

ABSTRACT

As database system development projects become larger and more global in the real-life scenarios, it is important to train computer science students to construct data model and database (CLO2-PLO2: Practical Skills), and to do this effectively in groups (CLO3-PLO5: Interpersonal Skills) through the Database Design and Organization course. The database group project is a significant assessment component but allowing students full freedom to define their own database requirements in the past resulted in superficial and non-realistic outcomes. To move towards authentic (alternative) assessment simulating how database systems are designed in workplaces, we present our DID+F360 (Design, Implement, Demo + Feedback 360) framework. Over 50 different case studies were designed, each modeled after commercial database systems to allow students to demonstrate their skills to execute real-life systems. Students in a team of four are first given a set of database requirements from a case study and assigned a mentor playing the role of the “client” to clarify the requirements in Week 3. A series of hands-on labs provide students with physical context in the form of tools required to complete the case study. In Week 8, each group submits the System Design in which the mentors provide template-based feedback. Finally, instructors assess the System Implementation and System Demo (students must demonstrate the database system the group has implemented) supported by the mentor feedback, peer evaluation and group reflection. The multi-perspective feedback framework better simulates the 360 degrees workplace feedback and allows the instructors to evaluate the effectiveness of teamwork more objectively.

OBJECTIVE

To train database students to build more realistic database models and systems in teams using authentic case studies and multi-perspective feedback.



ADDED VALUES

- Designed 57 unique database case studies modeled after real world requirements.
- Created structured feedback templates aligned to grading rubric to make feedback provision by mentors and instructors more efficient.
- Enabled students to interact with multiple parties (instructor, mentor and team peers) to enhance interpersonal skills.



USEFULNESS

- Avoid the design of fictitious (unauthentic) databases based on student’s imagination.
- Make grading and providing feedback timelier and more scalable for a large class with over 200 students.
- Provide more checks and balances by multiple parties for more effective teamwork among students and objective team-based grading for instructors (e.g., early detection of problematic members, deter sleeping members).



COMMERCIALIZATION POTENTIAL

- Earn royalty per case study requirements and sample solution download.
- Offer the database case study module as a paid micro-credential course.



RECOGNITION

Student Feedback:

“The leadership behaviours and highly productive and initiative traits among my team members led us to have great teamwork and allow us to have effective communication easing the process of completing the project.”

“All of the group members are giving their highest possible commitment, and they all strive to do the best with the abilities they have. Our team spirits are very strong, and we fight and have fun together when we are doing the project.”

Sample case study requirements

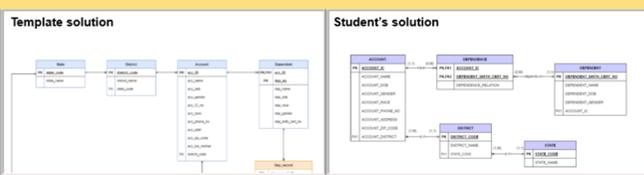
Database Case Study 20:

Vaccination Record Database System

1.0 Database System Requirements

The vaccination record system provides the history of all the vaccines Malaysians received. This system works closely with the Ministry of Health such that all users can access their vaccination history. To access his/her history, the user needs to create an account. Each

Compare student’s design solution to template solution



Grading rubric complemented with feedback template

Criteria	Points	5 Exemplary	3-4 Satisfactory	1-2 Needs Improvement	Feedback Template
Entities	5	Diagram captures all entities necessary for a database that would satisfy the requirements.	Diagram captures at least half the entities necessary for a database that would satisfy the requirements.	Diagram captures less than half or none of the entities necessary for a database that would satisfy the requirements.	<ul style="list-style-type: none"> * Module [has] [does not have] at least 4 entities. * [Number] correct entities are identified from the requirements: [entity names]. * [Number] incorrect entities: [entity names]. * [Number] missing entities: [entity names]. * [Number] unnecessary entities not in the requirements exist: [entity names].
		Diagram captures all attributes necessary for a database that would satisfy the requirements. Attributes are at the atomic level.	Diagram captures at least half the attributes necessary for a database that would satisfy the requirements. Attributes may not be at the atomic level.	Diagram captures less than half or none of the attributes necessary for a database that would satisfy the requirements.	<ul style="list-style-type: none"> * [All] [not all] attributes described in the requirements are listed in the correct entity. * Missing attributes in [entity names]. * Excessive/unnecessary attributes in [entity names].

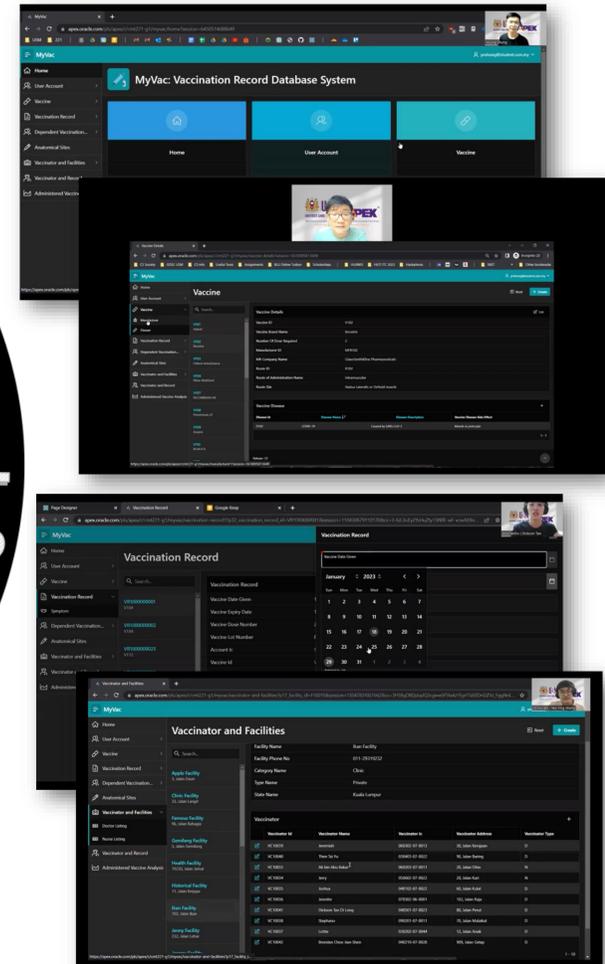
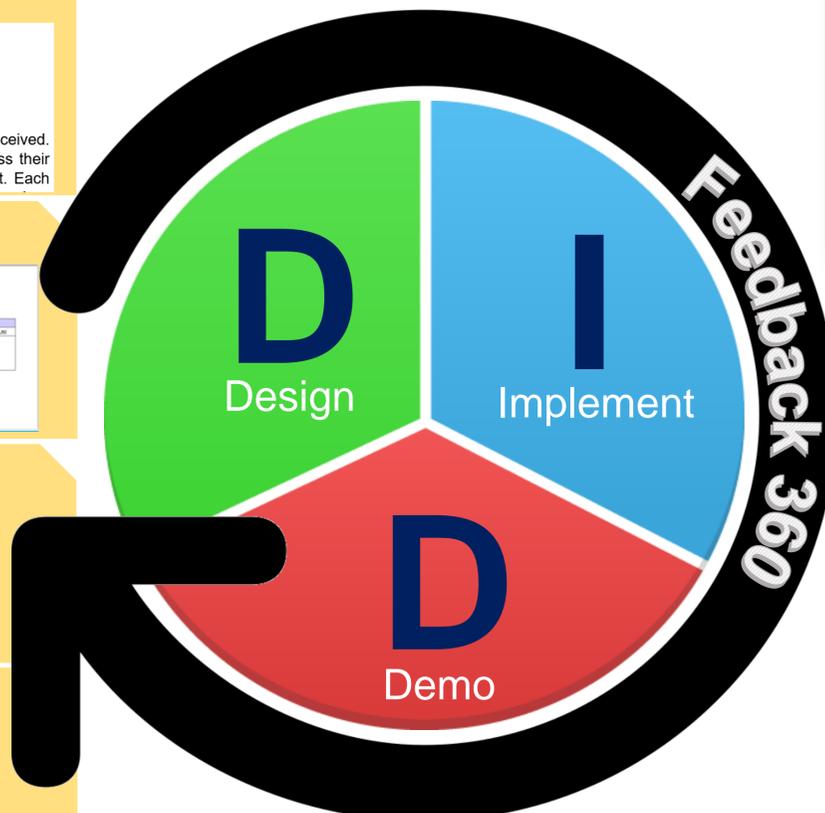
Prepare personalized feedback based on template

[G01] CS20 Feedback

ERD

1. [Module 1]

- Entity: Module has at least 4 entities. 4 correct entities are identified from the requirements: ACCOUNT, DISTRICT, STATE and DEPENDENT. 1 incorrect entity: DEPENDENCE.
- Attribute: Missing attribute in DEPENDENT.
- Keys: OK.
- Relationships: Incorrect relationship between ACCOUNT and DEPENDENT.
- Constraints: OK.
- Complexity: Not all entities, attributes and relationships are correctly identified and modeled. ERD has just the right complexity.



Jasy Liew Suet Yan



Nasuha Lee Abdullah



Nor Athiyah Abdullah



Teh Je Sen



Wong Li Pei