PART A

1. As a team, identify and describe the actors and construct a use case diagram. Describe what your use case diagram is showing (approx 100 words)

Firstly, the system will be developed to be used in a Garden Centre in order to manage all steps of a purchase, where the actors involved will be the Cashier, Customer, System Administrator and the Garden Manager.

The first actor to be described will be the Cashier, which will have most of the relations with the use cases due to the nature of the business, such as refund purchase, checkout purchase login etc. Moreover, the Manager is also able to relate with the same cases, however this actor will interact with specific use cases, (i.e., system users, check stock etc.). In terms of System Functionality, the System Administrator will be the one responsible for it by carrying out updates, fixing bugs and managing users as well. Lastly, the Customer will have a handy relationship within the system however, essential for the system usage since the Cashier will need initially a customer to successfully complete a purchase.

With that said, basically the use case diagram shows how the System will operate in order to efficiently manage a purchase process, maintaining the fast workflow and the organised storage of the items.

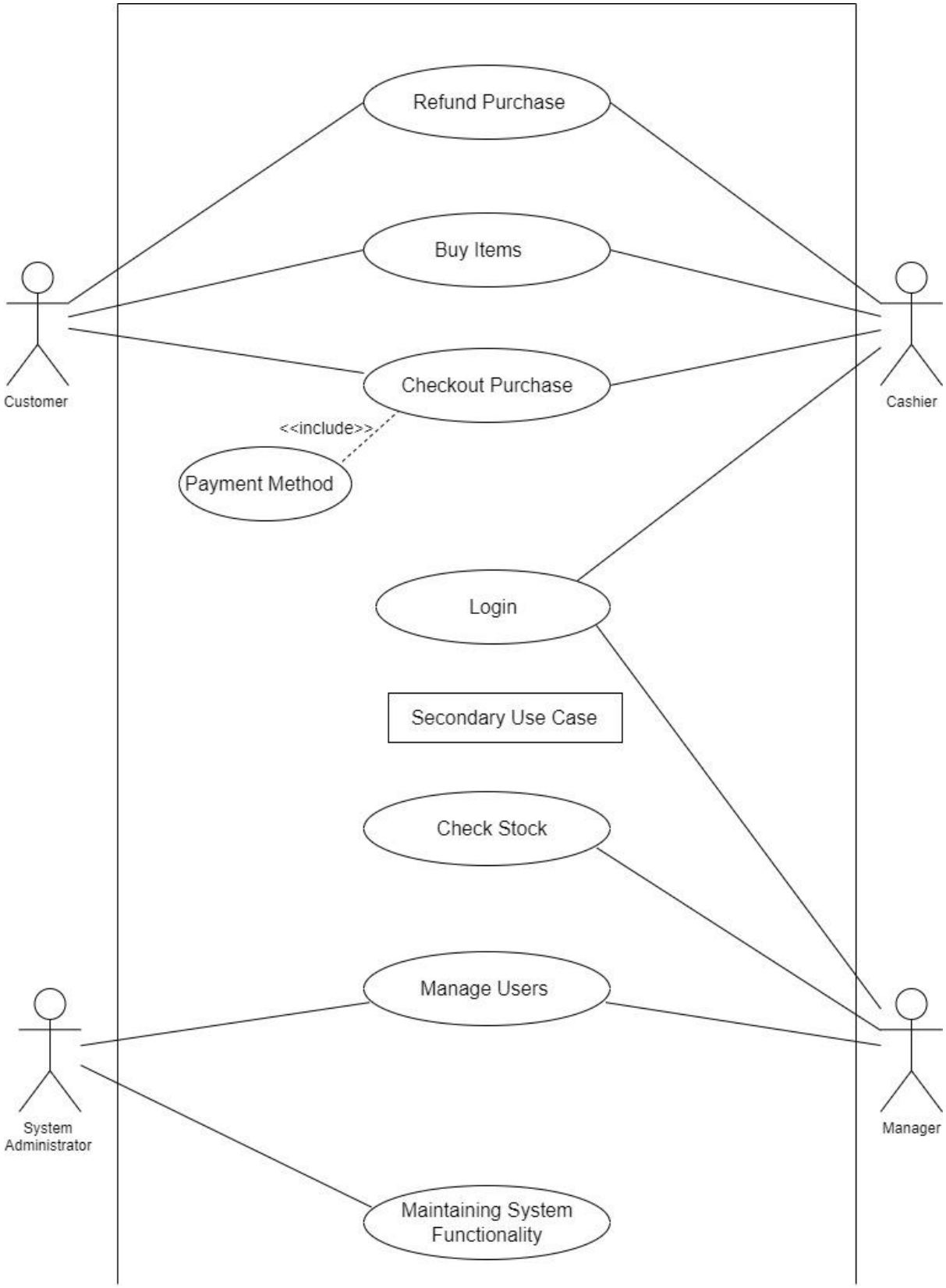

Figure 1: Use case diagram.

2. Each team member should select one distinct use case from the use case model and describe it in detail. The use case must contain an alternative flow or exceptional flow.

23146991 - Nickolas F. Franco

The use case described will be the Checkout Purchase, where a customer from time to time finds himself in a position that he can not successfully make the payment, the Cashier may cancel the purchase. Consequently, not achieving the use case's goal – exceptional flow.

Therefore, a successful Use Case would be considered where the Customer has his purchase completed as part of the normal flow, achieving the use case's goal.

23170981 – Sergio Oliveira

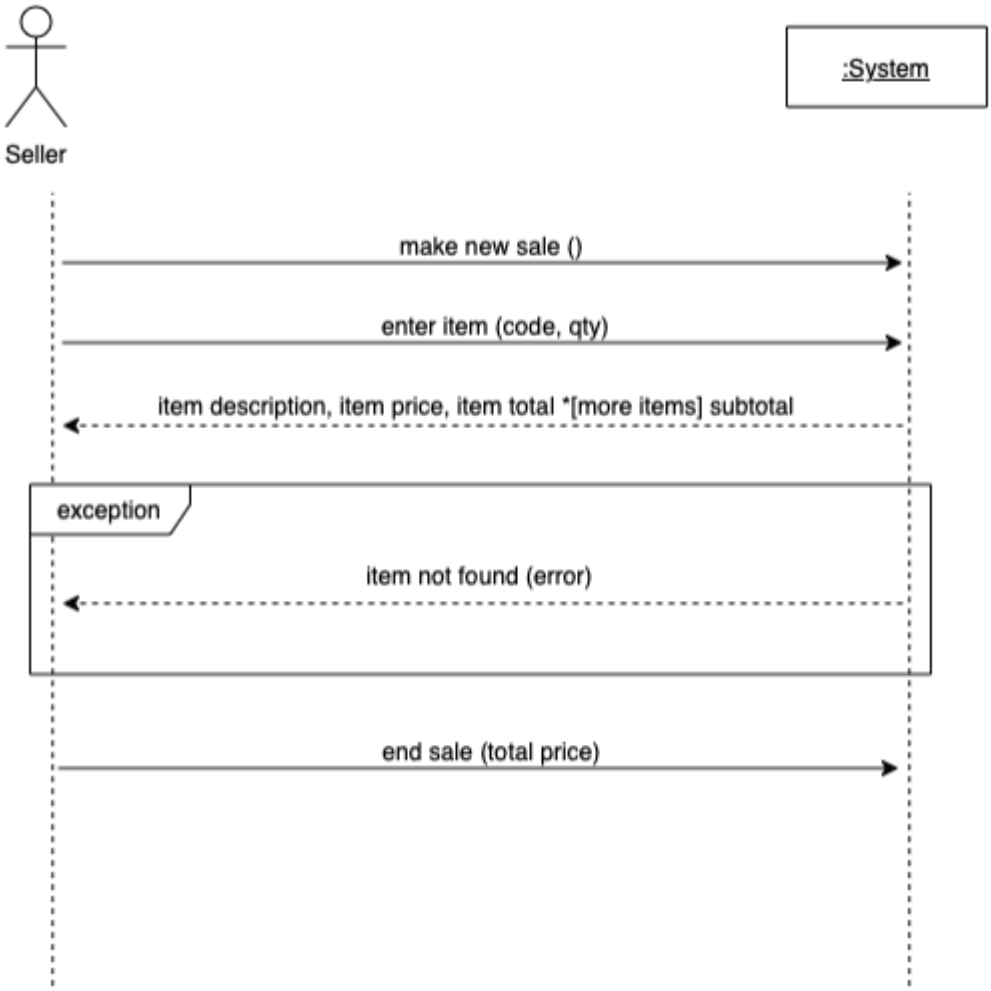

Figure 2: Use case diagram. By Sergio Oliveira.

USE CASE - BUY ITEMS

To describe my use case, I chose to apply the system sequence diagram.
Through it, you can see all the interaction between the actor and the system step by step.

The process of buying a product begins the moment the customer, who has no interaction with the system, arrives at the cashier to purchase an item. From there, the cashier has the responsibility to trigger the new sale process.

After that, the system will enable the user to enter the item through the bar code or description and its respective amount. Note that the system allows a sale to have one or more items. Consequently, a partial sum will be updated as the items are registered.

Therefore, after the end of the items which the customer wants to buy, the cashier will trigger the process of closing the sale, enabling a new step named checkout.

The exception granted in the system was attributed to the possible non-existence of a register for the item in which the customer wants to buy, thus returning to the system operator an error, making impossible the normal course of the system.

3. As a team, create a glossary in which all project-related terminology that requires clarification is both and fully defined.

Glossary: Garden Centre				
Use Case		Category	Type	Comments
Name	Description			
Employee	General idea that includes records of employees	ID	Attribute	To distinguish employees from each other
			Type	Unique Number
		Nome	Attribute	Personal registration of the employee in the company
			Type	String
		Address	Attribute	Personal registration of the employee in the company
			Type	String
		Phone	Attribute	Personal registration of the employee in the company
			Type	String
Garden Centre	Store responsible to sell garden items	Name	Attribute	Name of the company
			Type	String
		Address	Attribute	Address of the company
			Type	String
		Phone	Attribute	Contact phone number of the company

			Type	String
		Email	Attribute	Contact email of the company
			Type	String
Item	An Item for sale in a Garden Centre	Code	Attribute	To distinguish items from each other
			Type	Unique Number (Integer)
		Description	Attribute	Short description summaring
			Type	String
		Price	Attribute	Price of the item that will be charged
			Type	Integer

Glossary: Garden Centre				
Use Case		Category	Type	Comments
Name	Description			
POST	Point-of-sale-terminal(POST) - System used to record, calculate and handle payments	Sum_Item_price	Method	Sum each item and shows the grand total of the sale
			Type	Int
Sales	Sales transactions	Data	Attributes	Store data at the day the purchase was made
			Type	Date format (dd/mm/yyyy)
		Time	Attribute	Store data at the time the purchase was made
			Type	Time format (hh:mm)

		Identifier	Attribute	To distinguish sales from each other
			Type	Unique Number (Integer)
		Create_identifier	Method	The sales create itself the identifier when the trigger new sales is started
			Type	Unique Number (Integer)
Sales LineItem	Line create in sales transaction covering the quantity of the item	Qty	Attribute	The sales create itself the identifier when the trigger new sales is started
			Type	Integer
Specification	All informations that describe the specification and characteristics of the good			

Glossary: Garden Centre				
Actors		Category	Type	Comments
Name	Description			
Cashier	As a user, the cashier has a responsibility to records	Employee_ID	Attribute	To distinguish employees from each other
			Type	String
		Create_sales	Method	The cashier can create new sales
			Type	New sales transaction

	the items and/or search items asked by the customer and collect the payment	Login	Method	The cashier enter in the system to access functions such as create sales and search items
			Type	Boolean, the user has access or not
		Search_Item	Method	The cashier can execute this command to find item asked by the customer
			Type	Boolean
Customer	Customer arrives at checkout in the Garden Centre with items to purchase	Name	Attribute	Attribute solely used by the cashier if Customer has a Garden Centre's account.
			Type	String
Manager	Responsible to manage the application adding and deleting users	Employee_ID	Attribute	To distinguish employees from each other
			Type	String
		Add_User	Method	Create new user to get access the system
			Type	Use a method to create a new value for an Attribute.
		Drop_User	Method	Delete user that does not belong the process anymore
			Type	Use method drop to delete values for a Attribute

4. As a team, create a conceptual class diagram modelling the architecture of the proposed system. The conceptual class diagram should demonstrate the use of three or more of the following.

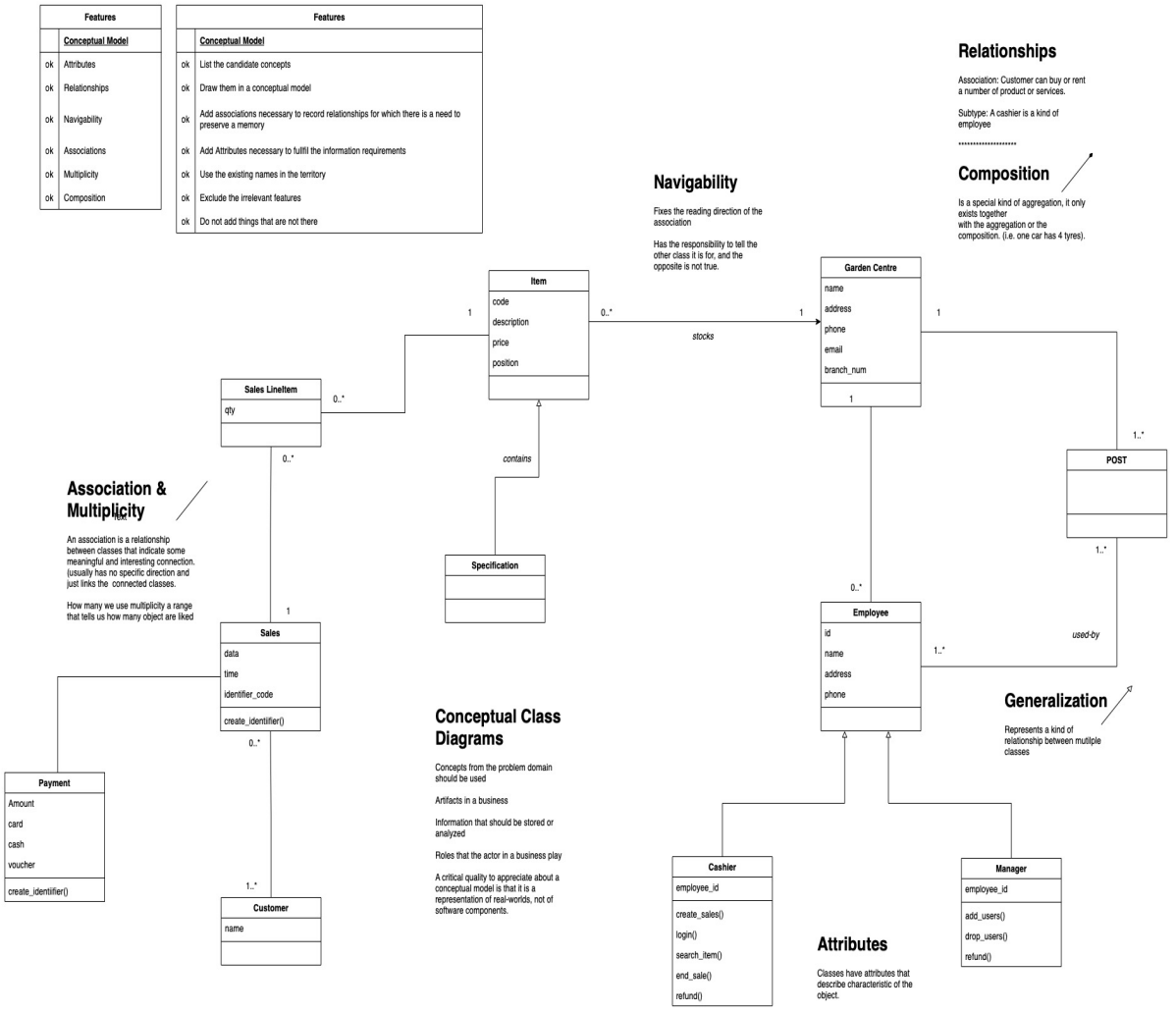

Figure 3: Conceptual Class Diagram.

The Conceptual Class Diagram basically is showing how the entire system will operate in order to successfully complete or not a purchase made by the Customer, considering as well the Staff's interaction solely with the System. Concepts from the problem domain were also used for the development of the current project.

Furthermore, the Conceptual Class Diagram presents the interactions between the classes, objects, attributes and the information that should be stored/analyzed. As well as the methods and actions that few classes and actors may have throughout the process.

In conclusion, Conceptual Class Diagram describedIn conclusion, the Concept Class Diagram described the critical quality of the conceptual model to represent a real world in which the concepts of associations, multiplicities, relationships, compositions, navigability and generalization were assigned.

5. Each team member should select one of the system operations and should develop a contract for it.

23146991 - Nickolas F. Franco

Contract	
Name	**receivePayment** (cash, credit/debit card, voucher: doubles)
Responsibilities	Record payment method, add it to the sale and create an instance of Payment. Display confirmation of payment and print receipt option.
Exceptions	If the payment can not be completed, the cancel option is available for selection.
Type	System
Pre-conditions	Finish checkout process after matching quantity, description and price with the purchase.
Post-condition	Select and confirm whether Customer has successfully paid for the purchase.

- If a new purchase, a new Sale was created (instance creation).
- If a new purchase, the new sale was associated with the POST (association formed).
- A SalesLineItem was created (instance creation).
- A Payment was created (instant creation).
- A Receipt was created (instant creation.

STUDENT 23170981 – SERGIO OLIVEIRA

Contract	
Name	Create Sales
Trigger	The customer arrives for the checkout process.
Responsibilities	Enable the sales process. Record the date and time when the process was started.
Precondition	Cashier has a access authorisation by login
Postcondition	The user can initiate the process of entering the items
Exception	The user does not have access
Type	System

Figure 4: Contract. By Sergio Oliveira.

6. Using appropriate design patterns, each team member should create a communication diagram based on the contract they developed in task 5, including a description of what the diagram is showing. (approx 100 words)

23146991 - Nickolas F. Franco

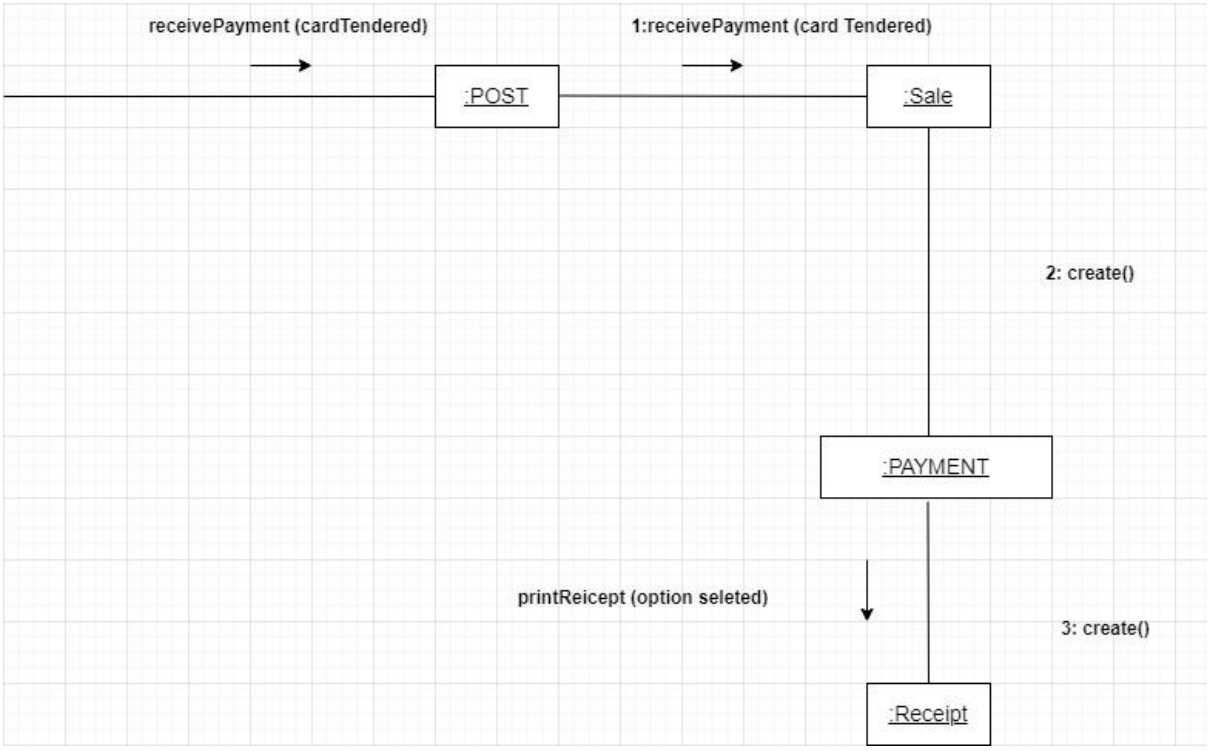

Figure 5: Communication Diagram. By Nickolas F. Franco

The Communication Diagram will show how an operation Receive Payment will occur and how the messages will flow through the Objects and Instances of Objects since the initial step – where the payment method has to be selected – the process involving other instances and lastly how the state operation finishes with the receive Payment Operation concluded – considering the state of the system and the post conditions.

Firstly, the message *receivePayment* is sent to an instance of a POST. This is sent after a Customer has finished his purchase in-store and the Cashier has

confirmed alongside the attributes entered such as quantity, type and price. It is important to mention that POST corresponds to the *receivePayment* system operation.

Secondly, the POST objects will send the message to a *Sale* of Instance, that will be recorded and sent to the sale. Next, the Sale Instance creates an instance of *Payment,* that creates a secondary message (i.e., *printReceipt*) allowing Cashiers to print a receipt or not – depending on the Customer demand.

STUDENT 23170981 – SERGIO OLIVEIRA

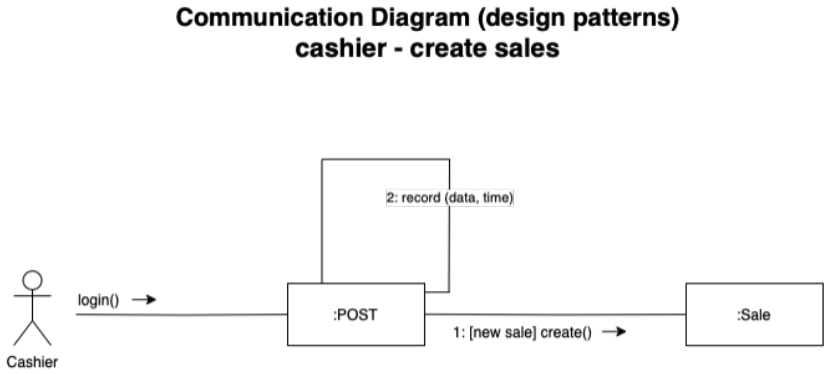

Figure 6: Communication Diagram. By Sergio Oliveira

Create Sale

The process described in the communication diagram here describes the interaction of the users from login to the process of entering items.

The use case relationship includes the client by association, who does not have direct interaction with the system but rather plays the role of the trigger for the start of the process.

Starting from the principle where the user is enabled to access the system, it is possible to simplify the process into two steps, which are the creation of the sale and the storage of the moment of its start.

To put it simply, the goal of this process is to enable the system for the user, role played by the cashier, to input the items that will be purchased.

The process described follows an abstract idea, being possible to replicate the patterns in all cases where a user is accessing a system, digitizing the item in search of obtaining the total sum.

7. Discuss how risk, quality and communication will be managed in your project. Provide justifications for your choices

Risks, quality and Communication will be all managed upon the Agile principles as they have useful practices rather than traditional approaches, which can have a considerable level or risk, cost and less efficient in comparison to Agile principles

With that said, throughout the software development involved risks may occur (i.e., confusing actors, miscommunication etc.) however, they can be significantly reduced by a proper amount of analysis and proper use of design modelling. As well as small incremented deliveries along with timely feedback between customer and client may reduce the chances of a client's dissatisfaction at the end of project.

Moreover, our project will be based upon the eleven Agile principles that are distinguished from any other processes and have a substantial impact on the quality of the project. Basically, there are two principles that take the quality into accountability. Customer satisfaction through early and continuous delivery of valuable software and a constant but suitable pace from the development team involved in the project, should be an excellent approach for quality management.

Lastly, Agile principles will be used for communication as part of the project management, preferably face-to-face conversations between the participants. Curiously, conversation will be a default standard as written plans, design etc. will in case of an immediate and significant necessity. Worth also to mention that constant communication and feedback will guide the project success for the business, especially in the execution phase.

In other words, risks, quality and communication are parameters to be considered part of a project management. Therefore, Agile principles and its practices have been and shown the most suitable and efficient approach, even used by big Techs (i.e., Google, Microsoft etc.). currently and good practices within Object Oriented Software Engineering for software development.

8. Describe and justify the development methodology you will follow.

Through the definitions of success, challenge, and loss, we were able to achieve personal, technical, and organisational goals through the Agile methodologies usage for the project.
In short, to justify such a choice these principles seek to deliver the project with all the features provided – *Successful, Challenged and Impaired*—within the time enabled.

In addition, the selected Agile philosophies support the combination of Extreme Programming and Scrum methods, which here were applied individual elements such as version control and the application of coding patterns, where moulded to our reality enabling the development, interaction between those involved and monitoring through weekly dismantling of the project in search of customer satisfaction.

Due to the constant change in the market and the need for adaptability due to the economic scenario, the difficulty of managing a Garden Centre becomes more real and challenging. For this reason, applying the XP development methodology allows us to eliminate these constant changes and requirements of the market and consequently achieve the desired success. The method's efficiency makes it possible to eliminate delays and miscommunication by emphasising face-to-face collaboration and thus allowing constant evolution by performing simultaneous phases, such as analyses, projects, coding, testing, and deployment.

Moreover, by having both of the methodologies selected, a team can break work into small and manageable pieces for a certain period, a so-called *Sprint* – completed within two or four weeks – with potentially shippable code. Also, teams may be small but well-organised, cross-functional and self-organised. Additionally, items are organised in a list based on priorities that are estimated with the relative effort of each one.

In summary, the reapplication of patterns enables the practicability of the methods and concepts mentioned earlier. In addition, the construction of the use case in the first stage of this project demonstrates the application of those methodologies for this concept. Based on the list of common objects identified, a significant amount of considerations followed by an analysis performed, allowed the team to develop a more robust solution to the domain problem.