Date:_____ Student: _____

L9 - Package and class design principles – individual/pair work

Design a software solution for an Emag-clone (similar to the actual Emag solution, but simplified for this exercise).

The clone should feature both a web app (server side rendering, synchronous calls) and a mobile app (IOS, Android).

The clone should allow users to register, buy and pay for products. For card validation & charging, the clone uses an external payment service from a bank.

The products are organized in categories and sub-categories (no sub-sub-category) and they cannot be in several categories. Each product has a description, a list of specifications (different for each product/category/sub-category)

In order to perform a sale/checkout, an order is created (shopping cart) and as another item is added, a new order item is added to the order. The order is saved in the system for up to 2 weeks from the last action.

If the buyer performs a checkout, the system will attempt to charge the creditcard and if succes, the order is finalized & sent for delivery. A confirmation email is sent via a transactional email service provider.

All information must be persisted in a structured format.

Missing (important) requirements are to be assumed to be as close to the actual Emag solution (please do not over-engineer or over-complicate)

Instructions:

- identify the main stakeholders
- identify the main actors
- create the the domain model diagram
- create the ER diagram
- research online the OOD Class design principles (SOLID & others) and list them with a 1-sentence description
- create the class diagram (using the principles above) [https://www.tutorialspoint.com/uml/uml_class_diagram.htm , https://edn.embarcadero.com/article/31863#classdiagrams]
- create the deployment diagram [https://www.tutorialspoint.com/uml/uml_deployment_diagram.htm , https://edn.embarcadero.com/article/31863#component-and-deployment-diagrans]
- list & highlight the patterns used (architectural, GOF, ...) and explain the reasoning (minimum 3 patterns needed, 2-lines explanation max for each)
- list all the assumptions you have made (where the requirements were incomplete/confusing/contradictory)