

Assessment model (Rubrics) Project Databases: Assignment 2

Variant A: Students

Individual contribution (80%)		Criterion	Points
The student should have developed part of the application that displays a list of students (from the database).	I	There is no list of students or there is a static list (not from the database).	0
	D	A list of students is displayed.	40
	S	The list is displayed immediately after the menu button has been loaded. The list is included in a ListView. There is a connection with the database.	60
	G	The list is displayed immediately after the menu button has been loaded. The list is included in a ListView. The query does not use SELECT * FROM.	80
	E	The list is displayed immediately after the menu button has been loaded. The list is included in a ListView. The query does not use SELECT * FROM. The list displays the Student number (or ID), First name and Last name in separate columns.	100
Group (20%)			
The group should have submitted a single joint .zip file.	I	The group has not submitted a .zip file or has not adhered to naming conventions. Or: The group has submitted a .zip file, but the file cannot be compiled or run.	0
	D	The group has submitted one .zip file, the application starts up without Exceptions and all menu items (Students, Lecturers, Rooms, Room allocation) can be opened without Exceptions. A division of duties for the various group members has been submitted as a text file within the .zip file.	40
	S	The application starts up without Exceptions and all menu items (Students, Lecturers, Rooms, Room allocation) can be opened without Exceptions. A division of duties for the various group members has been submitted as a text file within the .zip file.	60
	G	The group has submitted one .zip file. All features/functionalities can be used without Exceptions. The group has submitted a file outlining the division of duties. The group has used 'code comments'. Queries have been optimised. A single, consistent style has been used throughout the entire application. Error messages are shown in a separate window.	80
	E	The group has submitted one .zip file. All features/functionalities can be used without Exceptions. The group has submitted a file outlining the division of duties. The group has used 'code comments'. Queries have been optimised. The graphic design of the application has been distinctively improved to benefit user interaction. Error messages are shown in a separate window and are saved in a log. The error messages are useful and go towards improving the user experience.	100

Variant B: Lecturers

Individual contribution (80%)		Criterion	Points
The student should have developed part of the application that displays a list of lecturers (from the database).	I	There is no list of lecturers or there is a static list (not from the database).	0
	D	A list of lecturers is displayed.	40
	S	The list is displayed immediately after the menu button has been loaded. The list is included in a ListView. There is a connection with the database.	60
	G	The list is displayed immediately after the menu button has been loaded. The list is included in a ListView. The query does not use SELECT * FROM.	80
	E	The list is displayed immediately after the menu button has been loaded. The list is included in a ListView. The query does not use SELECT * FROM. The list displays the Lecturer number (or ID), First name, Last name and whether they are a supervisor in separate columns.	100
Group (20%)			
The group should have submitted a single joint .zip file.	I	The group has not submitted a .zip file or has not adhered to naming conventions. Or: The group has submitted a .zip file, but the file cannot be compiled or run.	0
	D	The group has submitted one .zip file, the application starts up without Exceptions and all menu items (Students, Lecturers, Rooms, Room allocation) can be opened without Exceptions. A division of duties for the various group members has been submitted as a text file within the .zip file.	40
	S	The application starts up without Exceptions and all menu items (Students, Lecturers, Rooms, Room allocation) can be opened without Exceptions. A division of duties for the various group members has been submitted as a text file within the .zip file.	60
	G	The group has submitted one .zip file. All features/functionalities can be used without Exceptions. The group has submitted a file outlining the division of duties. The group has used 'code comments'. Queries have been optimised. A single, consistent style has been used throughout the entire application. Error messages are shown in a separate window.	80
	E	The group has submitted one .zip file. All features/functionalities can be used without Exceptions. The group has submitted a file outlining the division of duties. The group has used 'code comments'. Queries have been optimised. The graphic design of the application has been distinctively improved to benefit user interaction. Error messages are shown in a separate window and are saved in a log. The error messages are useful and go towards improving the user experience.	100

Variant C: Rooms

Individual contribution (80%)		Criterion	Points
The student should have developed part of the application that displays a list of rooms (from the database).	I	There is no list of rooms or there is a static list (not from the database).	0
	D	A list of rooms is displayed.	40
	S	The list is displayed immediately after the menu button has been loaded. The list is included in a ListView. There is a connection with the database.	60
	G	The list is displayed immediately after the menu button has been loaded. The list is included in a ListView. The query does not use SELECT * FROM.	80
	E	The list is displayed immediately after the menu button has been loaded. The list is included in a ListView. The query does not use SELECT * FROM. The list displays the Room number (or ID), Room_Type (student/lecturer) and number_of_beds/number_of_people in separate columns.	100
Group (20%)			
The group should have submitted a single joint .zip file.	I	The group has not submitted a .zip file or has not adhered to naming conventions. Or: The group has submitted a .zip file, but the file cannot be compiled or run.	0
	D	The group has submitted one .zip file, the application starts up without Exceptions and all menu items (Students, Lecturers, Rooms, Room allocation) can be opened without Exceptions. A division of duties for the various group members has been submitted as a text file within the .zip file.	40
	S	The application starts up without Exceptions and all menu items (Students, Lecturers, Rooms, Room allocation) can be opened without Exceptions. A division of duties for the various group members has been submitted as a text file within the .zip file. The group has used 'code comments'.	60
	G	The group has submitted one .zip file. All features/functionalities can be used without Exceptions. The group has submitted a file outlining the division of duties. The group has used 'code comments'. Queries have been optimised. A single, consistent style has been used throughout the entire application. Error messages are shown in a separate window.	80
	E	The group has submitted one .zip file. All features/functionalities can be used without Exceptions. The group has submitted a file outlining the division of duties. The group has used 'code comments'. Queries have been optimised. The graphic design of the application has been distinctively improved to benefit user interaction. Error messages are shown in a separate window and are saved in a log. The error messages are useful and go towards improving the user experience.	100

Variant D: Room allocation

Individual contribution (80%)	Criterion	Points
The student should have developed part of the application that displays room allocation (from the database).	I There is no display of room allocation or there is a static list (not from the database).	0
	D Room allocation is displayed.	40
	S The allocation of the rooms is shown immediately after the menu button has been loaded. The allocation has been included in a GridView. There is a connection with the database.	60
	G The allocation of the rooms is shown immediately after the menu button has been loaded. The allocation has been included in a GridView. The query does not use SELECT * FROM.	80
	E The allocation of the rooms is shown immediately after the menu button has been loaded. The allocation has been included in a GridView. The query does not use SELECT * FROM. The view displays the Room number (or ID), the capacity of that room as well as the persons sleeping in that room in separate boxes/blocks. Rooms that are full are distinguished from rooms that are not yet full by way of a colour code (e.g. Red/Full, Green/Not full).	100
Group (20%)		
The group should have submitted a single joint .zip file.	I The group has not submitted a .zip file or has not adhered to naming conventions. Or: The group has submitted a .zip file, but the file cannot be compiled or run.	0
	D The group has submitted one .zip file, the application starts up without Exceptions and all menu items (Students, Lecturers, Rooms, Room allocation) can be opened without Exceptions. A division of duties for the various group members has been submitted as a text file within the .zip file.	40
	S The application starts up without Exceptions and all menu items (Students, Lecturers, Rooms, Room allocation) can be opened without Exceptions. A division of duties for the various group members has been submitted as a text file within the .zip file. The group has used 'code comments'.	60
	G The group has submitted one .zip file. All features/functionalities can be used without Exceptions. The group has submitted a file outlining the division of duties. The group has used 'code comments'. Queries have been optimised. A single, consistent style has been used throughout the entire application. Error messages are shown in a separate window.	80
	E The group has submitted one .zip file. All features/functionalities can be used without Exceptions. The group has submitted a file outlining the division of duties. The group has used 'code comments'. Queries have been optimised. The graphic design of the application has been distinctively improved to benefit user interaction. Error messages are shown in a separate window and are saved in a log. The error messages are useful and go towards improving the user experience.	100