**Assignment 5: Software Design**

In this assignment, you are requested to create a software design for *EZShop*, an easy shopping app for a supermarket chain, whose requirements are listed below. To do so, you should follow the same approach that we present in the P3L2 lesson, that is, analyze the requirements to identify and refine (1) classes, (2) attributes, (3) operations, and (4) relationships in your design. Just to be completely clear, **your task is to design the system, not to implement it**.

## Requirements

1. You can assume that each existing customer already has a card with a [QR code](http://en.wikipedia.org/wiki/QR_code) that can be scanned by the *EZShop* app and encodes the following customer information:
   1. First name
   2. Last name
   3. Zip code
   4. Email address
2. Similarly, all products in the supermarket are suitably labeled with a QR code that can be scanned by the app and encodes the item’s ID and price.
3. Alcoholic beverages are treated differently from other products, as a special tax applies to them. To indicate that, their ID always starts with an “AB” prefix.
4. Before starting to grocery shop, a customer would launch the *EZShop* app. When started, the app would:
   1. Launch the QR code scanner
   2. Read the customer card’s QR code
   3. Suitably initialize the customer’s information based on the information read
   4. Initialize an empty cart with an empty list of items and coupons
   5. Go into grocery shopping mode
5. While in grocery shopping mode, a customer can then use the app to add an item to his/her list of currently purchased items before putting them in the physical cart. To do so, the customer must press the **“+”** button on the app’s screen. At that point, the app:
   1. Launches the QR code scanner
   2. Reads the QR code of the item, which provides the item’s ID and price
   3. Adds the item’s ID and corresponding price to the list of current items in the customer’s cart
6. While in grocery shopping mode, a customer can also scan coupons for a given product. In that case, the app:
   1. Launches the QR code scanner
   2. Reads the QR code of the coupon, which provides the product’s ID and the corresponding discount (assume a simple coupon model, in which a coupon contains a discount that simply applies to each item with that ID)
   3. Adds the product’s ID and corresponding discount to the list of current coupons in the customer’s cart
7. If a customer makes a mistake or changes his/her mind about an item, he/she can remove the item by pressing the **“-”** button on the app’s screen. At that point, the app:
   1. Launches the QR code scanner
   2. Reads the QR code of the item, which provides the item’s ID and price
   3. Removes an entry corresponding to that ID from the list of current items in the customer’s cart
8. When done, a customer can press the **“Pay”** button on the app’s screen, which generates a QR code that encodes (1) the customer’s information, (2) the items and coupons lists, and (3) the total value of the cart, computed taking into account the items in the items list (including whether they are alcoholic beverages or not) and the coupons in the coupons list. The customers would then show the QR code to a cashier, who would scan it, possibly double check the items list with the items physically in the cart (based on a random selection meant to keep customers honest), collect a payment method from the customer, and conclude the transaction when the payment is successfully completed, which would also send an email to the customer with their receipt.

Your design should be expressed using a UML class diagram, and the level of detail of the design should be analogous to the level of detail we used in the P3L2 lesson. If needed, you can also provide an optional additional document, in which you provide additional information about your design, such as assumptions or rationale for some design decisions.

In case you are not familiar with any UML tool, one option is to use [ArgoUML](http://argouml-users.net/index.php?title=Main_Page), which is a free UML editor. (If you are using the VM we provided, ArgoUML is already installed).

To submit your assignment, you should do the following:

* Create a directory called Assignment5 in the usual **personal GitHub repository we assigned to you**.
* Save your UML class diagram in the Assignment5 directory as a PDF file named design.pdf.
* If you created the additional information file, save it in the same directory in markdown format and name it design-extrainfo.md.
* Commit and push your file(s) to your remote repository.
* Submit the commit ID for the files on T-Square.