

Software Development III - Object Oriented Architectures & Secure Development

© Howest University of Applied Sciences

N-tier applications using mock data and MySQL

General assignment

- Write a layered FXML application to manage invoices.
- Each invoice has the following properties:
 - `id : int`
 - `description : String`
 - `unitPrice : double`
 - `quantity : int`
 - `vat : int`
 - `totalPrice : double` , equals `unitPrice × quantity + vat %`

About the UI

- Using SceneBuilder (or FXML...) create a UI allowing users to:
 - get an overview of all invoices
 - in a list
 - each list item displays the invoice's details (don't worry too much about the visual aspect)
 - add a new invoice
 - provide input fields for everything except `id` , which should be assigned automatically and `totalPrice` , which is to be auto-calculated.

Creating a solution with in-memory repositories

- First, make sure the invoice management system works with an in-memory repository.
- Seed the in-memory repository with some data, for example:

ID	Description	UnitPrice	VAT %	Quantity
3	Laptop	699.99	21	3
1	Cheese	3.99	6	18
7	Smartphone	499.99	21	5
99	OLED TV	1299.99	21	4
12	Apples	2.99	6	76
8	Bananas	4.99	6	23
10	Cookies	5.99	6	54
5	Milk	1.99	6	9

- Implement the necessary logic (in the correct layer!) to verify that invoices cannot be added with a negative price or a VAT percentage different from 6 or 21. If that happens, an `InvoiceException` is to be thrown.
- Try the UI: are you able to view the existing invoices and add new invoices?
- Make sure it is impossible to add invoices with a negative price or invalid VAT percentage. Catch the `InvoiceException` and handle it properly: a nice customised error message should be displayed.
- What happens to any newly added invoices when the program is shut down and re-launched?

Creating the MySQL database

- Create a new database schema called `invoices-db`, with one table `invoices`.
- Make sure this table contains the necessary columns to store the data in a correct fashion.
- The `id` field (primary key) should be auto increment.
- Create a MySQL user (and associated password) that can only access the `invoices-db` schema and that can only perform `SELECT` or `INSERT` operations on the `invoices` table. Why is this a good practice?

Creating and using your MySQL repository

- Create the MySQL connection class and repository.

- Make sure `SQLException` s do not bubble up to the user interface. Make correct usage of the `InvoiceException` class and use logging where appropriate.

General tips

- Apply the various techniques studied during the classes.
- There isn't one "single solution". Make sure you can motivate your choices.
- It is your software, take ownership.