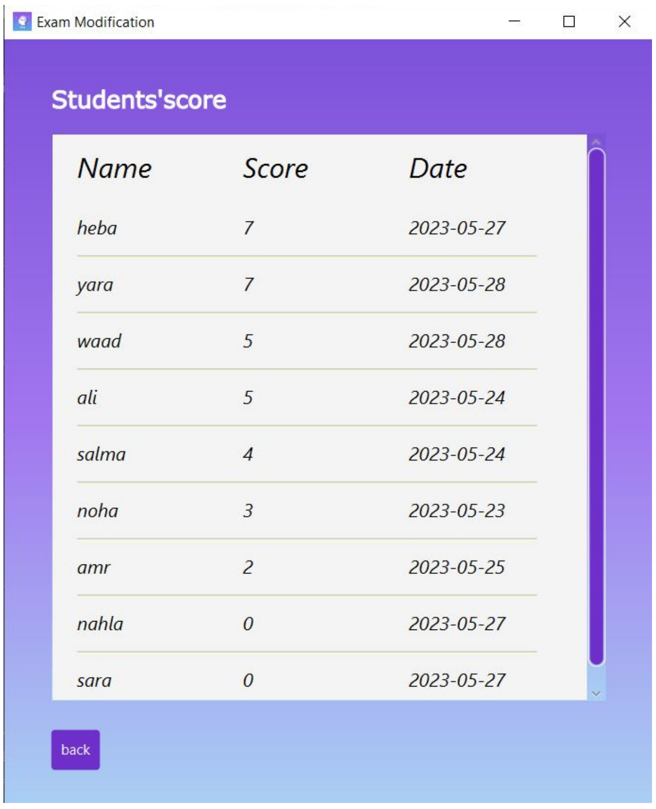
1. The `DeleteExam()` method is called when the user clicks the "Delete" button in the exam editor.
2. The method then gets a list of all the exams in the `Exams` directory.
3. For each exam in the list, the method creates a button with the name of the exam.

When the user clicks on a button to delete an exam, the `DeleteExam()` method is called again. The method then deletes the exam file from the `Exams` directory and the exam result from the exam result directory .



1. The `ScoreScreen()` method is called when the user clicks on the "Scores" button in the exam editor.

3. The method then gets a list of all the participants' scores from the CSV file.
4. For each participant, the method adds a row to the `GridPane` with the participant's name, score, and date.
5. The method then creates a button to go back to the exam editor.
6. The method then adds the back button to the `GridPane`.



Name	Score	Date
heba	7	2023-05-27
yara	7	2023-05-28
waad	5	2023-05-28
ali	5	2023-05-24
salma	4	2023-05-24
noha	3	2023-05-23
amr	2	2023-05-25
nahla	0	2023-05-27
sara	0	2023-05-27

ERRORS

1 Linking Between Packages

Import other packages in the same package and adding a stage to the new variable
which contains the main file in the other packages .

2 Writing Upper case letter Problem

Using the method toLowerCase() which ensures that the string will always be
lowecase .

3 Entering the exam even if the user had entered it before

Checking the result folder for the choosen exam and
searching for the user score to know if he had done this exam before .

4 Linking between the pages and the answers

Using Indexing to know the current page and by linking the question and
the user answer in a hashmap we could easily retrieve the user answer .

5 Not Getting the correct answer even after getting it from hashmap

Using Trim() method to ensure that there is no
spaces in the choosen & correct answer which ensures that the condition will be executed correctly .

6 The size of the labels in the gridpane were bigger than the scene size

we used vertical & horizontal scrollbars .

7 Facing a problem with user input

making a validation class to make sure that the textfileds don't have the right data type .

8 Using a fixed header variable

resetting the header value every time a new exam is chosen .

IDEAS TO DEVELOP PROGRAM

1 **Make a sign up option**

2 **Add an admin option**

that has more permissions than the teacher like accepting students requests .

3 **Make the teacher able to edit instructions screen**

4 **Make the teacher able to make the students join the exam again .**

5 **Add a database like SQL , a sever and security**

6 **Add an experimental exam**

as the student can choose the type of exam and the difficulty level .

REFERENCES :

1- We used AI Chatbots like :

CHATGPT

Bing **BING AI**

To help us to write the code as well as searching .

2- We used **AI Bard** from Google

and **Draw.io** Website

to make the Flowchart .

3- We used **Eclipse** Software

to make the Uml .

4- We were coding with **NetBeans IDE** Software .