

COMP2411 – Database Systems

Deadlines: Forming Group – Oct 4, 2025 | 1st Stage – Nov 1, 2025 | 2nd Stage – Nov 30, 2025

Weighting: 5% | 15%

Project Title: Campus Maintenance and Management System (CMMS)

What to do and what to submit?

You are required to work in groups (as finalized on Oct 3) of **FOUR to FIVE (4 ~ 5)** students.

This group project is divided into **TWO** stages.

1. For the **First Stage**, each group submits a report to Blackboard on or before **1-Nov-2025 (Saturday)**:

- According to the description of the CMMS, provide in the report an **ER Diagram** to clearly define all necessary entity types (which must include “weak” or “normal” entity types) as well as various relationship types. The ER diagram *must include one-to-one, many-to-one, and many-to-many relationship types*. It should roughly match the requirements given below but you have freedom in the details.

In your submission, you should describe *your Assumptions/Requirements, which should match your ER diagram*.

- Include in the report a **Relational Schema** for the CMMS derived from the ER diagram.
- Include in the report a **Project Plan** or a **Project Schedule**.

Your group's first stage submission file should have a name of the form GroupXXStage1.pdf or GroupXXStage1.doc with XX being your group ID. The report should have the following five sections (using the five corresponding phrases as section titles): **Group Members, Assumptions/Requirements, ER Diagram, Relational Schema, Project Plan (Schedule)**.

2. For the **Second Stage**, each group submits the following to Blackboard on or before **30-NOV-2025 (Sunday)**:

- **A User Guide** explaining how to set up and use your CMMS.
- **Source code** file(s).
- **Testing data** file(s).
- A soft copy of the **presentation file** describing what was done by the group (on the ERD, relations, the functionalities, and other relevant aspects).

- A report containing an analysis concerning the strengths, weaknesses, and areas that can be improved, etc. You also include the group members in the report, as was done for stage 1. The report file should have a name of the form GroupXXStage2.pdf or GroupXXStage2.doc with XX being your group ID
- **A FIVE-minute (at most) video demonstration** to introduce and demonstrate your CMMS (with respect to all the supported functionalities) using your testing data.
Note: all group members must be present in the video.
- **Contribution of Work:** A list to describe the workload of each member (in percentages of the total workload). The list should be **signed by all group members** to show your consensus.
- **Peer Evaluation Form:** Each member should complete the Peer Evaluation Form and submit it to Blackboard.

Note: All the above materials, except the Peer Evaluation Form, should be submitted only by one member of your group. All of the submitted files (and folders) should be named as GroupXXStage2ZZZZZ, with XX being your group ID and ZZZZZ being a string indicating what is inside the file.

Late Submission Penalty – 33% per day.

Campus Maintenance and Management System

You are required to create a Database Application of CMMS and to use embedded SQL statements to support the following functional requirements.

1. Campus Maintenance and Management (CMM)
 - a) You will design your database for use by a university or a company with respect to campus maintenance and management. FYI: The project was shaped based on the instructor's observation at PolyU.
 - b) The CMM administrator shall be able to organize and manage the maintenance, renovation, and cleaning activities of the campus. As long as the CMMS is concerned, this administrator works like a DBA, but that person is not a worker or manager described in the database.
 - c) The people of interest to the database include an executive officer, some mid-level managers, and some base-level workers. Supervising relationships exist among these people. Most of the CMM activities are assigned to these people. The company has a limit on the total number of mid-level managers, and a limit on the total number of base-level workers.
 - d) Different kinds of CMM activities are related to different parts of the campus. (The phrase "related to" should not be a relationship type in your design.) Examples include rooms in buildings, levels in buildings, and buildings; squares, and gates. You should include activities

to address daily campus cleaning, activities to address issues due to campus ageing (e.g., window repair), and activities to address weather-related issues (e.g., pool of rainwater on some ground). Some activities may use chemical products.

- e) Mid-level managers can supervise (the maintenance of) buildings. Some activities (e.g. fixing the windows) can be contracted to external companies.

2. Search and Updating

- a) There should be an interface to allow users to insert tuples into various relations, and interfaces allowing users to update/delete tuples of various relations. The interface should also allow a “set-based” insertion option.
- b) There should be an interface to allow users to run SQL queries.
- c) There should be an interface to allow users to find scheduled cleaning activities for a given time period for given buildings, so that they can be informed about whether the scheduled activities will make any parts of the building unusable, and whether harmful chemicals are used in the activities.

3. Report Generation

- d) There should be an interface to generate reports for the administrators to review the system, such as analysing the number of workers performing various types of activities on various parts of the campus. For this interface, the executives do not provide SQL queries to generate such reports.

Feel free to add any other functionalities that you want the CMMS to have.

You need to meet the requirements for the ER diagram (and the implied requirements caused by the ER requirements), and the functional (interface) requirements. You do not need to have too many attributes for the entity types. You do not need to cover every kind of situation for campus maintenance and management.

Moreover, you should follow good practices in naming entity types, relations, and attributes. You should also be consistent with such naming.

Comments: The application appears to be similar to the company database in some ways.