

DBMS Term Project-Phase#1

Phase I - Requirement Analysis and Specification Report

Due Date: 17.11.2023

Explain your project topic in detail and specify requirements of your project.

Evaluation: Detail description, operations and constraints, entity count, language of report

Details: Explain your project topic in detail.

- Determine the problem domain.
- Determine specifications of your project.
- Determine description of Solution System needed to satisfy Problem Domain.
- Make the list for the system's services (operations, processes) and constraints based on the requirements of the system.

How can you explain the details, try to answer the following questions?

- *What do you want the system to do?*
- *What is the purpose of the system you want?*
- *What is the purpose of the system you want?*
- *Who are the stakeholders? What are their roles in the system?*
- *Create use scenarios, system transactions. You can draw data flow, use case, activity, sequence diagrams.*
- Draw an ER to accurately represent set of business rules requirements.

Entity Count: Entity means "table" or "relation". Your database should contain more than or equal to 5 entities. Your solution system should contain the most appropriate number of entities. Do not try to specify a minimum number of entities.

ER Diagram will be your conceptual design. Specify any assumptions clearly that you are making. You can use any tools to draw the ER diagram.

Then convert your conceptual schema into a logical model that can be implemented in a relational DBMS. Convert the ER diagram to a database design. Document your design in Database Schema format.

Namely, create necessary tables and define relationships between them in a relational DBMS that you've chosen within your project. Use appropriate naming conventions for all of your tables and attributes.

Additional Information About Terms:

- Before you begin any technical work, it's a good idea to apply a set of requirements to engineering tasks. These tasks lead to *an understanding of what the business impact of the software will be, what the customer wants, and how end users will interact with the software*. Requirements engineering is important to understand what the customer wants before you begin to design and build a computer-based system. Requirements engineering encompasses seven distinct tasks: inception, elicitation, elaboration, negotiation, specification, validation, and management.
- **Problem Domain:** The problem domain is the domain in which a system is going to be used. Therefore, it is important to look at requirements from an operational point of view. You establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired, and the effectiveness of preliminary communication and collaboration between the other stakeholders and the software team.
- **Specifications:** Specification means different things to different people. Specification is the activity of the requirements phase where the results of elicitation (ask the customer, the users, and others what the objectives for the system or product are, what is to be accomplished, how the system or product fits into the needs of the business) and analysis are formally captured and documented in an appropriate format for the use and review of all stakeholders.
- **Solution Domain:** The solution domain is the domain in which engineers use their ingenuity to solve problems. The primary characteristic that differentiates the solution domain from the problem domain is that, invariably, requirements engineering in the solution domain starts with a given set of requirements. In the problem domain requirements engineering starts with a vague objective or wish list. The extent to which the input requirements for the solution domain are "well formed" depends on the quality of the people within the customer organization that developed them. In an ideal world, all the requirements would be clearly articulated, individual testable requirements.
- **Constraints** specify bounds identified for their proposed solutions. Typical constraints include schedule, budget, and process constraints. Besides constraints derived from nonfunctional requirements, other constraints include:
 - specific platform constraints (e.g., embedded, mobile, web),
 - constraints from existing and reused architectures,
 - constraints on new technology (e.g., cloud-based solutions), incorporating commercial off-the-shelf (COTS) products,
 - numerous other constraints placed on both product and processes.