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: 13546
Functionality: The sorting by category is not correctly implemented: you implemented a filter by category (-1). The cart is not cleaned after checkout (-1). The cart contains duplicate items if you add the same item twice (-1). I have added an item to the cart but this is not present. You must exit and start again the application to see the cart (-1). Search functionality is case sensitive (-1).
User Interface: After you have searched for an item that contains in the description a given string it is not clear how to go back to the display of all the items (you have to search for an empty string) (-1). The buttons to decrease the quantity of an item is not visible (-1). After the checkout the application is exiting (-1). The user should be able to go to next/previous result page, but this is not implemented (-1).
Exception: No exception
Quality of the Code: OK
Lab 2.
FAIL Too many functions are not implemented

Student ID: 13282
Functionality: OK
User Interface: The "previous" button should not be visible if there is no previous page. (-1)
Exception: NONE
Quality of the Code: OK

14 - 1 = 13
Final mark: 13 + 2 from lab assignments = 14