Criteria for grading

14 points in total - Minimum to pass 8, according to 5 dimensions:

1) Functionality (Coverage of the requirements)

A: all functions implemented
B: (-1 point) if a minor functional specification or requirement is not implemented or not correctly implemented (logical errors)
C: (-2 points) if 2 functions or requirements are not implemented or not correctly implemented
D: (-3 points) if 3 functions or requirements are not implemented or not correctly implemented
Fail: if many functions or requirements are not implemented and the project is very much smaller than the requested one

2) GUI usability

A: if the GUI is well designed
B: (-1 point) if there are some small issues, such as labels that cannot be read easily or strange names.
C: (-2 points) When there are small issues and it is not immediate to understand how a function can be executed through the GUI.
D: (-3 points) If there are more severe problems, such as windows that are popping up without content of labels that are not shown
Fail: severe problems such as the GUI blocks or you cannot go back to the beginning of the interaction state

3) Errors and exceptions

A: if no run time errors or exceptions
B: (-1 point) If you got one or more run time error or exception but the application is still running.
C: (-2 points) If many errors or exceptions are raised.
Fail: There are many exceptions and the application cannot be used.

4) Code structure

A if the code is well written and well structured (e.g., a reasonable number of classes is used and the naming conventions are applied)
B (-1 point) if the student has not created a reasonable set of classes or the code is messy with strange names and unclear method calls

5) Fix to Start:

Fail: If the application does not start or some necessary files are not found.

In addition to the points from the project, the students may get up to 2 points from the lab assignments (explained in the lab 1) up to 14 points.

Formula for the project evaluation:
Project mark = 14 - negative points
Final mark = min(Project mark + lab points, 14)

Example:
A student gets the following scores:
Functionality: A
User Interface: B(-1)
Exception: C(-2)
Quality of the Code: A

14 - 3 = 11
Final mark: 11 + 1 from lab assignments = 12

---------------------------------------------

Student ID: 13533

Functionality: The scoreboard is not ranking users but sessions. If I enter twice with the same name I am considered two different persons. (-1)
User Interface: A
Exception: None
Quality of the Code: A

14 - 1 = 13
Final mark: 13 + 0 from lab assignments = 13
========

Student ID: 13480

Functionality:
The timer is not shown in the user interface - one must open the scoreboard (this is not implementing the requested functionality) (-1). The scoreboard is not showing the aggregated percentage of success - it shows the percentage of success for each tested grid - and the users are not ranked according to the score (-1). There is no possibility to force the system to present only questions that the user has already correctly replied (-1)
User Interface: A button is supposed to show the top 5 more difficult questions, but it shows all the questions with their error rate - in principle this is another mistake. (-1)
Exception: I found the same question in two cells (-1).
Quality of the Code: OK

No Labs.
14 - 5 = 9
Final mark: 9 + 0 = 9
========

Student ID: 13208

Functionality: Even if I click on “visualize only wrong answers” the system propose again questions that I have replied correctly (-1). The scoreboard function is not implemented (-1). The top 5 most difficult questions are not shown (-1).
User Interface: The possible replies and the entered replies are not carefully indented. After the user has entered the nickname the first time (to start the application) there is no need to ask it again when the “test” button is pressed. (-1)

Exception: NONE
Quality of the Code: OK
14 - 4 = 10
Final mark: 10 + 2 from lab assignments = 12
========

Student ID: 13455

Functionality:  
The system in some cases can show the same question twice. (-1) The statistics of correctly replied questions is not updated. (-1) The scoreboard function is not implemented. (-1) There is no possibility to force the system to present only questions that the user has not already correctly replied (-1). FAIL

User Interface:  
The field used to enter the user name is excessively big for the given task. When you click on “new quiz” the window jumps on another place of the screen (upper left) and the questions are not shown until you click in the middle part of the window. The about menu function does not show anything about the size of the program. The correct answer is shown when you ask the system to check the entered replies, but you do not see anymore the answer that you entered. (-2)

Exception: I got one exception. (-1)
Quality of the Code: OK

Labs 2.
FAIL
========

Student ID: 13546

Functionality:  
The correct answer of a question is not shown when the system is requested to test the answers (-1). The top 5 more difficult questions are not shown (-1). The scoreboard function is not implemented (-1). There is no check box to load only the questions that the user replied incorrectly. (-1) The timer is not reset when you start a new grid of questions (-1). FAIL

User Interface:  
The possible answers for a question are always the same, so it is not possible to use this application. If you change the answer, then the user interface show two answers. FAIL
Exception: NO
Quality of the Code: n.a.

Lab 2.
FAIL
========

Student ID: 13642

Functionality:  
The list of top 5 more difficult questions is not shown (-1).

User Interface:  
The idea that one has to click a radio button (of the question) to show the question in another area of the window is not intuitive. The “not correct” checkbox has an unclear meaning. In the requirements it was asked that when a checkbox is clicked the system must not ask again a question that was replied correctly. (-2)
Exception: NONE
Quality of the Code: OK

\[ 14 - 3 = 11 \]
Final mark: 11 + 2 from lab assignments = 13

Student ID: 13453

Functionality: OK
User Interface: OK
Exception: NONE
Quality of the Code: OK

\[ 14 - 0 = 14 \]
Final mark: 14 + 2 from lab assignments = 14