Format of DSO34BT Proposal

	Task	Description	Mark Allocation
1.	**Name of the Project**	• Name the Project	
2.	**Domain Analysis**	• Explain the general field of business • Show understanding of terminology/glossary being used • Show the general knowledge and understanding of the business environment • Tasks and procedures currently performed • Customers and users • Competing software • Similarities to other domains	6
3.	**Define the Problem**	• Express the Difficulty you want to solve from the domain • Or Opportunity that will result in benefit or improved productivity or sales	4

4.	**Define the Scope**	Narrow the scope by defining a more precise problem
		• Apply knowledge of **Integrated Result Based Management** (IRBM) to define the Inputs, Activities, Outputs, outcomes, and the Impact that the application will have on the community
		• Answer the following questions:
		  • Assess: What is the current situation?
		  • Think: What caused it? Who is involved?
		  • Envision: What are we going to achieve?
		  • Plan: How are we going to do it? With whom? When? With what resources?
			10

5.	**Vision and Objectives**	Write the Vision and Objectives (according to **SMART** principles) of the project	4
6.	**Users of the System**	Indicate the users of the system and their roles	
7.	**Mandatory Functions**	• The system should be able to Add/Register, Delete/Remove and Update data in the database	6
8. 8.	**Functional Requirements** **(Each functionality counts 2 marks)**	• FURPS is an acronym representing a model for classifying software quality attributes (functional and non-functional requirements): • Write the aspect of what the proposed system must do, which contribute in solving the customer's problem and represents a negotiated agreement among stakeholders • What inputs and outputs should the system accept • What computations should the system perform • The timing and synchronization of events	50

9.	**Non-functional requirements**	• FURPS is an acronym representing a model for classifying software quality attributes (functional and non-functional requirements): • **Functionality** - Capability (Size & Generality of Feature Set), Reusability (Compatibility, Interoperability, Portability), Security (Safety & Exploitability) • **Usability (UX)** - Human Factors, Aesthetics, Consistency, Documentation, Responsiveness • **Reliability** - Availability (Failure Frequency (Robustness/Durability/Resilience), Failure Extent & Time-Length (Recoverability/Survivability)), Predictability (Stability), Accuracy (Frequency/Severity of Error) • **Performance** - Speed, Efficiency, Resource Consumption (power, ram, cache, etc.), Throughput, Capacity, Scalability • **Supportability** (Serviceability, Maintainability, Sustainability, Repair Speed) - Testability, Flexibility (Modifiability, Configurability, Adaptability, Extensibility, Modularity), Installability, Localizability	4
10.	**Use Case**	• Describe sequence of actions that a user performs in order to complete a given task as a key activity in requirements using a diagram • This should cover full sequence of steps from beginning to until the end of the task • Describe the user's interaction with the system and **not** computations performed by the system • And **not** actions a user does manually	16
11.	**Tools and Technologies to be used**	a) Indicate the tools you intend to use for the project (e.g. Java, PHP, ASP, etc.)	

Total 100