

Zewail City of Science and Technology
University of Science and Technology
Communications & Information
Engineering Program
CIE 206 – Spring
2021



مدينة زويل للعلوم والتكنولوجيا
Zewail City of Science and Technology

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
"نرفع درجات من نشاء وفوق كل ذي علم عليم"

Database Management Systems

Project Requirements

Project Description

Objectives

This project should make the student able to:

- ☐ Go through the phases of creating a relational-database-based system
 - a. Analysis phase
 - b. Design phase
 - c. Implementation phase
- ☐ Use database design tools (e.g. SQL server, Oracle, MySQL.....)
- ☐ Create database application
- ☐ Work in a team and learn how to communicate and organize work with others

Project Phases

<i>Project Phase</i>	<i>Deliverables</i>	<i>%</i>	<i>Due</i>
Analysis Phase	Project Proposal	5%	Week 3
Design Phase	ER Diagram Delivery (initial version)	8%	Week 5
	ER Diagram Delivery (Final)	10%	Week 7
	GUI Wireframe	5%	Week 7
	Database Schema & Scripts	7%	Week 9
Implementation Phase	GUI Submission	10%	Week 11
	Project Alpha release	10%	Week 13
Final Delivery	Project Presentations	10%	Week 15
	Project Discussion	30%	Week 15
Teamwork skills	Two Peer Assessment Forms & TAs feedback	5%	Week 11/15

Note: The submission deadline will be Saturday 11:59 pm for all phases in their respective week

Project Requirements

It is required to create a relational-database-based application. Your application should be based on a *real-world* model. You will go through the following phases:

1- Team Formation Phase

Number of students per team is 2 to 4 students.

Note: *Workload distribution should be vertical. A student is expected to work across all layers in the application.*

2- Analysis Phase

- ☐ Your team should select a *real-world* application that uses database for its operation.
- ☐ Identify the requirements for your database. You may:
 - a . Meet people who use the real-world model of your system,
 - b . Get a list of requirements from a documented system, or
 - c . Identify reasonable and acceptable requirements by yourself.
- ☐ Applications are not supposed to be neither *too complex nor trivial*.
- ☐ Think of non-conventional applications.

Prepare and deliver a *project proposal* report.

3- Design Phase

In this phase, you will identify the structure of the database that will be used in your system as well as Graphical User Interface pages and components.

- ☐ Identify the database entities and the relationships between these entities. This is done by thinking about the real-world model of your system.
- ☐ Draw the "Entity-Relationship" (ER) diagram for your database or the "Extended Entity-Relationship" (EER) if applicable.
- ☐ A good, clear and a well-defined ER diagram will make it easier for you when creating your tables and relationships.

Prepare and deliver the *ER diagram* report.

Your ER report will be evaluated by instructors to correct any design errors

For the GUI Design:

- ☐ Identify different views/pages needed for your project's functionality, don't assume that the system users know anything about the database systems or database queries. Your interface should hide such technical details from the user.
- ☐ Design a wireframe for each view/page outlining the different controls in that view/page. Make sure the wireframe contains all needed controls to cover all use case scenarios/functionalities.

Prepare and deliver the *GUI Wireframe* report.

For the Schema Design:

- ☐ Starting from the corrected ER report
 - a . Follow the algorithm described in the textbook to map the diagram into database relations (schema)

- b. Show primary and foreign keys
- c. Create the database using a database engine (e.g. SQL Server, Oracle, MySQL etc)

Prepare and deliver the *database schema report* and *database creation scripts*.

4- Implementation Phase

In this phase you will convert your design into a working system. This phase implies the development of the *user interface* as well as the *application logic* and the integration between all layers.

- ☐ Identify the tools you are going to use for implementation.
- ☐ Develop friendly GUI using tools like VC#.net, VB.net, Java, etc.

Prepare and deliver the *GUI layer implementation*.

- ☐ Develop application logic.
- ☐ Integrate between application layers.

Prepare and deliver the *Alpha release* as well as *demo presentation*.

5- Final Delivery

- ☐ Fix issue pointed out by instructors during the implementation phase.
- ☐ **Prepare *final presentation* to show your work and deliver final source code of your application**

Analysis Phase Deliverables

Project Proposal Report

Report should contain

- 1- Cover page. (described above)
- 2- Proposed project description: one or two paragraphs to describe the project (the system) you intend to implement.
- 3- List of system users: who can use your system? (2-4 types of users)*
- 4- For each user, write functionalities that your system provides (5-10 functionalities per user type)*
- 5- List of real-world entities that should be present in your system. (10-15 entities)*

Notes:

- ☐ It is required to write three different project ideas.
- ☐ Storing data is NOT functionality; rather it is what users will do with stored data.

* *The above numbers are for guidance and may be somehow tolerated.*

Design Phase Deliverables

1- ER Diagram Report (Initial & Final)

Report should contain

- 1- Cover page. (described below)
- 2- Problem definition.
- 3- List of system users and privileges of each user.
- 4- List of entities and a brief description for each entity. This doesn't mean to list the attributes of the entity. It means to describe what this entity represents in the database.
- 5- List of relationships and a brief description for each relationship.
- 6- ER Diagram.
 - a. The first page should show the entities and the relationships between all the entities (with no attributes on entities. Relationships attributes should be shown here).
 - b. The rest pages should show the attributes of each entity.

Note: *State explicitly any reasonable assumptions or restrictions you have.*

2- GUI Wireframe Report

Report should contain

- 1- Cover page. (described below)
- 2- A wireframe for each view/page in the program including a title and description of the view/page functionality.

Note: *(optional) You could look up Wireflows and use them instead, as they are better at describing user flow in the program.*

3- Database Schema Report & Scripts

Report should contain

- 1- Cover page. (described below)
- 2- The new ER diagram after correcting any errors in ER report.
- 3- Database schema diagram showing
 - a. Database relations (tables) showing primary keys.
 - b. Foreign keys showing the referenced relations. (Can be shown as arrows from referencing relation to referenced relation)
- 4- A .sql file containing the database creation scripts (tables and constraints).

Implementation Phase Deliverables

1- GUI Submission

Submission should contain

- 1- Softcopy of fully developed GUI pages with input data validation needed.
- 2- Report with following:
 - a. Cover page. (described below)
 - b. Workload distribution for GUI development.

2- Project Alpha release

Release should contain

- 1- ID.txt file. (Information about the team and its members)
- 2- Workload distribution.
- 3- Softcopy of the project:
 - a. Logic fully developed.
 - b. Layers fully integrated.
 - c. Database creation scripts with ready sample data - At least 20 tuples in major tables.
- 4- Project demo Presentation.

Final Phase Deliverables

- 1- Softcopy of the project edited and updated with feedback from Alpha Release.
- 2- Project Presentation.
- 3- Report with following:
 - a. Cover page. (described below)
 - b. Workload distribution for the entire project.

Delivery Schedule: To be announced.

Evaluation Criteria

☐ User support (15%)

- Different types of users supported by your application.
- No hardcoded passwords. Users' data should be stored in a DB table.
- Each individual user should have a username and a password.
- Application should enable users to sign up, login, logout and change password.
- Only first admin user can be created manually then he should be able to change his password.
- Admin should be able to create other admins through the application.
- Encrypted password is recommended.

☐ System Functionality (50%)

- How comprehensive is the functionality supported by your application for different types of end users.
- All access to the database must be done through the application. Direct access to database table is not allowed.

☐ Reporting Facility (20%)

- Statistical reports depending on your application
 - Detailed statistical reports for specific parts of the database
 - Managerial level reports to see some overall statistics the whole application

☐ GUI (15%)

- Do not expect users to be database programmers.
- Do not expect users to memorize IDs. Use names instead of IDs.
- Interface should be operation-oriented not table oriented. i.e. categorize your end user interface items with respect to functionalities to be supported rather than tables to be accessed.

☐ Individual's role: Grade for each individual according to his/her role.

Important Note

- Each team member should identify EXACTLY his/her role in the project; as he/she will be evaluated accordingly, not necessarily that all team members got the same mark.

Project Deliverable Documents

Cover Page

All project deliverables should have the following cover page

Zewail City of Science and Technology
University of Science and Technology
Communications & Information Engineering Program
CIE 206 - Spring 2021

<Write here team number>
example1: **team 7**

Database Management Systems

<Project Name>

<Report Title (Deliverable Description)>

Team Number: #

Idea 1:< First idea title>

Idea 2:< Second idea title>

Idea 3:< Third idea title>

Team Members:

<Name>

<ID>

Contact info:

Write one email to be able to contact your team.

<Date>

Appendix A

Super entity and sub entity (Inheritance)

This appendix explains how to handle inheritance (super-sub entity) relationship during project design phase.

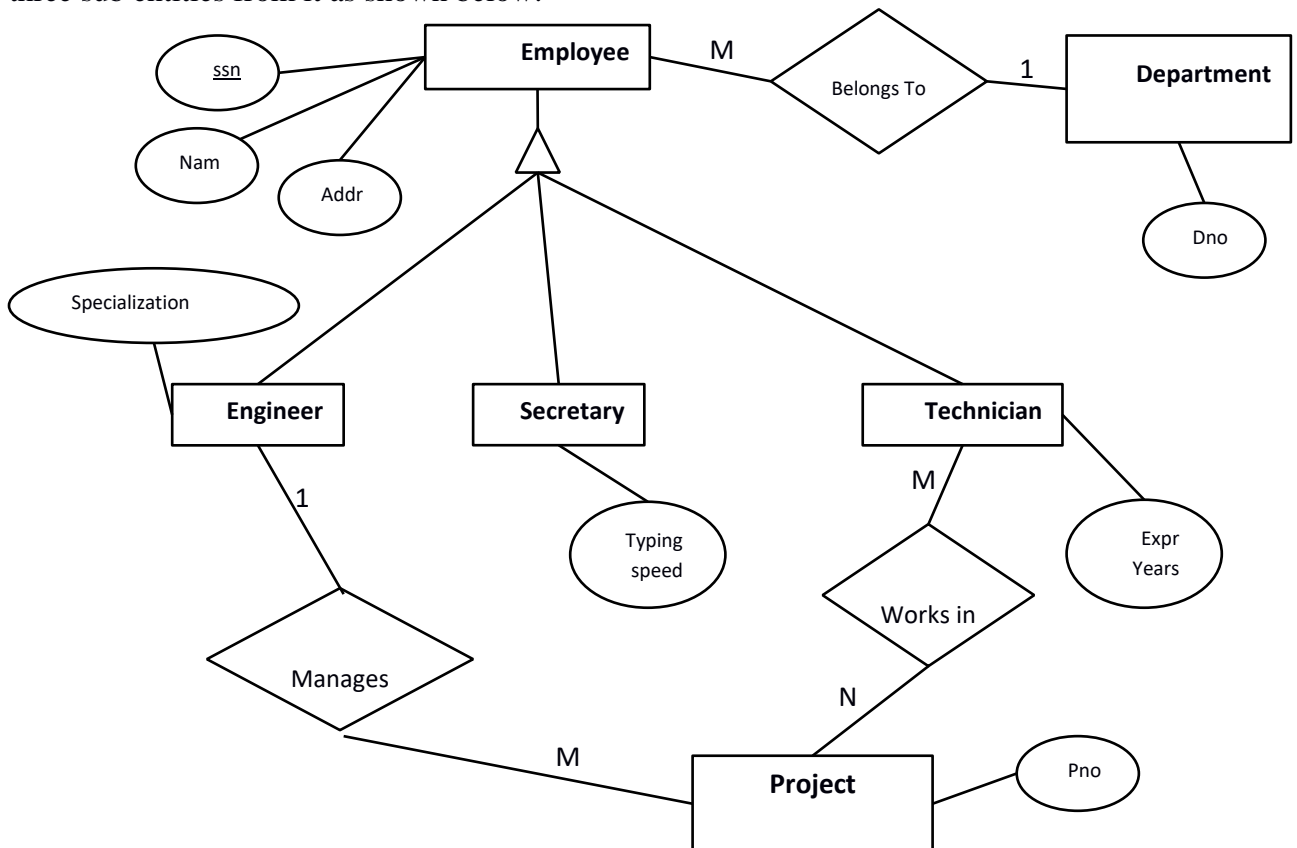
Example:

Assume you are designing a database for a company with the following partial requirements:

- ❑ ALL employees has ssn, name, Addr,.....etc. and belongs to a department
- ❑ An engineer is an employee with a specialization and manages company projects
- ❑ Other types of employees are secretary with a typing speed and technician who works in projects

Here we have three types of employees with some common attributes and relationships but each type also has its own different attributes and/or relationships

To represent such a case in ER diagram, make employee a super entity and derive three sub entities from it as shown below:



Here entity employee is a super entity as it has attributes and relationships common to Engineer, Secretary, and Technician entities. Each sub entity inherits both

the attributes and relationships from its super entity. In addition, it has the same key (as will be shown in the schema below)

The super entity is connected to its sub entities through a triangle as shown in the figure. Please use the same notation in your project ER.

Mapping super entity and sub entity into schema

Super-sub entity relation may be considered as a special type of one-to-one relationship as each sub-entity is related to exactly one super

So the above ER mapping to a schema is:

Department	<u>Dno</u>	...rest of Dept table
-------------------	------------	-----------------------

Project	<u>Pno</u>	...rest of Project table	Mngr_ssn
----------------	------------	--------------------------	-----------------

Employee	<u>ssn</u>	Name	Addr	Dnorest of Emp table
-----------------	------------	------	------	------------	-----------------------

Engineer	<u>ssn</u>	Specializationrest of Eng table
-----------------	------------	----------------	-----------------------

Secretary	<u>ssn</u>	Typing Speedrest of Sec. table
------------------	------------	--------------	------------------------

Technician	<u>ssn</u>	Experience Yearsrest of Tech. table
-------------------	------------	------------------	-------------------------

Works_IN	<u>Tssn</u>	<u>Pno</u>
-----------------	-------------	------------

Foreign keys: (shown in bold)

- Employee (Dno) references Department (Dno)
- Project (Mngr_ssn) references Engineer (ssn)
- Works_IN (Tssn) references Technician (ssn)
- Works_IN (Pno) references Project (Pno)
- Engineer (ssn) references Employee (ssn)
- Secretary (ssn) references Employee (ssn)
- Technician (ssn) references Employee (ssn)

Note:

The last three keys are foreign keys from the super-table and at the same time the primary keys in the sub-tables. This is one possible representation of one-to-one relationships. Other possible representations are applicable too.