Supermarket Billing Software

Most of the supermarkets use software for preparing the final bill before the goods are packed. This software itself can very easily be designed by someone starting in Java (Note that most of the actual software used for billing in supermarkets use Java itself). This billing software sums the amount for individual items and then adds them up to get the final sum that the customer has to pay. The number of individual items, the price of each item should be editable for the user. Also, there should be an option to remove the item from the list altogether.

We should choose the only design pattern:

Factory Method | Abstract Factory | **Prototype** | Singleton

Requirements for UML:

- For the UML class diagram: the fields / methods are implemented logically according to

the feature (you don't have to define a lot of fields/methods, define only those that are necessarily obvious). And when you are defining primitive types, the types should be consistent with: https://www.uml-diagrams.org/data-type.html

- For the UML class diagram: the relationships between classes should also be defined (if

there is an association then define it). Look at here: https://www.uml-diagrams.org/class.html. When there is a relationship between classes, you should also indicate it in your class fields/methods (for example, if class A has the instance of class B, then indicate class B as a field in the class A).

- When you implement the UML diagram in code, follow OOP principles. And your code

should be consistent with your UML diagram, say all the methods/fields in your UML diagram should also be implemented in your code. Also, show how your classes can be used. (For example, if you created classes for the Factory design pattern, also write a code snippet that implements, creates objects etc).

UML Diagram of Supermarket Billing Software