Project requirements

The following requirements must be tackled for the final project. Some requirements are mandatory, without them your project will not make sense. Requirements marked with * after the points are optional, you can choose as many optional requirements as you want.

Starting points	0
Maximum points	13

Data Model

1. **(2p)** Create a diagram for your project which models the classes, interfaces and relationships between them. You **must** model at least 3 classes and 2 interfaces. You **must use** at least two collections (list, set, map, etc.).
2. **(1p)** Create at least one test case (unit test) per class in your model.
3. **(1p)*** Use configuration files and/or program arguments to start the program in different ways.
4. **(1p)*** Identify and handle at least 3 language Exceptions (FileNotFound, IOException, etc.) that apply to your project
5. **(1p)*** Create at least 3 custom Exception useful for your project.
6. **(1p)*** Use threads to accelerate processing and/or to handle multiple clients.

Database connection

1. **(0p)*** Install a database management software and create the entities related to the model of your project (classes, relations). This part is related to the DB course, therefore no points are awarded. Yet, this task is **mandatory** if you want to tackle other database connection tasks below.
2. **(1p)*** Create a connection with the database, query data and show it on screen (or UI).

3. **(1p)*** Insert or update data into the DB tables.

4. **(1p)*** Manage users or roles for different access levels to the DB.

User Interface

1. **(1p)*** Validate input data before using it in your program (especially when interacting with the DB). All inputs must be validated. This does not mean you need to create a GUI, command-line-interfaces works also. By user interface we mean that a user can input data into the application.

2. **(1p per Scene)*** Create Scenes for your project. Each Scene must provide a different behaviour for your project. Example: one scene for an admin that manages users, one scene for a user managing the stock, one scene for a client trying to buy from the stock. The stock can be any entities present in a database.
   Note: max 2 points