EEEE3084 Coursework 3 Rubric

This rubric is for guidance - any changes will be notified on Moodle Dr Sendy Phang **20/09/2022**

Task description		***Grades (left to right = lowest grade to highest grade)***					
Question 1 (10%)	Task 1a	No diagram and no discussion or Diagram and discussion wrong.	Either diagram or discussion is not provided.	Incomplete diagram with succinct discussion.	Diagram and its discussion is correct. But not following UML standard drawing convention	Diagram and its discussion is correct and professionally made.	
	Task 1b	code is not correct / does not run and there is no commentary how the code should work.	code is not correct / does not run but there is some commentary how the code should work.	The classes run but does not compatible with the unit test.	Classes run, compatible with the unit test.	Classes run, compatible with the unit test. Classes have been developed carefully considering reusability, portability and good programming practice. MVC framework implementation.	
Question 2 (30%)	Task 2a	code is not correct / does not run and there is no commentary how the code should work.	code is not correct / does not run but there is some commentary how the code should work.	The classes run but does not have all the requested capability (incomplete).	Classes run and have all the capability requested.	Classes run and have all the capability requested. Classes have been developed carefully considering reusability, portability and good programming practice. MVC framework implementation.	
	Task 2b	code is not correct / does not run and there is no commentary how the code should work.	code is not correct / does not run but there is some commentary how the code should work.	Classes run and perform the requested action.	Classes run and perform the requested action. Classes have been developed carefully considering reusability, portability and good programming practice. MVC framework implementation.		
	Task 2c	code is not correct / does not run and there is no commentary how the code should work.	code is not correct / does not run but there is some commentary how the code should work.	Classes run and perform the requested action.	Classes run and perform the requested action. Classes have been developed carefully considering reusability, portability and good programming practice. MVC framework implementation.		
	Task 2d	No code. Code is not correct / does not run and there is no commentary how the code should work.	Unit test perform artificial test or just done some test.	Comprehensive test perform.	Test classes run and perform the requested action. Test classes have been developed carefully considering reusability, portability and good programming practice		
	Task 2e	No diagram and no discussion or Diagram and discussion wrong.	Either diagram or discussion is not provided.	Incomplete diagram with succinct discussion.	Diagram and its discussion is correct. But not following UML standard drawing convention	Diagram and its discussion is correct and professionally made.	
Question 3 (10%)	Task 3a	code is not correct / does not run and there is no commentary how the code should work.	code is not correct / does not run but there is some commentary how the code should work.	The classes run but does not have all the requested capability (incomplete).	Classes run and have all the capability requested.	Classes run and have all the capability requested. Classes have been developed carefully considering reusability, portability and good programming practice. MVC framework implementation.	
Question 4 (35%)	Task 4a	No GUI nor controller. No commentary what so ever to provide high level outline what is expected in the controller file	There is GUI but has no event handler. GUI and controller do not interlinked.	GUI and controller works but missing major capability. GUI untidy difficult to navigate/use. Controller messy.	App as a whole work properly and have most the requested features/capabilities (minor incomplete); Reinvent many functions/class which otherwise available as public java package;	App as a whole work properly and have all the requested features/capabilities. Architecture of GUI is creative, use different type of GUI elements; Very user friendly; exploit JAVA built-in OOP features (does not reinvent the wheel); satisfactory performance (time and resource).	App as a whole work properly and have all the requested features/capabilities. Architecture of GUI is creative, use different type of GUI elements; Very user friendly; Controller class has been developed carefully considering reusability, portability and good programming practice; exploit JAVA built-in OOP features (does not reinvent the wheel); Excellent performance (time and resource). GUI considers different monitor resolution (portability). MVC framework implementation.
	Task 4b	No diagram and no discussion or Diagram and discussion wrong.	Either diagram or discussion is not provided.	Incomplete diagram with succinct discussion.	Diagram and its discussion is correct.	Diagram and its discussion is correct and professionally made.	
	Task 4c	No diagram and no discussion or Diagram and discussion wrong.	Either diagram or discussion is not provided.	Incomplete diagram with succinct discussion.	Diagram and its discussion is correct. But not following UML standard drawing convention	Diagram and its discussion is correct and professionally made.	

Good Programming Practice Indicator (15%):	Comments	No comments	Some comments, too brief or vague	Useful comments throughout code
	Variable names	Variables have non-descriptive names	Variable names are words but are ambiguous or vague	Variables sensibly named
	Code structure	no package consideration	some packaging but improper	simple and good use of packaging and properly design considering functionalities. Minimal code kernel in controller file. MVC framework implementation.
	Assesor functions	no get/set functions	Some get/set functions	Has all required get set
	Tidyness and indenting	No indenting, commented non-working codes, improper use of user-defined class/functions	Indenting partially correct, code are tidy, user-defined class/functions correctly used	Code is of a professional standard that could be re-used by others: Indenting neat and consistent, there is no very long line of code, class/functions arguments are alligned neatly (for class/function with many arguments, arguments are listed as separate lines), package/class/function naming follow the naming convention in Java (capitalisation, etc), see (https://www.oracle.com/java/technologies/javase/codeconventions-namingconventions.html)