Assessment model (Rubrics) Project Databases: Assignment 2

Conditional requirements	The group has submitted <u>one</u> ZIP file containing the Visual Studio solution and a text file with the division of tasks (who did which part of the assignment).

Variant A: View all students

Individual contribution (80%)		Criterion	Points
The student should have developed part of the application that displays a list of students (retrieved from the database).	Unsatisfactory	There is no list of items or there is a static list (not retrieved from the database).	0
	Weak	A list of items (retrieved from the database) is displayed[1].	40
	Satisfactory	A list of items (retrieved from the database) is displayed. The list of items is refreshed each time the menu item is clicked (displayed items remain the same). The items are displayed in a structured way (with headings for all fields except ids).	60
	Good	A list of items (retrieved from the database) is displayed. The list of items is refreshed each time the menu item is clicked (displayed items remain the same). The items are displayed in a structured way (with headings for all fields except ids). The list is using Bootstrap styling (at least for the header and to add a hover effect).	80
	Excellent	A list of items (retrieved from the database) is displayed. The list of items is refreshed each time the menu item is clicked (displayed items remain the same). The items are displayed in a structured way (with headings for all fields except ids). The list is using Bootstrap styling (at least for the header and to add a hover effect). The list can be filtered by last name (so only students with a certain last name are displayed).	100

[1] the list of students should be ordered by last name

Variant B: View all lecturers

Individual contribution (80%)		Criterion	Points
The student should have developed part of the application that displays a list of lecturers (retrieved from the database).	Unsatisfactory	There is no list of items or there is a static list (not retrieved from the database).	0
	Weak	A list of items (retrieved from the database) is displayed[1].	40
	Satisfactory	A list of items (retrieved from the database) is displayed. The list of items is refreshed each time the menu item is clicked (displayed items remain the same). The items are displayed in a structured way (with headings for all fields except ids).	60
	Good	A list of items (retrieved from the database) is displayed. The list of items is refreshed each time the menu item is clicked (displayed items remain the same). The items are displayed in a structured way (with headings for all fields except ids). The list is using Bootstrap styling (at least for the header and to add a hover effect).	80
	Excellent	A list of items (retrieved from the database) is displayed. The list of items is refreshed each time the menu item is clicked (displayed items remain the same). The items are displayed in a structured way (with headings for all fields except ids). The list is using Bootstrap styling (at least for the header and to add a hover effect). The list can be filtered by last name (so only lecturers with a certain last name are displayed).	100

[1] *the list of lecturers should be ordered by last name*

Variant C: View all rooms

Individual contribution (80%)	Criterion		Points
The student should have developed part of the application that displays a list of rooms (retrieved from the database).	Unsatisfactory	There is no list of items or there is a static list (not retrieved from the database).	0
	Weak	A list of items (retrieved from the database) is displayed[1].	40
	Satisfactory	A list of items (retrieved from the database) is displayed. The list of items is refreshed each time the menu item is clicked (displayed items remain the same). The items are displayed in a structured way (with headings for all fields except ids).	60
	Good	A list of items (retrieved from the database) is displayed. The list of items is refreshed each time the menu item is clicked (displayed items remain the same). The items are displayed in a structured way (with headings for all fields except ids). The list is using Bootstrap styling (at least for the header and to add a hover effect).	80
	Excellent	A list of items (retrieved from the database) is displayed. The list of items is refreshed each time the menu item is clicked (displayed items remain the same). The items are displayed in a structured way (with headings for all fields except ids). The list is using Bootstrap styling (at least for the header and to add a hover effect). The list can be filtered by size (so only rooms with a certain number of beds are displayed).	100

[1] *the list of rooms should be ordered by room number*

Variant D: View all activities

Individual contribution (80%)		Criterion	Points
The student should have developed part of the application that displays a list of activities (retrieved from the database).	Unsatisfactory	There is no list of items or there is a static list (not retrieved from the database).	0
	Weak	A list of items (retrieved from the database) is displayed[1].	40
	Satisfactory	A list of items (retrieved from the database) is displayed. The list of items is refreshed each time the menu item is clicked (displayed items remain the same). The items are displayed in a structured way (with headings for all fields except ids).	60
	Good	A list of items (retrieved from the database) is displayed. The list of items is refreshed each time the menu item is clicked (displayed items remain the same). The items are displayed in a structured way (with headings for all fields except ids). The list is using Bootstrap styling (at least for the header and to add a hover effect).	80
	Excellent	A list of items (retrieved from the database) is displayed. The list of items is refreshed each time the menu item is clicked (displayed items remain the same). The items are displayed in a structured way (with headings for all fields except ids). The list is using Bootstrap styling (at least for the header and to add a hover effect). The list can be filtered by name (so only activities with a certain name are displayed).	100

[1] *the list of activities should be ordered by date/time*

Group contribution (20%)	20 points for every criterion met

Criterion	Points
The standard C# coding convention is used. *(lowerCamelCase for local variables and method parameters, UpperCamelCase for classes, methods, properties and constants)* Classes/methods/variables/constants have meaningful names. Enumerations are used for limited option values. Constants are used instead of hardcoded values.	20
Methods are no longer than 10 lines of code. Method return values and method parameters are used (instead of using globals).	20
Object Orientation is correctly applied (good OO structures, using interfaces, passing objects instead of separate parameters). Encapsulation is applied correctly (public vs private).	20
SQL code is only used in repositories. SQL injection is prevented. Database primary keys are auto-generated when inserting records.	20
All exceptions are handled (preventing the application to crash). Error/exception messages are clearly displayed to the user.	20