Analysis

Analysis is closely tied with Inception because the outcome of an Analysis are requirements, use cases and use case descriptions. Meanwhile, the outcome of inception is a product backlog, a use case diagram, several use case descriptions and documents that define how SCRUM will be used as a methodology (scrum roles, scrum schedule…). SO: a part of Inception will fall into the Project Description (requirements, technical stuff) and another part (Scrum stuff) will fall into the Process Description.

PROBLEM DOMAIN DESCRIPTION

VIA's Campus Café was freshly inaugurated, providing the students with a place to have a cup of coffee, relax, study and play boardgames inside of the guest canteen. However, the fact that it is recent means that there has not been time for their management methodologies to be honed. So far, their strategy for managing orders relies solely on human interaction: when a customer fancies a coffee, they walk up to the barista and place an order. The barista then either memorizes the order or writes it down on paper for another colleague.

This procedure has its imperfections, though: both these steps enable human error and take time, which could make the queue longer and upset those at the end of it. Another thing that could increase queue time would be if one employee was working alone. First of all, they would have to play the role of both barista and cashier. Secondly, memorizing all orders is too demanding, so they would have to write them down. Subsequently, with the external pressure, they are also more prone to making mistakes in the orders, which leads to a "snowball" of issues for one person to deal with alone.

The problem also arises on the other side of the counter: for example, customers who are introverted might be reluctant to approach a barista and make their order. Alternatively, they might take a long time, which increases waiting time for other people. Another thing that can cause complications is the indecisiveness of humankind: if a customer wants to change their order, they will have to phrase this request to an employee, which can confuse them and take up more of their valuable time.

Additionally, the inconvenience of the present situation does not stop there: currently, the existing menu is on paper, and it does not feature all of the possible extras at the Campus Café. This makes it so that customers do not always know what syrups are available to add to their coffee, making them miss out on a potentially more tasteful experience. Furthermore, if a change to the menu is to be made, a whole new set of physical menus will have to be printed and laminated, which is harmful for the environment, unsustainable and, overall, a waste of resources.

In summary, the Campus Café at VIA is struggling to keep up with orders in days of higher demand, in which it would be overwhelmed with customers, causing longer waiting times, general dissatisfaction and unreasonable pressure on employees.

PRODUCT BACKLOG - KAMIL

After outlining the problem domain, the next step is proposing the requirements in order to analyze the specific issues in detail. By doing so, a better understanding of the problem can be obtained. This is also the way of communicating with the product owner as the requirements should reflect their wishes and needs. Therefore, conducting meetings with the product owner is essential to create reliable and goal-oriented requirements.

The requirements were formulated in the form of a Product Backlog, which is a SCRUM artifact. This template features an user story ID, the priority level, an estimate of hours to complete it and the user story as a small sentence.

The main role in the system will be the customer as this system is about ordering items. As a consequence, the customer needs to be able to place and modify their order. Other two crucial actors are barista and cashier where a barista is realizing orders and cashier is modifying and accepting orders. An indispensable role is admin who is granting permissions for cashiers and baristas and modifying the items available in the system. Less essential, but still needed, is a display actor who is displaying the pendings orders and the one ready for pick-up.

The user stories were formulated according to the SMART principle: they are Specific, Measurable, Achievable, Relevant and Time-Bound. In practice, each letter of SMART means:

- **Specific:** They are formulated in a clear way that cannot be misunderstood and that is not vague.
- **Measurable:** The user story is formulated in a way in which it is easy to later determine whether it was accomplished or not.
- **Achievable:** The user story is realistic in a sense that, after considering all factors involved in the process, it is possible to accomplish it.
- **Relevant:** Each user story brings the project as a whole closer to completion, as opposed to being dispensable or out of place.
- **Time-bound:** Fair estimates should be elaborated for each user story, enabling accurate planning of workload.

PRODUCT BACKLOG

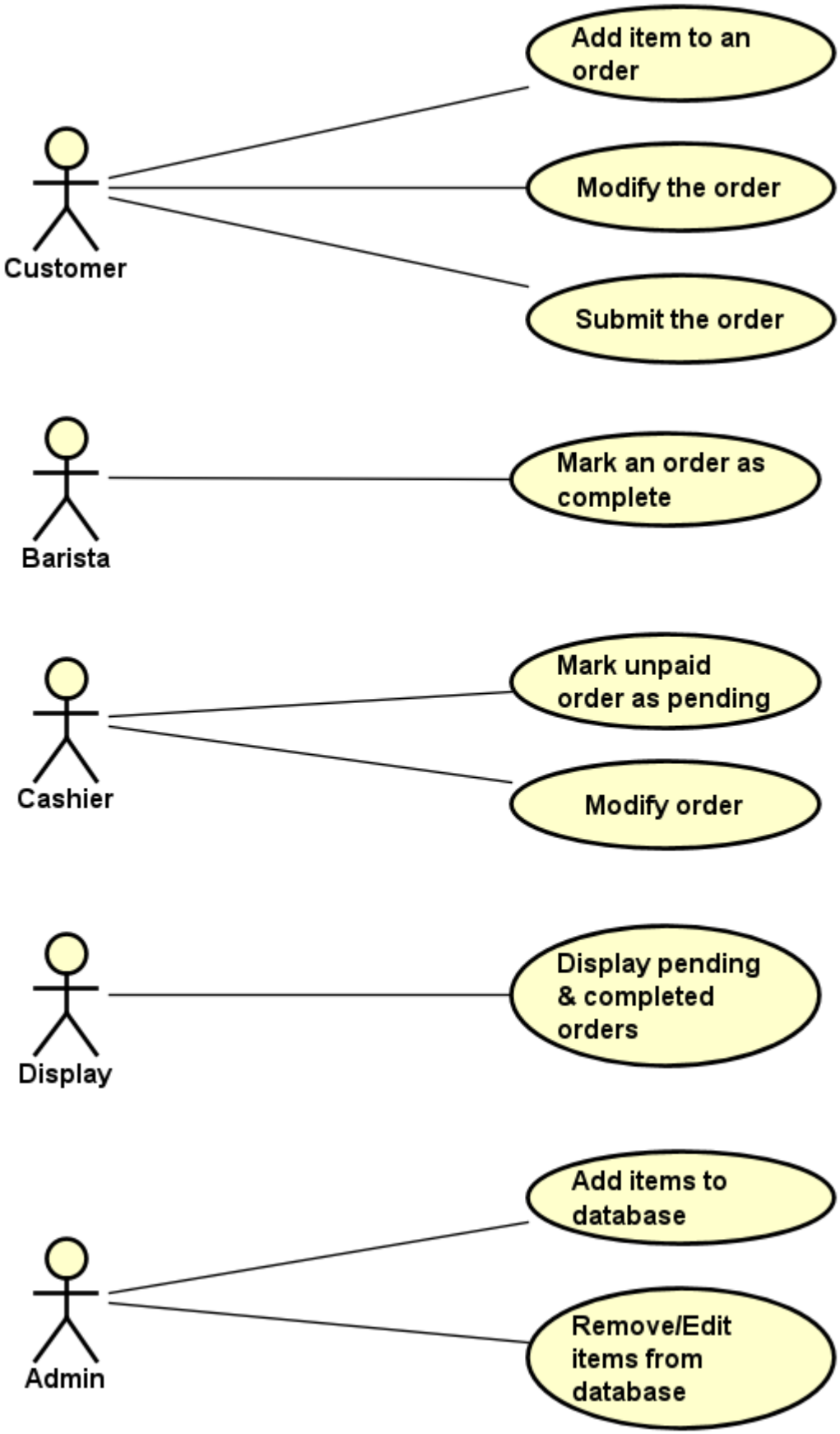
ID	PRIORITY	ESTIMATE	ITEM
0	Critical	96h	System foundation
1	Critical	96h	As a customer, I want to add an item to my order so that I can see a summary of my order later at checkout and buy it. When adding an item, it will be necessary to provide information about the item (id?), quantity and (optional) extras.
2	Critical	70h	As an admin, I want to add the item to the list of products of the Campus Café so that they will be visible for customers when they order.

			This would include adding products inside the list of products by providing the type (drink, snack,...), display name, price and description.
3	Critical	38h	As an admin, I want to remove the item from the list of products of the Campus Café so that they will no longer be available for customers when they order. This would include removing products inside the list of products by providing the type (drink, snack,...), display name, price and description.
4	Critical	46h	As a customer, I want to see every item available in the Campus Café so that I know what I may order. The items will be taken from the Campus Café's menu. When the customer starts the order, the items will be displayed by type and the basic information provided will be the name and the price. It is also possible to go into detail and see a description.
5	Critical	22h	As a customer, I want to submit the order by paying for it so that a barista can start making it.
6	Critical	22h	As a barista, I want to see all the pending orders so that I can start one at a time.
7	Critical	22h	As a barista, I want to mark an order as complete so that a customer can collect it.
8	Medium	48h	As the display monitor, I want to fetch and display both pending and completed orders for customers to know the status of their order.
9	Medium	22h	As a cashier, I want to accept an order when a payment is in cash so that it will appear for baristas.
10	Medium	22h	As a customer, I want to be prompted for extras to put in my drink (like syrups) so that my coffee will be more tasteful.
10	Medium	22h	As a customer, I want to be able to remove an item from my order during the ordering process in case I change my mind.
11	Medium	22h	As a customer, I want to be able to cancel my order at any time so that I am free to make that decision.
12	Medium	14h	As a customer, I want to be able to choose to pay in cash, MobilePay or card so that I have these options at my disposal.
13	Low	22h	As a cashier, I want to be able to edit an order's comment so that wishes from customers can be fulfilled.

14	Low	22h	As a cashier, I want to cancel orders when they are not paid for so that they will not use unnecessary resources.
15	Low	22h	As a customer, I want to be able to edit an item from my unsubmitted order, for example, choose a different syrup for my coffee, in case I change my mind.

No non-functional requirements were formulated.

USE CASE DIAGRAM

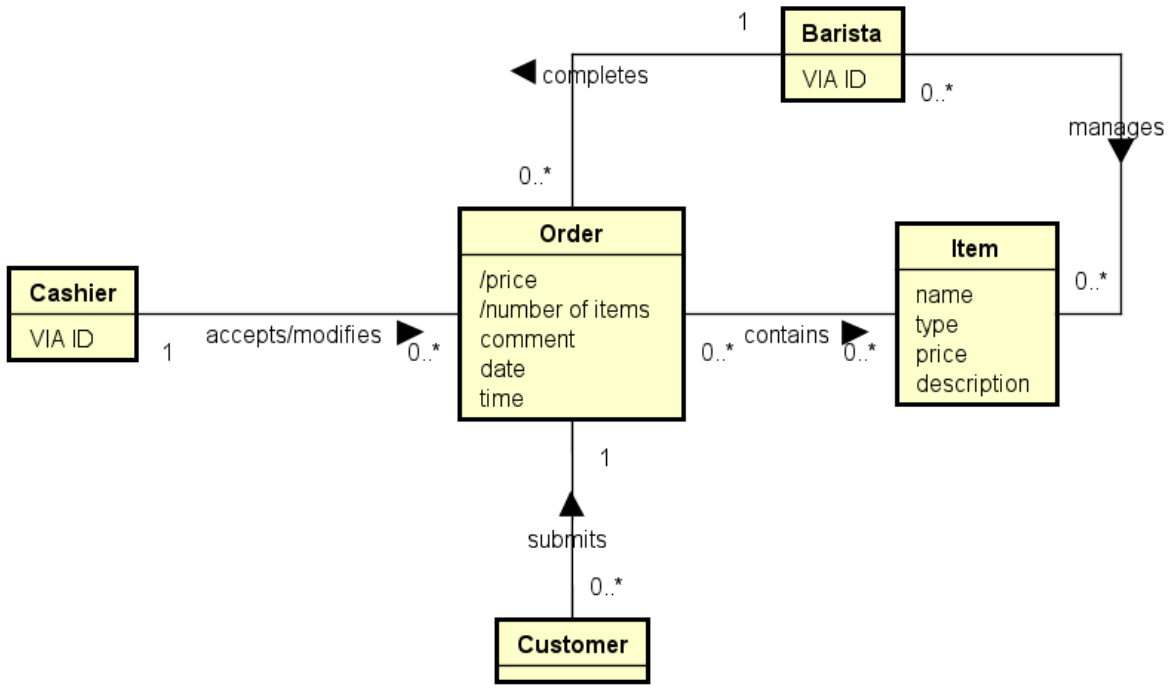


USE CASE DESCRIPTION Robert

Since the main purpose of the system is to create and manage the orders of the customers, the starting point, and the most relevant use case is adding items to the order. The main actor is the customer whose goal is to select and add an item to a new, or already existing order to be able to see a summary. In an ideal scenario the customer browses through the listed items and chooses one, specifying the desired quantity and extras, which they optionally add. Subsequently, the item is added to the order.

The precondition for the previous use case is that the items have to be in the system, which leads to the use case of adding items. The primary actor of this use case is the Admin, his role being adding items to the list of products from which the customers can later choose. The main success scenario is as follows: the admin picks the type of the item from a predefined list of types such as drink, snack, etc., enters the price with a description for it and confirms the changes.

DOMAIN MODEL



The picture above is the first Domain Model extracted from the Problem Domain description. This diagram will further be useful when designing a database and a class Diagram.