

## Vending Coffee Machine at University

**Name:** Andrés Felipe Sánchez Sánchez

### Functional Requirements:

- System must show the user the offer of products (drinks of type coffee).
- System must receive the money of the user and count it.
- System must let the user choose the product of his/her preference.
- System must let the user choose additions for the product.
- System must supply the product if the pay of the user is enough.
- System must give the user the change of money not used for the buy.
- System must let the administrator user pick the collected money.
- System must let the administrator put ingredients of products in the machine.

### Non Functional Requirements

- The system should be available all time.
- The system should communicate with users through an user interface.
- The system should keep the paid money in a secure deposit and only let the administrator collect that money.

### Case Uses:

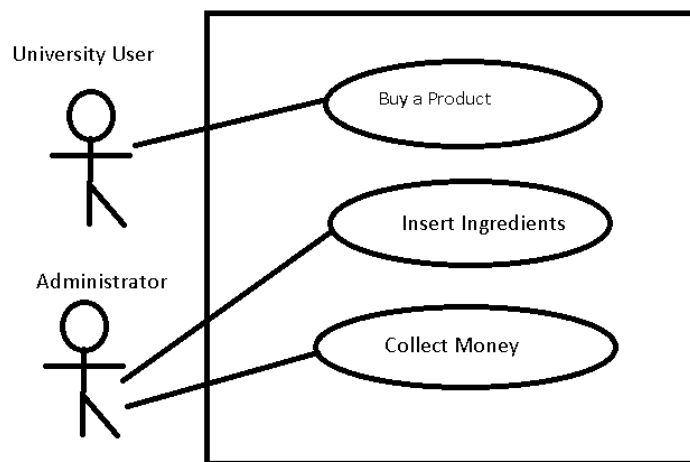
<b>Title</b>	<b>Buy a Product</b>
<b>Primary Actor</b>	University user
<b>Success Scenario</b>	The university user presses any key in the interface, then the system shows him/her the offer of available products (drinks) and their prices. When the user select the product, then the system show the available additions to that product and the user can optionally select some of them. After that the machine collect the money and verifies that the money is enough, then the machine cook the product with the additions and supply the product to the user. Finally the machine returns the cash not used in the purchase to user and save the money collected in the secure deposit.

<b>Title</b>	<b>Insert Ingredients</b>
<b>Primary Actor</b>	Administrator user
<b>Success Scenario</b>	The system administrator types a password and then the machine lets her/him insert ingredients.

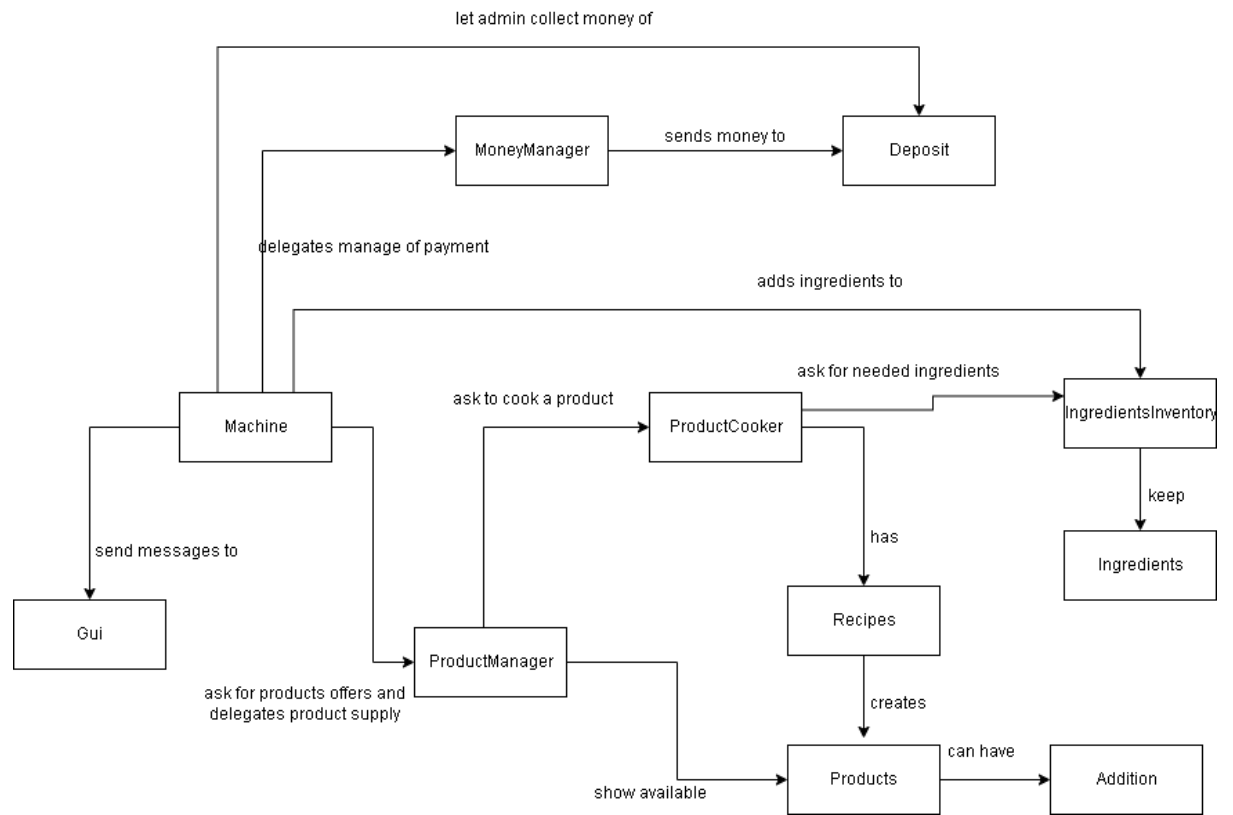
---

<b>Title</b>	<b>Collect Money</b>
<b>Primary Actor</b>	Administrator user
<b>Success Scenario</b>	The system administrator types a password and then withdraws an amount of money from the machine's deposit.

#### Use Cases Diagram



#### Conceptual Object Model



## Class Diagram

To get a best view please click [here](#).

To view the implementation in Java Code you can see the repo in the next link

ANDRES FELIPE SANCHEZ SANCHEZ / OOP Thinking · GitLab

