# Assignment #1: Data model [10%]

This assignment relates to the following Course Learning Requirements:

CLR 1: Identify, explain, and use various technologies used in the Enterprise environment.

CLR 3: Implement Web Server integration with enterprise applications.

CLR 5: Utilize as well as defend against common security vulnerabilities found in enterprise applications and the multi-server networked environment.

CLR 6: Implement and Integrate various Java based technologies used in the enterprise environment.

Objective of this Assignment:

# Execute the planning of enterprise applications by defining a schema of a data model in terms of MVC and choosing a communication model.

# Pre-Assignment Instructions:

1. To prepare you for this assignment, read the modules 3, 4, and 5 content and follow the embedded learning activities.
2. You require a drawing software that allows you to draw diagrams and charts.
   1. The following tool is a suggestion:
      1. Standalone app of <https://app.diagrams.net/> was used, which can be downloaded here <https://github.com/jgraph/drawio-desktop/releases/tag/v13.6.2>
      2. You are welcome to use any other such as Microsoft Visio (Windows only), pen and paper (literally).
3. You also must have the following database installed on your computer.
   1. For simplicity:
      1. You can use MySQL v.8+ (<https://dev.mysql.com/downloads/>), an example of the data model is provided in module 5.
      2. You may use any database you wish, moreover, if you decide to switch your solution to NoSQL it would be considered a bonus – ***NOTE: it should be correct and workable, otherwise no bonuses will be added).***

**Assignment description:**

In this assignment, you will start by developing a backend for a Twitter-like application.

The functionality of the application should have the following:

1. At least 2 roles
   1. Producer; and
   2. Subscriber
2. The Producers role is the same as the Subscriber, but has some extra capabilities:
   1. The producer can produce messages, that the Subscribers get
3. User may have both roles at the same time;
4. Users having the Subscriber role, which means they can subscribe to as many Producers as they want to;
5. All messages stored in the database, can be easily searched based on the following criteria:
   1. User (Producer) ID – means who wrote it;
   2. Message content

**Assignment Tasks:**

What you should do:

1. Acquiring MVN design pattern, you need to build a data model for the application described in the ***Assignment Description*** section together with API Contracts
2. As a result, you should have the following:
   * ERD diagram (if you use RDBMS) as a picture of JPEG/PNG format; (1 point)
   * Script to create the data model in the DB (SQL for RDBMS, CQL for Cassandra, etc.), which should be runnable, and creates an expected data structure; (2 points)
   * Script to populate the database with few records (2-3 users, 5-7 messages); (2 points)
   * A of the following queries: (5 points)
     1. Get list of users; (1/2 point)
     2. Get list of content producers; (1/2 point)
     3. Get list of content subscribers (full); (1 point)
     4. Get list of content subscribers, subscribed on specific producer; (1 point)
     5. Get all messages; (1/2 point)
     6. Get all messages created by specific producer; (1/2 point)
     7. Get all messages for given subscriber (it may include messages from multiple producers). (1 point)

**Assignment Submission**

You need to submit an archive with name pattern {Course #}\_{Section #}\_{Last name}\_{First name}.zip containing the following:

* + ERD diagram (if you use RDBMS) as a picture of JPEG/PNG format;
  + Script to create the data model in the DB (SQL for RDBMS, CQL for Cassandra, etc.);
  + Script to populate the database with few records (2-3 users, 5-7 messages);
  + List of queries in a regular text file;
  + Screenshots demonstrating the result of each query execution.