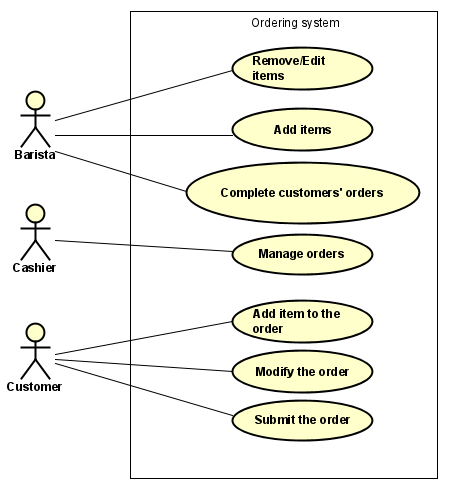
*Use case diagram*



*Fully dressed Use case description template*

| Use case section | Purpose/Guidelines |
| --- | --- |
| Use case name | Must be unique – start with a verb |
| Scope | The system under design (is often obvious) |
| Level | User goal or subsystem |
| Primary Actor | Who is necessary to start the Use Case |
| Stakeholders and interests | Who cares about this Use Case, and what do they want? |
| Preconditions | What must be true for the Use Case to start  (and worth telling the reader) |
| Postconditions | What must be true on successful completion (and worth telling the reader) |
| Main success scenario | A typical, “sunny scenario” of the Use Case |
| Extensions | Alternative scenarios of success or failure |
| Special requirements | Related non-functional requirements |

*Add items - Use case description*

| Use case section | Purpose/Guidelines |
| --- | --- |
| Use case name | Add items to database |
| Scope | Items |
| Level | The goal is to add items to the list of products by providing the type (drink, snack,...), display name, price and description. |
| Primary Actor | Admin |
| Stakeholders and interests | Campus Cafe wants to be able to add items from their menu to the system. |
| Preconditions | Have information about the item to be added. |
| Postconditions | The item is added to the list. |
| Main success scenario | 1. Pick the type of the item from a predefined list received from the system. 2. Type the display name for the item, the price which is a number and the description of the item. 3. Proceed with adding. 4. System displays all the information about the item. 5. Accept the changes. |
| Extensions | The actor can cancel at any time.  .  In step 1 if there is no accurate type, the actor can add the type.  In step 2 if the display name of the item already exists, the system displays the error. |
| Special requirements | Not relevant |

*Remove/Edit Items - Use case description*

| Use case section | Purpose/Guidelines |
| --- | --- |
| Use case name | Remove/Edit Items |
| Scope | Items |
| Level | The goal is to remove or edit items from/in the product list. |
| Primary Actor | Barista |
| Stakeholders and interests | Campus Cafe wants the system to reflect changes that are made to the menu. This can be achieved through functionality for editing and removing. |
| Preconditions | Items to be removed or edited. |
| Postconditions | The item is removed or edited from/inside the list. |
| Main success scenario | 1. Search for item. 2. SYSTEM will show the existing information in editable fields, along with the option to remove. 3. If EDIT, edit the fields to be changed. 4. If REMOVE, SYSTEM asks for confirmation of deletion. 5. Accept the changes. |
| Extensions | The actor can cancel at any time. |
| Special requirements | Not relevant |

*Add item to the order - Use case description*

| Use case section | Purpose/Guidelines |
| --- | --- |
| Use case name | Add item to the order |
| Scope | Orders |
| Level | The goal is to add the chosen items to an order. |
| Primary Actor | Customer |
| Stakeholders and interests | Customers want to add items to an order so that they can see a summary of the order with all the items they would like to buy. |
| Preconditions | The items have to be in the system |
| Postconditions | The chosen items are inserted into the order. |
| Main success scenario | 1. The customer browses through the items. 2. The customer chooses an item they would like to purchase. 3. The customer specifies the quantity. 4. The customer is prompted for extras for this item. 5. OPTIONAL: The customer adds an extra. 6. The item is added to the customer’s order. |
| Extensions | The item is out of stock and cannot be added to the same order. |
| Special requirements | Not relevant |

*Manage orders - Use case description*

| Use case section | Purpose/Guidelines |
| --- | --- |
| Use case name | Manage Orders |
| Scope | Orders |
| Level | Accept, cancel and edit orders |
| Primary Actor | Barista |
| Stakeholders and interests | Baristas and Clients so the orders can be nice and clear |
| Preconditions | An order has to be created |
| Postconditions | The order is accepted, cancelled, or adjusted in accordance with the customer’s wishes. |
| Main success scenario | 1. Choose whether to ACCEPT, CANCEL or EDIT 2. If ACCEPT, go to step 3, if CANCEL, go to step 5, if EDIT, go to step 7.   ACCEPT:   1. The barista receives the order and accepts it. 2. Go to step 10   CANCEL:   1. The barista receives the order and cancels it. 2. Go to step 10.   EDIT:   1. *The customer asks the barista outside of the system to edit something in his order.* 2. The barista updates the order according to the customer’s wishes. 3. Go to step 10. 4. End use case. |
| Extensions | None |
| Special requirements | Not relevant |

*Brief use case description*

**Complete customers’ order**

After physically preparing an order, a barista can mark it as complete so that it will become ready for pickup from customers.

**Modify an order**

A client can modify his order by removing or editing items (changing quantity or adding extras, e.g., syrup), or by making a comment in the order.

**Submit the order**

For an order to become available for the baristas or cashiers to handle, a customer has to finish it by paying for it.