

Goodies To Go Online Café (G2G)

Mini-project

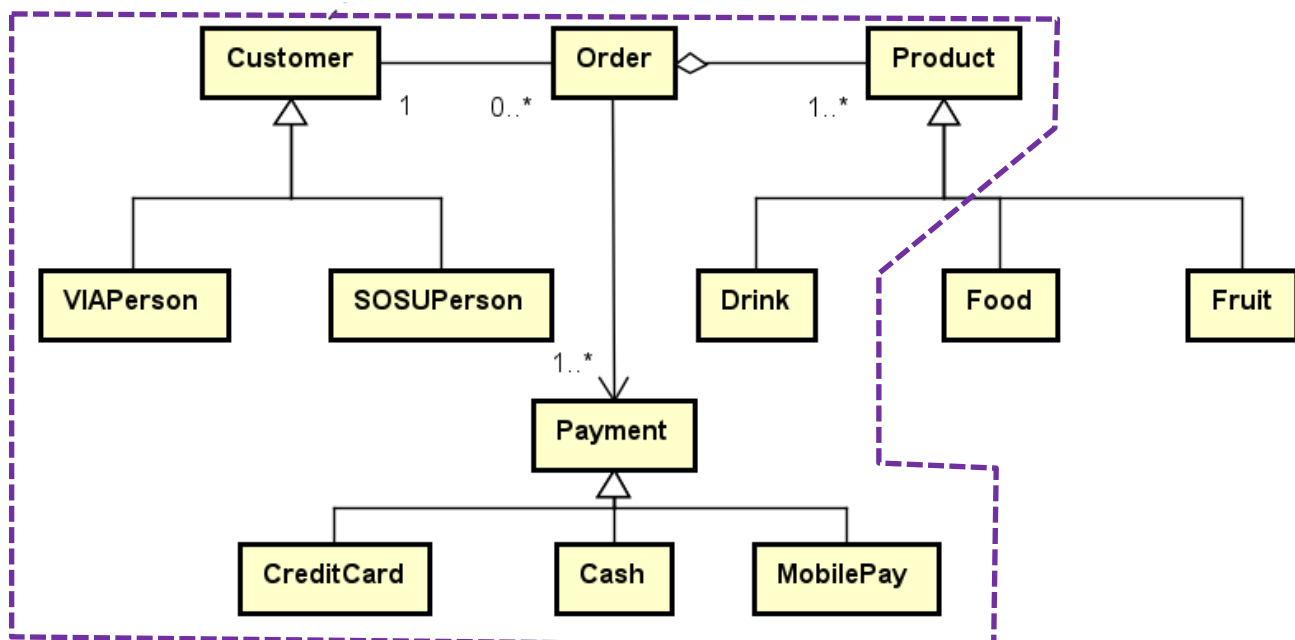
N/B: To be presented on the 24th of March

Case Study:

The objective of this case study is to design and implement a simple ordering system combining some programming concepts and F# programming. This is an example case study where we consider VIA Canteen that sells some products such as food and drinks. For now, orders are taken in-house by order entry personnel. However, some of its customers come from outside (like the SOSU or other buyers using the banegarden). On a closer look, VIA canteen realizes that it cannot be effectively run with its current setup and therefore needs a new online order management system (with a new name Goodies To Go - G2G).

At the moment, G2G offers drinks, food and fruits. Further, it is considering adding to its product line in-demand take-away goodies and other products that are not just food, fruit and drinks. G2G is very ambitious; it hopes to open a second order-processing location on the Arhus (or other VIA Campuses) within the next one year.

Our challenge/goal is to design and implement a system that not only meets G2G's immediate needs but also is flexible enough to support other types of products in the future. The solution will allow customers (VIAPerson, SOSUPerson) to order goodies through their computer (or mobile phone). Below is a simple domain model showing some of the objects and associations.



In order to develop this application, a simple domain-specific language for describing the objects, products and all the different categories of drinks (or food, fruit) as well as computing the prices need to be developed.

In this “first sprint” of the mini-project, you will define data types to describe domain in addition to one product (Drink). Define the different types of drinks in the café and implement the price computing function. You need to decide for yourself the prices for the different drinks. For instance, the **drinks** can be different types of **coffee, tea, juice, soda, milk**, together with their corresponding **sizes** (Small, Medium, and Large) and should have different **prices**.

The types should have at least:

- 3 different types of coffee drink
- 3 different types of tea drink
- 3 different types of juice drink

You will be computing the different prices for the above drinks depending on the type and size of the drink.

Sprint 1:

You are allowed to develop your own strategy for describing the domain objects and the different drinks together with their sizes as well as functions that compute the prices for a given type of drink.

As a minimum, you should define a simple data type for describing the different drinks and implement a function to compute the price. (You may include food and fruit if you want but it is not a requirement in this first sprint)

Hints: You will need to define the data type for drinks (**union**) and the three chosen category, for instance, coffee or tea or juice together with the size (**record**). You will also need to define the price computation **function** that uses **pattern matching**.