Special Note

Do **not** submit your work until you have been put into a project group on Canvas. Submitting early will mean that your group members will not be able to see the assignment grade or the grading on the rubric.

How can you tell if you have been put into a project group? See the FAQ section in milestone 0 for more information.

General

As you start to work together, it is worth considering what you want your working relationship to look like. "***Who* is on a team matters less than how the team members interact, structure their work, and view their contributions**". In particular, please do think about how you can work together to provide each other to provide a healthy and productive environment. If you feel that problems have come up within your group, please talk to your project TA or instructor as soon as you can.

Project Topic

You are allowed to choose your own topic as long as it is not something from our lectures or tutorials, another course, the Internet, previous students' work, your co-op term, etc. The work you submit must be something original created for this class during the current semester.

Blacklist – Some project topics that you may **not** use are:

- Employees and departments (projects, managers, employees) –textbook example
- A bookstore or library (used in previous semesters)
- Car rental service (used in some previous semesters)
- MP3 storage (used in some previous semesters)
- A school setting (students, instructors, departments, courses, etc.) – We're going to use this as a case study in various parts of the course.
- Banking (accounts, customers) – another common example
- Airlines (flights, planes, passengers, destinations) – another common example used by many textbooks
- Retail sales (customers, products, orders, line orders) – another common example. Instead, if you want to do something in retail, consider other aspects of retail, like maybe inventory tracking, shipping, etc., and don't focus on ordering.
- Hospital or medical clinic – again, common
- Any other examples presented in the course
- Any project that you have previously done (i.e., you must do a new project and cannot reuse an existing project idea)

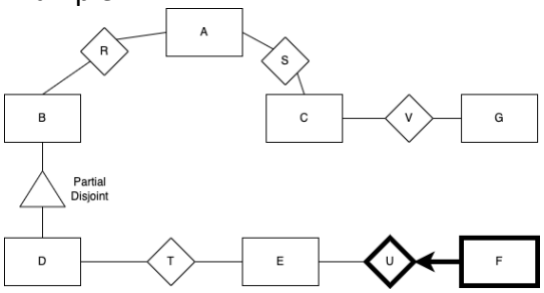
Before you decide on a topic, you will want to carefully consider whether your project would be considered as a variation of something on the blacklist. For example, AirBnB is a topic not on the blacklist but the general structure of this is very similar to a car rental (which is on the blacklist). We **don't** want to see a re-skinned version of a blacklist project. If you feel that your project may be considered similar, carefully explain how your project idea will be significantly different from a banned topic.

ER Diagram Requirements

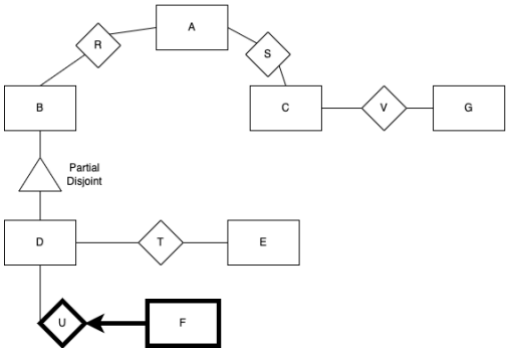
We expect each project to have:

- At least 7 different entities and 7 relationships, not counting weak entities and ISAs
    - All entities must have their keys clearly identified.
    - What does "not counting weak entities and ISAs" mean? It means we only count owner/parent entity sets. For example, consider the following two examples (note that the examples are meant to illustrate how we count entity sets and does not include all the requirements we expect for the milestone 1 ER diagram).

Example 1:



Example 2:



Both example 1 and example 2 would be considered as five entity sets (specifically, we would count entity sets A, B, C, E, and G).

- At least one meaningful ISA relationship
    - An ISA relationship only counts as one entity. E.g., a parent entity with six children counts as one entity, not seven.
    - For each relationship, identify the cardinality constraint and other constraints, such as participation constraints. We encourage you to provide a mix of constraints (some 1-to-many, many-to-many, 1-to-1).
- All entities must have their keys clearly identified using the notation discussed in class
- At least one meaningful (non-trivial) weak entity; but, if this is not suitable for your particular application, then replace it with one more meaningful ISA relationship.

Project Platforms

You may use any legal platform that you like, as long as:

- The final application uses a relational DBMS to manage the data.
- You don't use an automated system or tool (e.g., Github Copilot) to generate the code and SQL (we want you to write these on your own).
- You do not use a GUI generating tool (e.g., Netbeans).
- All other requirements are met. Ask, if in doubt.
- **You must use version control for your code. CPSC 304 will provision repositories for each team to use and you must periodically commit your code to this repository.**

The department has an Oracle server available for your use (see here for more details). You will have a chance to use Oracle with Java and PHP in tutorial. The course staff will be able to support Java and PHP projects that use the department RDMBS. We will not be able to support any other combination of technology.

If you choose to use something outside of the department provided infrastructure (e.g., a MySQL server on your own machine), you may be asked to submit additional information. Furthermore, problems with non-supported platforms will NOT be accepted as an excuse for late project submission.

Deliverables

All of the following items must be put together into a **single** PDF file.

1. A completed cover page (template on Canvas)
2. A brief project description answering these questions:
    a. What is the domain of the application? Describe it.
    The domain of an application refers to the area of knowledge your application resides in. For example, if I am making an application for a hospital, the domain would be something like healthcare/patient management/logistics (it would depend on what the application is trying to do).
    b. What aspects of the domain are modeled by the database? In answering this question, you will want to talk about what your project is trying to address and how it fits within the domain. It is likely that in the process of answering these questions you will bring up examples of a real-life situation that the application could be applied to.
3. Database specifications: (3-5 sentences)
    a. What functionality will the database provide? I.e., what kinds of things will people using the database be able to do.
4. Description of the application platform: (2-3 sentences)

   a. What database will your project use (department provided Oracle, MySQL, etc.)? See the "Project Platforms" section of this document for more information.
   b. What is your expected application technology stack (i.e., what programming languages and libraries do you want to use)? See the "Project Platforms" section of this document for more information.
      i. You can change/adjust your tech stack later as you learn more about how to get started for the project via latter tutorials.
5. An ER diagram for the database that your application will use. It is OK to hand-draw it but if it is illegible or messy or confusing, marks will be taken off. You can use software to draw your diagram (e.g., draw.io, GoogleDraw, Microsoft Visio, Powerpoint, Gliffy, etc.) The result should be a **legible** PDF or PNG document. Note that your ER diagram must use the conventions from the textbook and the lectures. For example, **do not** use crow's feet notation or notation from other textbooks).
   a. Please limit your diagram to a letter size page (8.5 x 11 inches). If you require additional space, talk to your project mentor **beforehand** as this might mean that your project is a bit more complicated than what we expect.
6. Your E/R diagram should adhere to the expectations listed above.
7. Other comments, as appropriate, to explain your project.

Check the Milestone 1 assignment on Canvas for the rubric. Refer to the syllabus for information on late submission/penalty rules.

FAQ

1. **What tech stack should we use?**
   There is still time from this assignment to when you actually start implementing the project. If you change your minds by the time you start implementation, you are not tied to your decision in Milestone 1.

2. **I don't know PHP. Will this hinder me?**
   No. We do not expect CPSC 304 students to come into the course knowing PHP. Picking up a new language is a skill that many Computer Scientists have to acquire. This project is a safe environment where you can work on these skills; if you are having trouble, please go to the PHP office hours and the TAs will be happy to give you some pointers on how to get started.

3. **Do we start coding from an empty repo? Where do we start coding once we are ready?** You can reuse the code provided in Tutorial 6 and 7; in fact, this is recommended as opposed to starting from scratch. You will need to use the provisioned repository given to you by the CPSC 304 course staff.

   WARNING: Do not start coding until you get your feedback for milestone 2 as feedback from your TA may cause substantial changes.

4. **I want to ask my project mentor a question but I don't know who they are**.

   Project mentors will be assigned shortly after milestone 0 has concluded. However, as there are a lot of logistical issues behind the scenes, it will take a few days for those things to be settled. Once the project mentor assignment has been finalized, it will be released via Canvas. Keep an eye on Piazza for more information.

5. **Once we are finished our milestone, does everyone in the group need to submit a copy or only one person?**
   Once you have been put into a group on Canvas (see the FAQ section in milestone 0), you can submit your work. Only one person needs to submit the deliverables; **always double check your submission to ensure that you have submitted the right documents and that the submission has gone through**. Reasons such as "I didn't realize I submitted the wrong file" are not considered as a valid reason for regrades or for waiving a late penalty.

   In some semesters, there are milestones that need to be submitted individually so always double check. Milestones that need to be done individually will explicitly say so in the description document.