

# **COMP 3111:**

## **Source Code Documentation**

Authors:

Rishab Sukhlecha (20727721)  
Gonzalo Carretero (21158252)  
Richard Blazek (21160047)

Group: 46

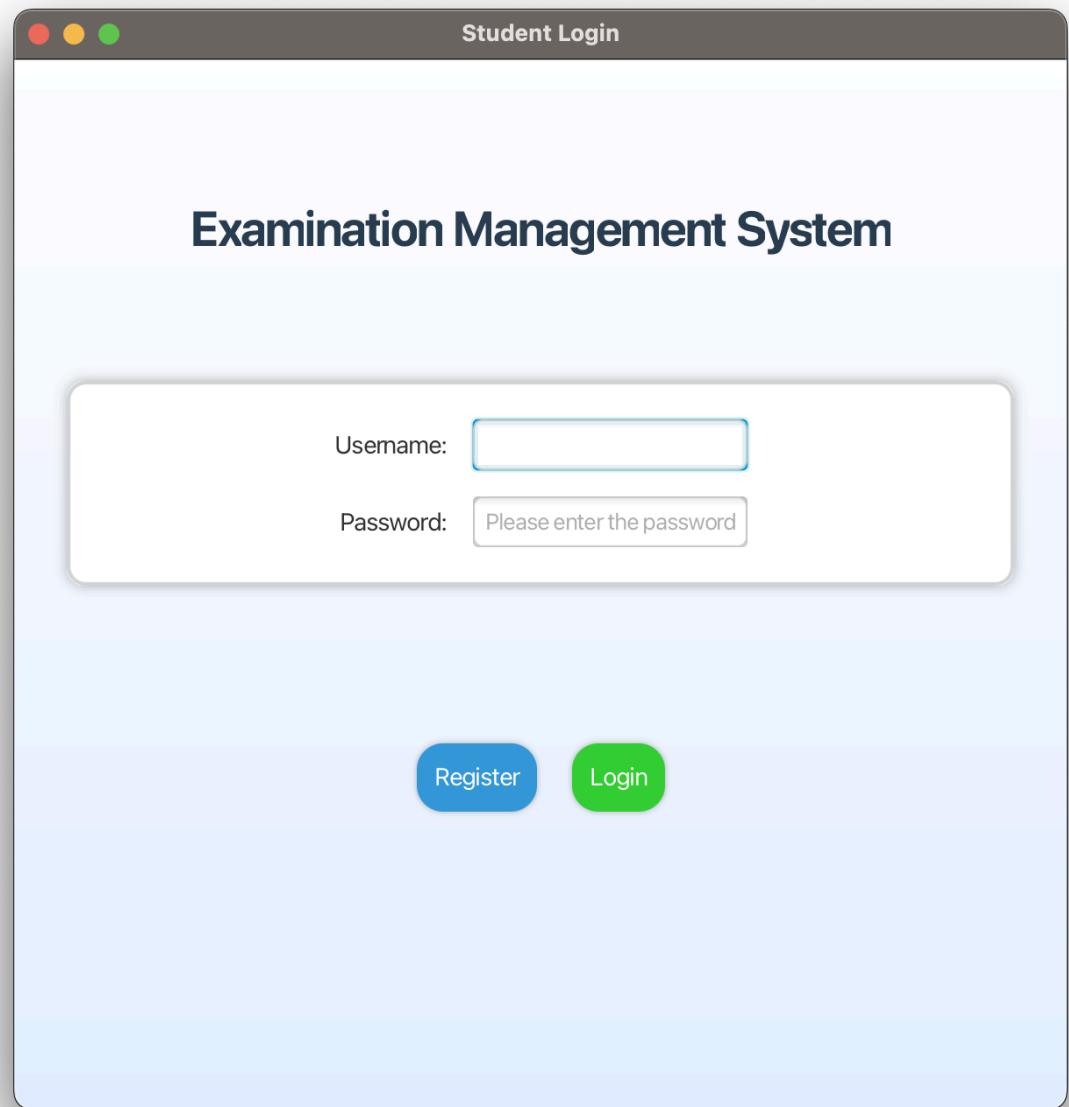
Date: 21-11-2024

Screenshots of the execution of the application showing sample inputs and outputs (Task 1):

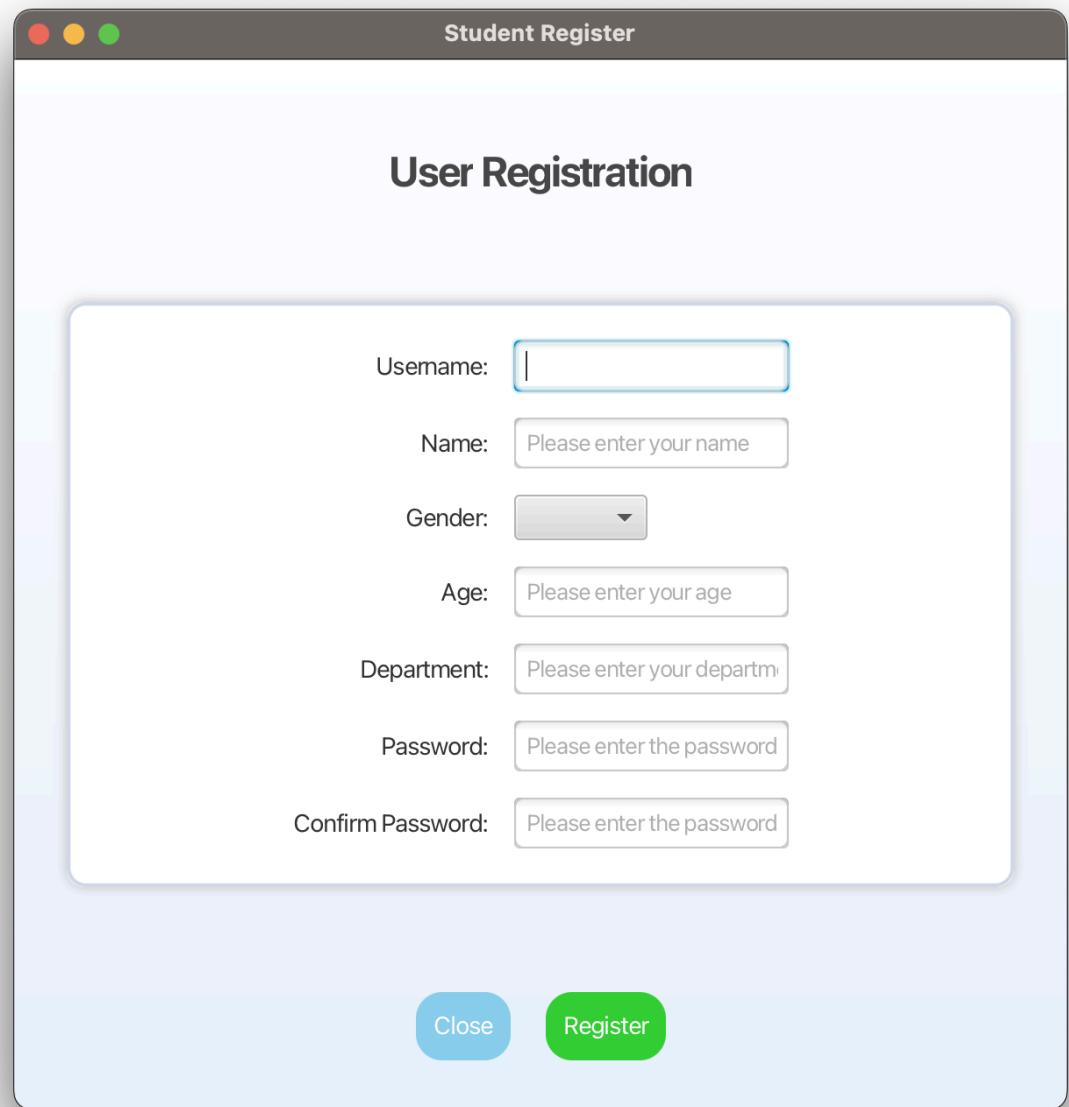
1. Open the app



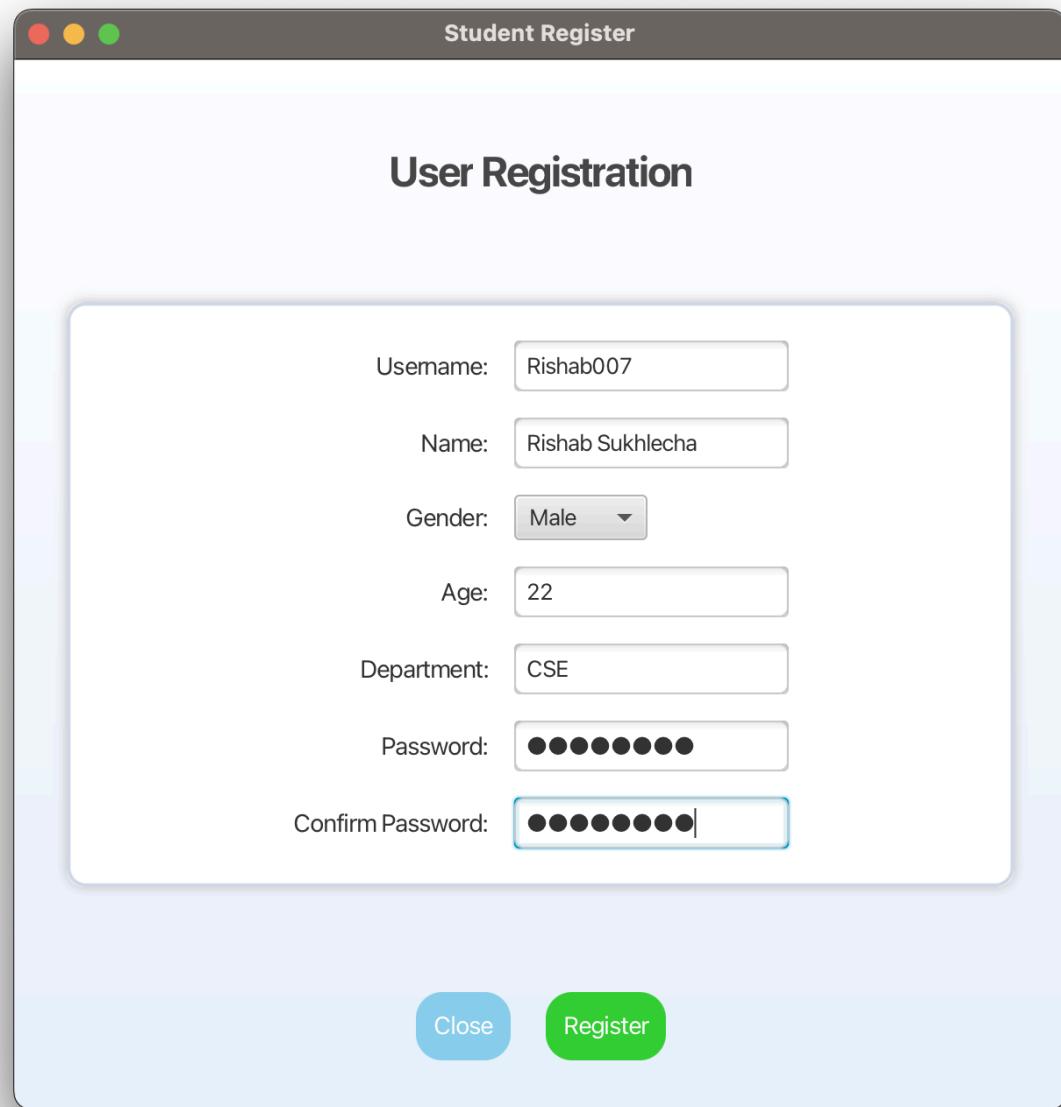
2. Click "Student Login"



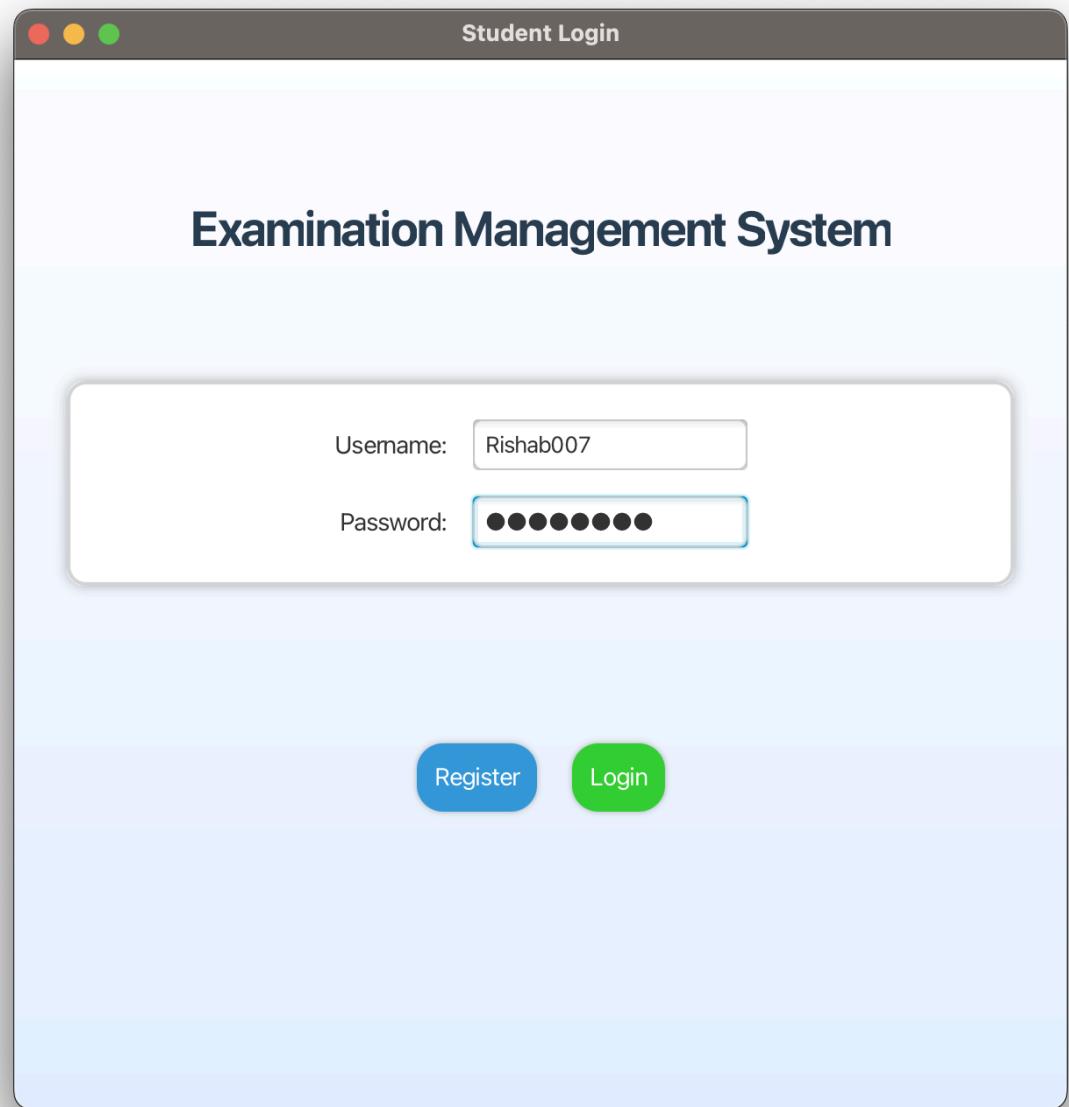
3. Click “Register”



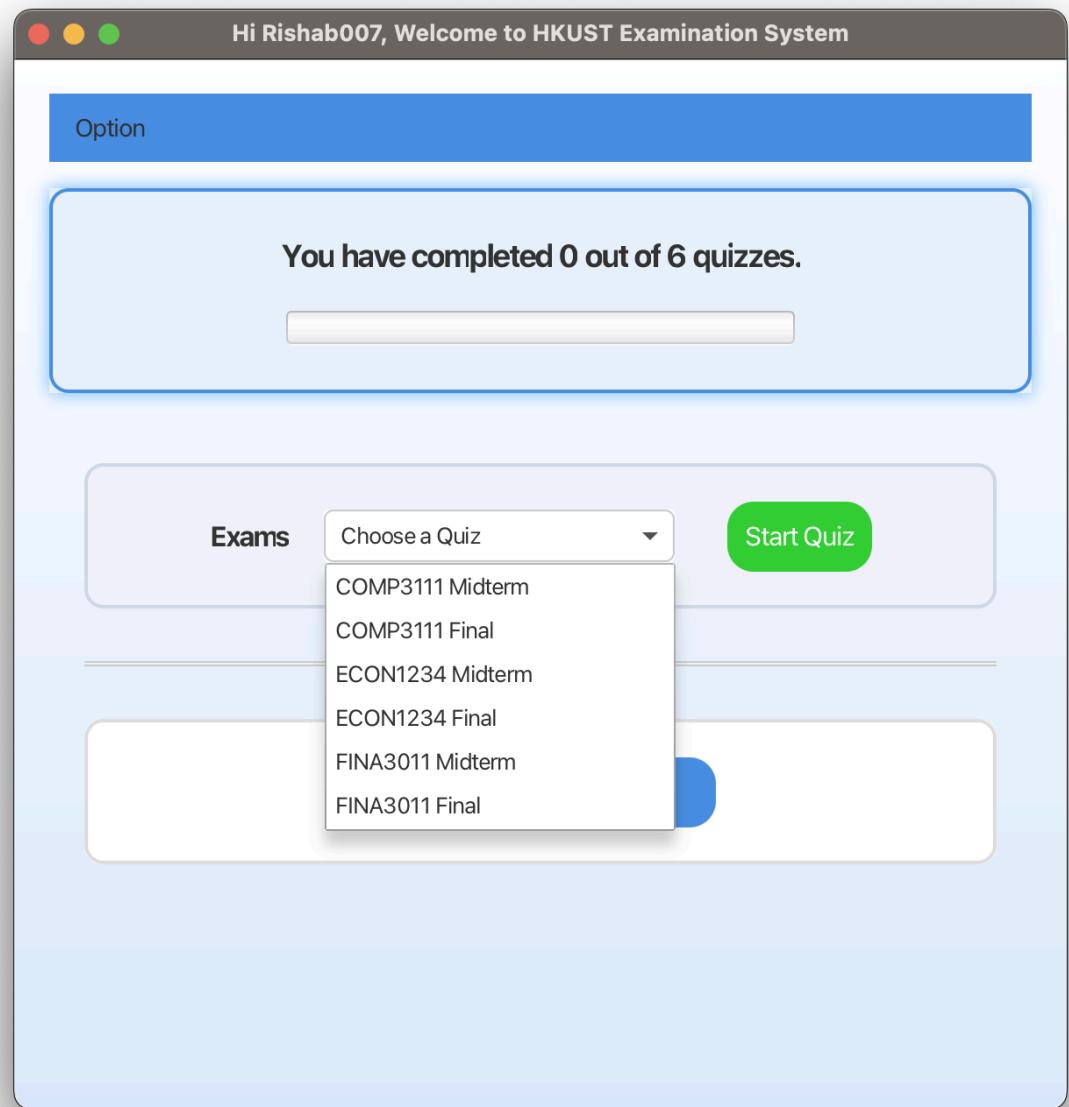
4. Enter Registration Information and click “Register”



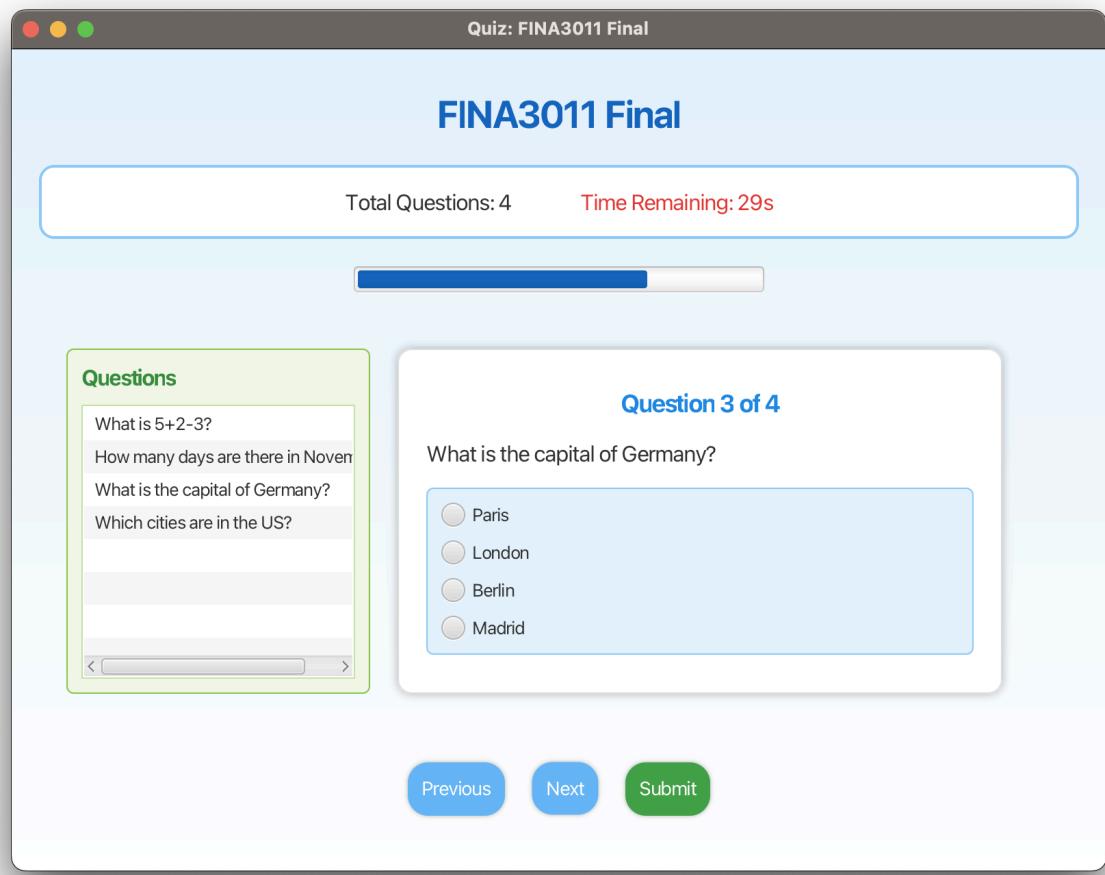
5. Enter Username and Password just created and click "Login"



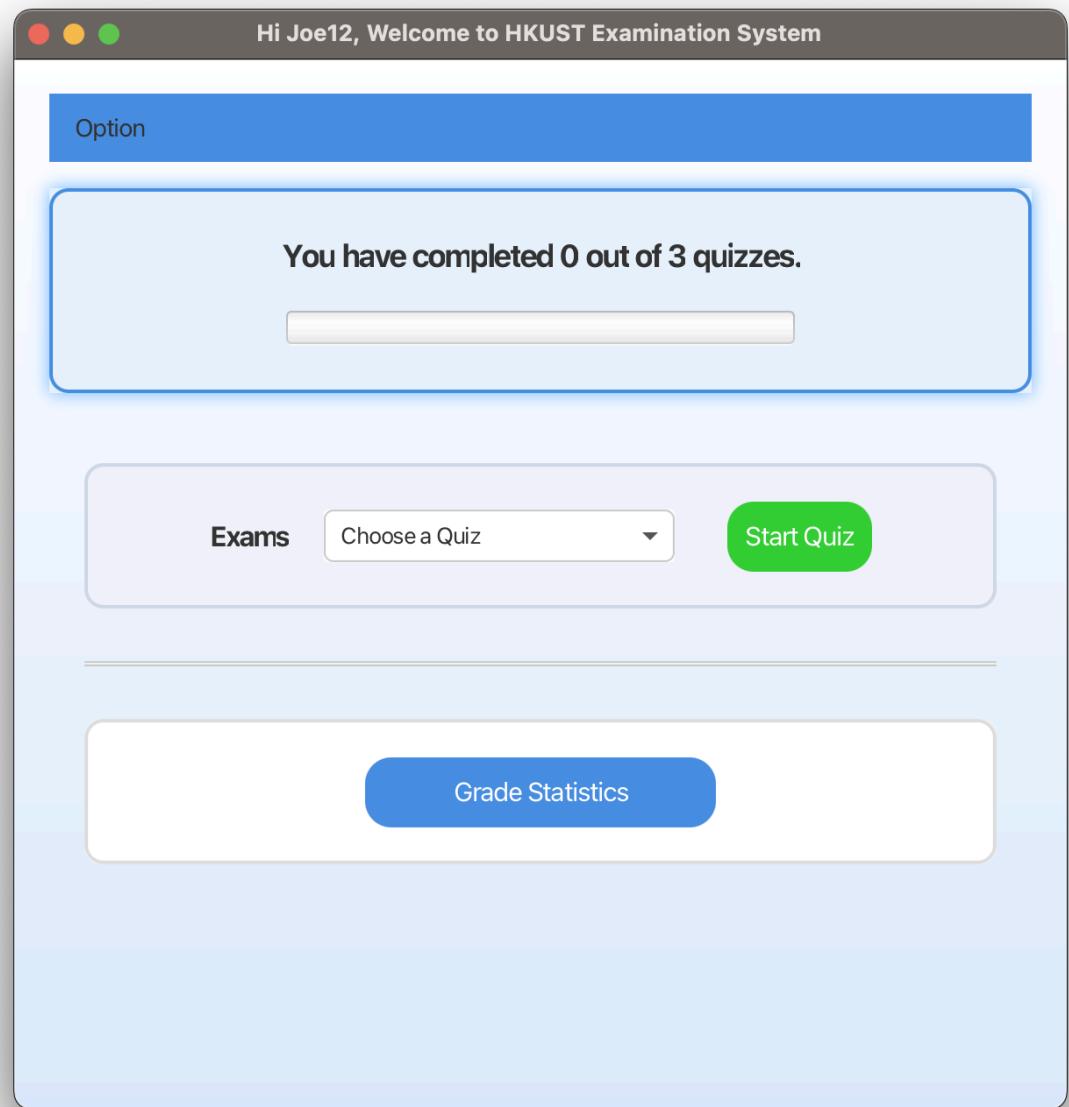
6. Pick a exam to do and click “Start Quiz”



7. Answer questions



8. Click "Grade Statistics"



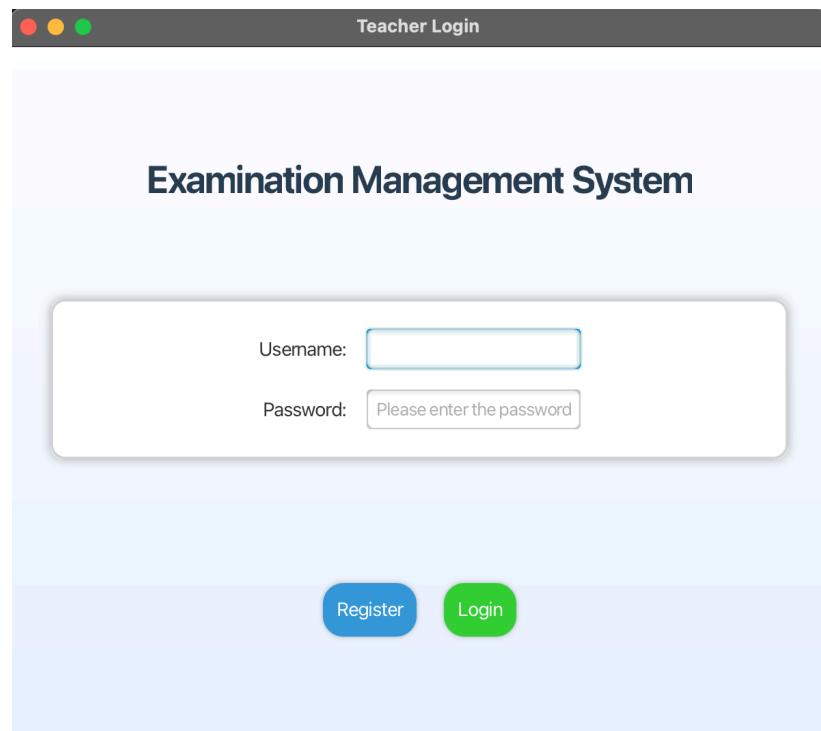
9. Pick a course and click “Filter”
10. Click “Reset”

Screenshots of the execution of the application showing sample inputs and outputs (Task 2):

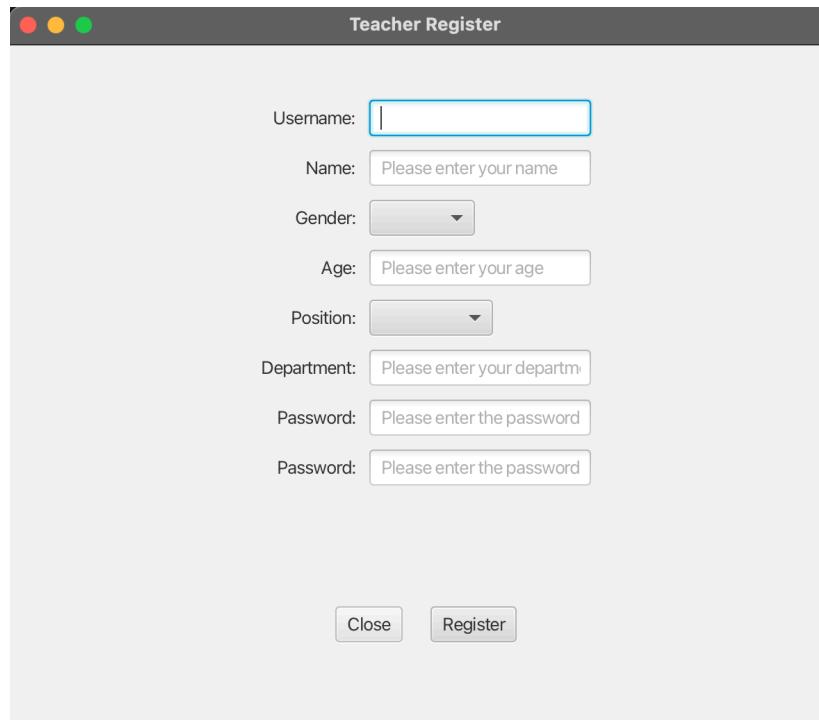
- Open the app:



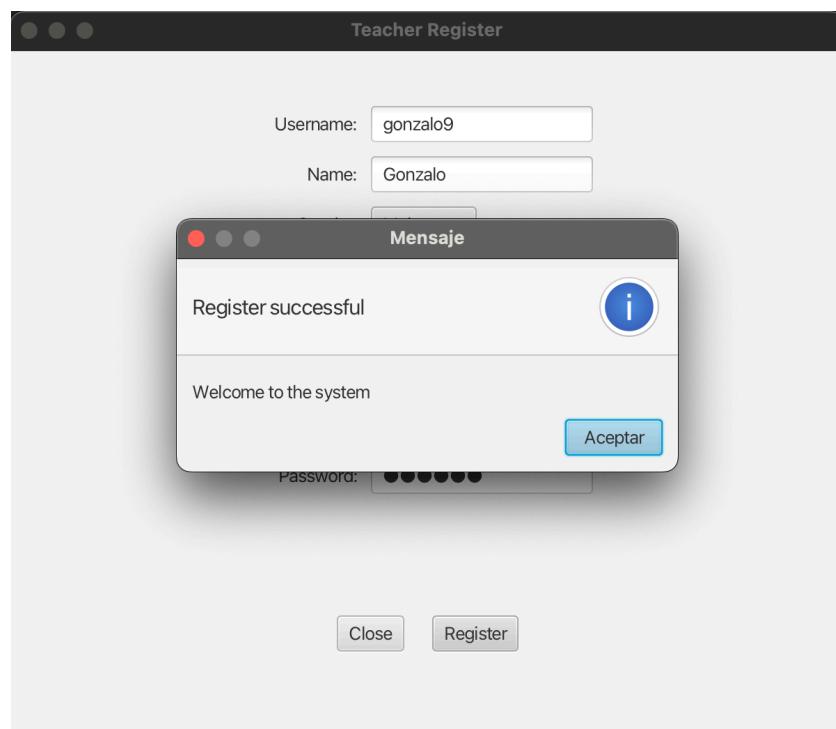
- Click on Teacher Login button:



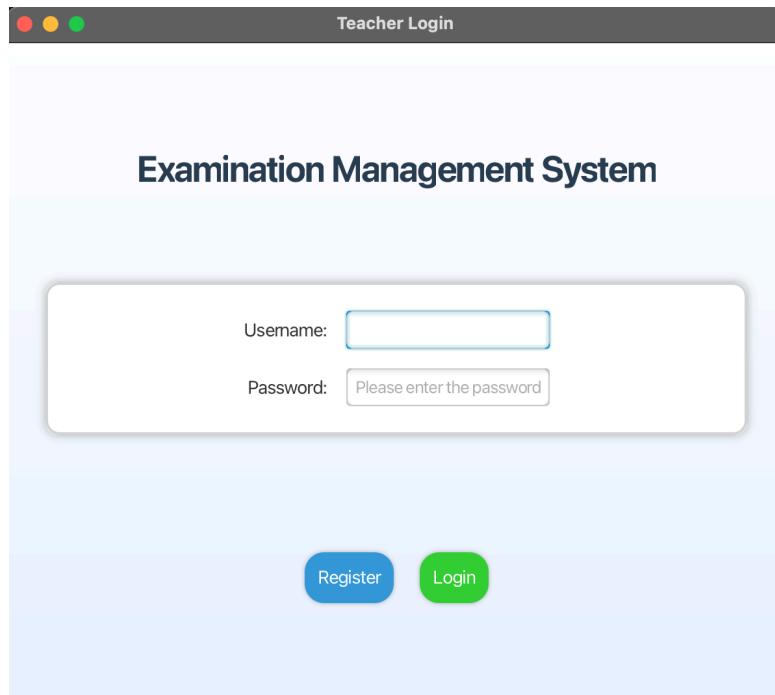
- Click Register button:



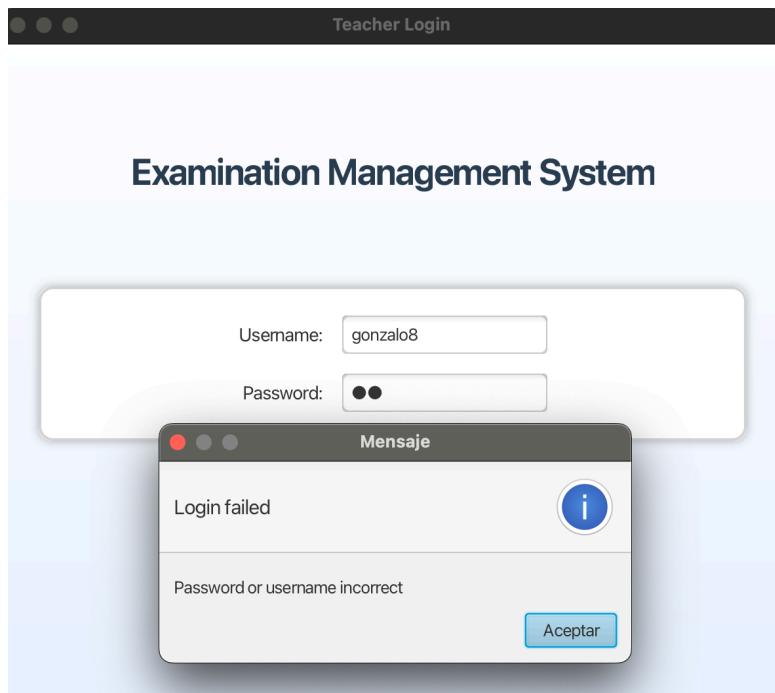
- After filing the inputs and clicking register:



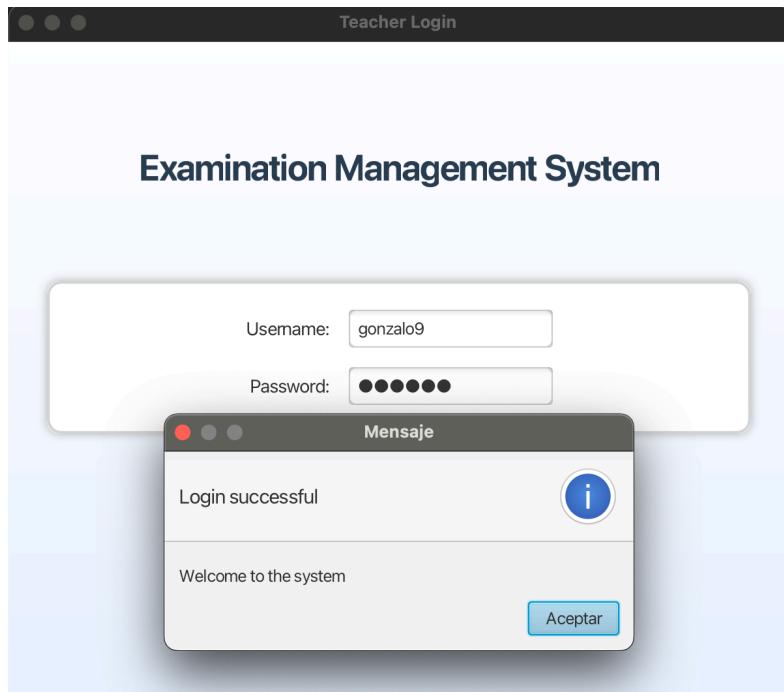
- After closing the pop up goes back to the login page. This same action happens when instead of clicking register button you click the close button:



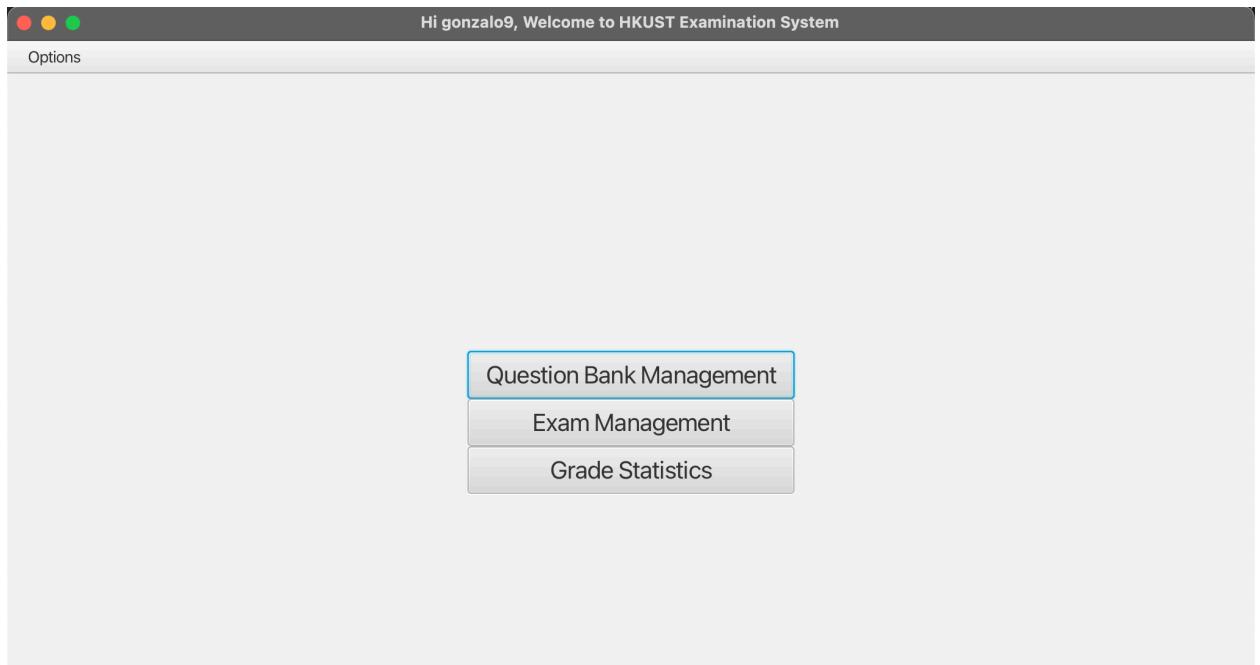
- If we enter an invalid username or password then an error message pops up:



- If a valid username and password are entered and click Login button:



- After closing the pop up:



1. After clicking the Question Bank Management button:

Question	Type	Score	Reset	Filter
Question	Option A	Option B	...	
What does the "Big O" notation describe?	The exact time complexity of an algorithm	The worst-case time complexity of an algorithm	T1	Type
In a binary search tree, what is the maximum number of children a node can have?	One	Two	T1	Option A
Which of the following are examples of sorting algorithms with $O(n \log n)$ average time complexity? (Choose all that apply)	Merge Sort	Quick Sort	B1	
Which of the following are valid ways to handle a "divide by zero" error in programming? (Choose all that apply)	Use a try-catch block (or equivalent in the language)	Ignore the error and proceed with the division	C	Option B
				Option C
				Option D
				Answer
				Score
				Add Update

(The table is scrollable in order to see all the columns)

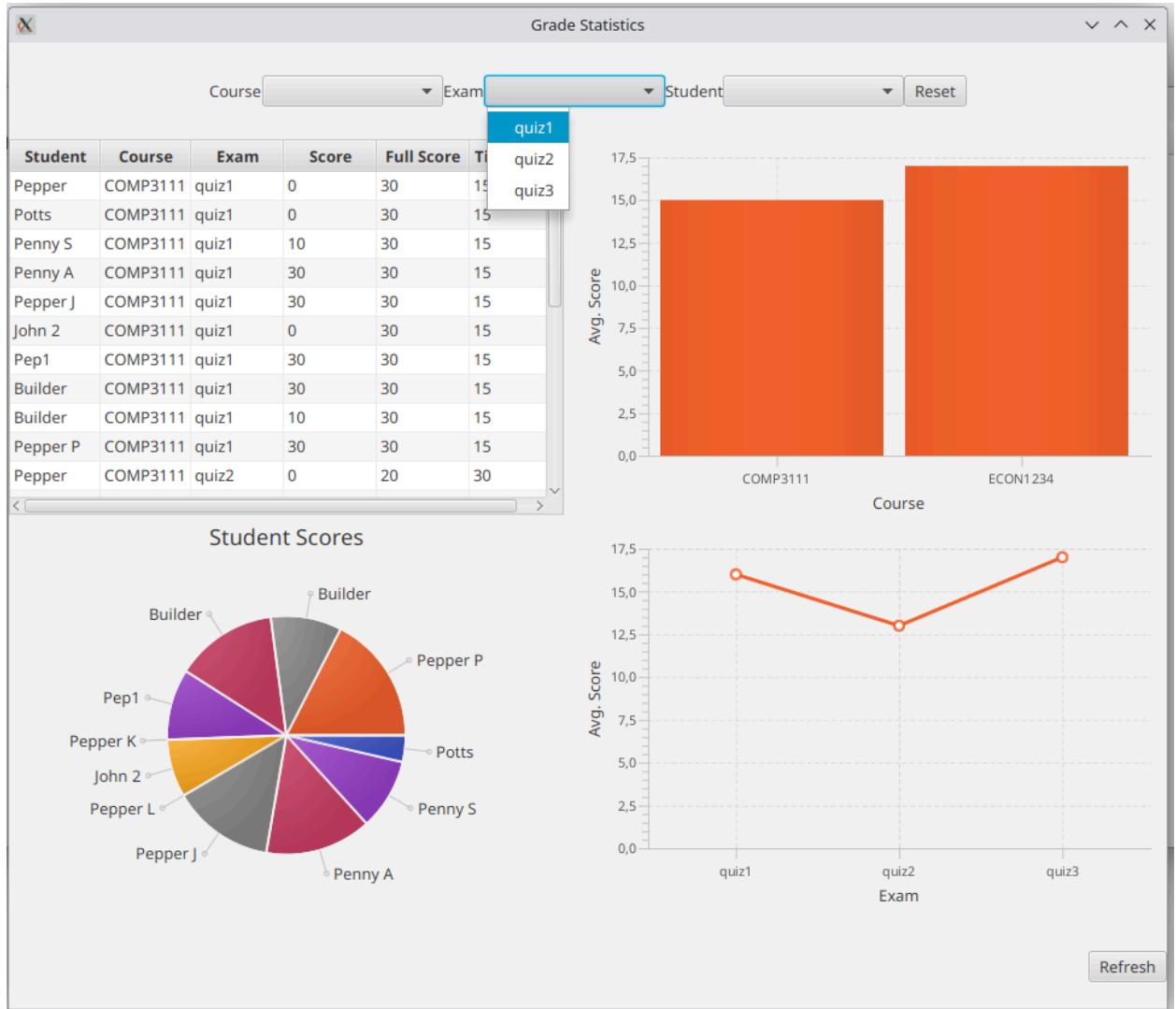
The overall functionality of this window and its buttons will be shown in the complementary video provided.

2. After clicking the Exam Management button:

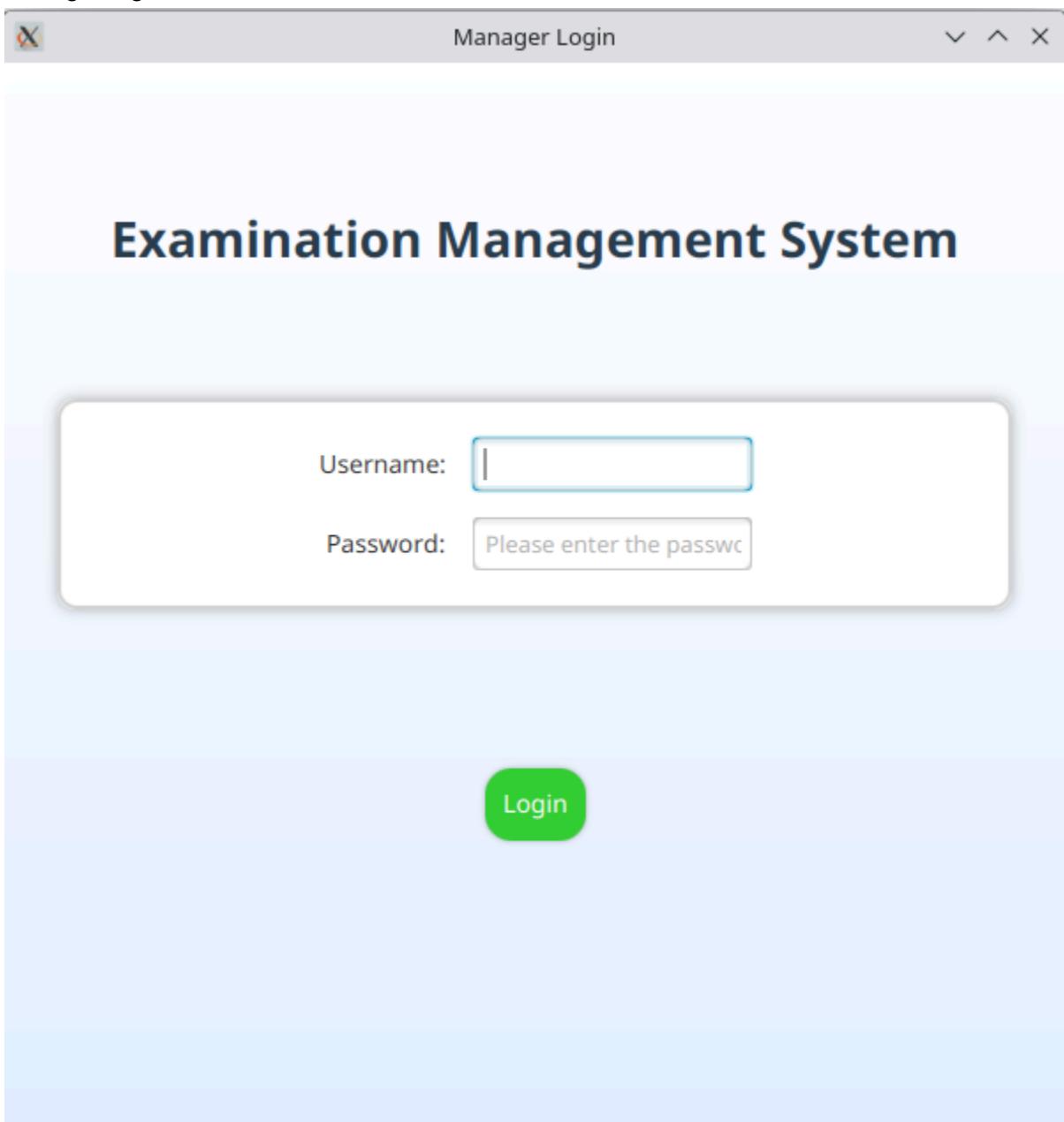
The overall functionality of this window and its buttons will be shown in the complementary video provided.

Screenshots of the execution of the application showing sample inputs and outputs (Task 3):

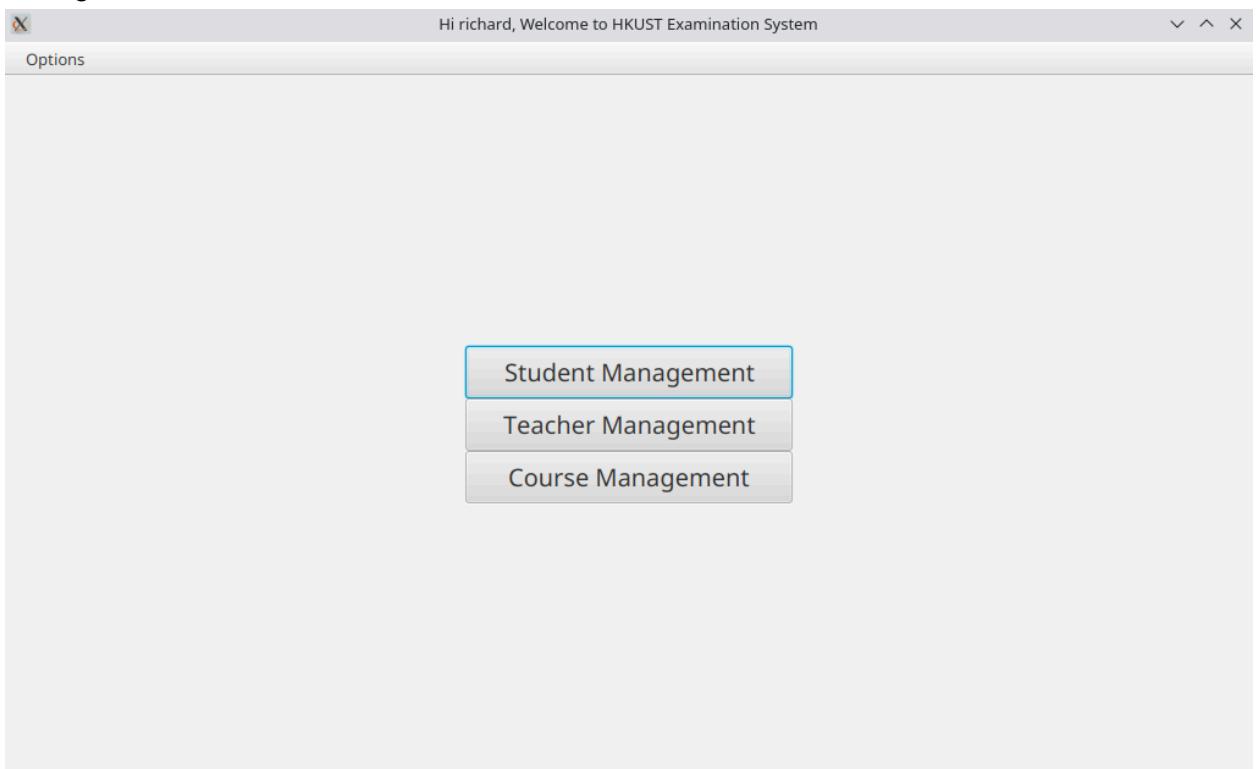
1. Teacher statistics:



2. Manager login:



3. Manager view:



#### 4. Student management:

## 5. Teacher management:

## 6. Course management:

## Report on Unit Testing:

The screenshot shows a unit testing interface with a toolbar at the top labeled "Run" and "Test model". Below the toolbar is a list of test classes and their execution times. To the right of the list is a summary of the test results.

Test Class	Time
examsystem (comp3111)	119 ms
> ExamTest	10 ms
> StudentTest	
> ManagerDbTest	
> CourseDbTest	
> StudentDbTest	
> DataSerializerTest	90 ms
> serializerTest()	90 ms
> TeacherTest	
> DataStorageTest	19 ms
> GradeTest	

Summary on the right:

- Tests passed: 36 of 36 tests – 119 ms
- /home/richard/.jdks/corretto-21.0.5/bin/java ...
- Process finished with exit code 0

## Report on Coverage Test:

Coverage	Test model	⋮		
File	Up	Down	Left	Right
Element	Class, %	Method, %	Line, %	Branch, %
comp3111.examsystem	41% (19/46)	48% (171/355)	31% (499/1567)	43% (150/346)
controller	0% (0/26)	0% (0/182)	0% (0/1061)	0% (0/196)
Main	0% (0/1)	0% (0/2)	0% (0/7)	100% (0/0)
model	100% (19/19)	100% (171/171)	100% (499/499)	100% (150/150)
Manager	100% (1/1)	100% (3/3)	100% (5/5)	100% (0/0)
Question	100% (1/1)	100% (12/12)	100% (21/21)	100% (0/0)
Teacher	100% (1/1)	100% (8/8)	100% (15/15)	100% (0/0)
DataStorage	100% (1/1)	100% (3/3)	100% (15/15)	100% (0/0)
DataCollection	100% (1/1)	100% (8/8)	100% (15/15)	100% (0/0)
Gender	100% (1/1)	100% (2/2)	100% (3/3)	100% (0/0)
Student	100% (1/1)	100% (8/8)	100% (15/15)	100% (0/0)
Type	100% (1/1)	100% (2/2)	100% (3/3)	100% (0/0)
Grade	100% (1/1)	100% (6/6)	100% (9/9)	100% (0/0)
Course	100% (1/1)	100% (5/5)	100% (9/9)	100% (0/0)
ManagerDb	100% (1/1)	100% (5/5)	100% (8/8)	100% (2/2)
Exam	100% (1/1)	100% (15/15)	100% (25/25)	100% (2/2)
GradeDb	100% (1/1)	100% (6/6)	100% (22/22)	100% (14/14)
TeacherDb	100% (1/1)	100% (12/12)	100% (22/22)	100% (16/16)
ExamDb	100% (1/1)	100% (11/11)	100% (27/27)	100% (16/16)
CourseDb	100% (1/1)	100% (11/11)	100% (24/24)	100% (16/16)
StudentDb	100% (1/1)	100% (12/12)	100% (22/22)	100% (22/22)
QuestionDb	100% (1/1)	100% (11/11)	100% (21/21)	100% (22/22)
DataSerializer	100% (1/1)	100% (31/31)	100% (218/218)	100% (40/40)

With JACOCO:

The screenshot shows a browser window displaying a JaCoCo coverage report for the 'ExamSystem' project. The report is a table with the following columns: Element, Missed Instructions, Cov., Missed Branches, Cov., Cxty, Missed Lines, Lines, Missed Methods, Methods, and Missed Classes. The table includes a header row and several data rows corresponding to different classes. The last row is a summary for 'Total'. The 'Lines' column shows the total number of lines of code, and the 'Cov.' column shows the percentage of lines covered. The 'Cxty' column shows the number of cyclomatic complexity units. The 'Methods' column shows the total number of methods, and the 'Classes' column shows the total number of classes. The 'Sessions' tab is visible at the top right of the browser window.

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Cxty	Missed Lines	Lines	Missed Methods	Methods	Missed Classes	Classes
comp3111.examsystem.controller.student		0 %		0 %	125	125	470	470	68	68	6
comp3111.examsystem.controller.teacher		5 %		0 %	105	120	424	461	70	85	7
comp3111.examsystem.controller.manager		0 %		0 %	64	64	284	284	57	57	8
comp3111.examsystem.controller		1 %		0 %	55	57	181	184	51	53	10
comp3111.examsystem		0 %		n/a	3	3	11	11	3	3	1
comp3111.examsystem.model		100 %		97 %	5	266	0	545	0	173	0
Total	5.751 of 8.608	33 %	211 of 392	46 %	357	635	1.370	1.955	249	439	32

Created with JaCoCo 0.8.10 202304240956

Documentation on the implemented tasks using JavaDoc:

The .html files can be found in the team repo. You may find directions on how to open them inside the Readme file!

**Supplementary Notes:**

Video is in the GitHub Repo. There is one video for each task that we have prepared. Thank you!