Lab Report 1(CA2.1) Use Case Modelling

Student Name: Aleksejs Polikarpovs

Course: Contemporary Software Development

Module: PROC_IT801 Software Processes

Lecturer: Angela Sweeney

Submission Date: 13/10/2021

Contents

Aim

Aim is to produce appropriate use case diagram based on provided requirements and document use-case flow of events for Generate/Place order use case. Produced use-case diagram will be used in future development of XYZ online shop and other UML diagrams will be created based on final result.

Produced diagram will give a clear overview of the system and use-case description document shows main flow and alternative ways of flow, and how they should be handled.

Pre-requisites

To complete the report following items will be required:

- system requirements with a full description of how a system should operate
- use-case flow of events document
- Visual Paradigm modelling software

Methods

What is Use-case diagram and why it should be created

In UML (Unified Modelling Language), a use-case diagram model is the behaviour of a system and it helps to capture the requirements of the system. A use-case model is built through an iterative process, and the diagram is a graphical representation of a user's possible interaction with a system.

This graphical representation:

- defines who exists outside the system – actors
- identifies actors who interact with the system
- illustrates a set of use cases for a system
- defines what should be performed inside the – system use cases

Each use-case is documented with a textual description in a use-case flow of events document. The document should be structured as:

- use-case name
- objective

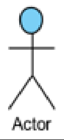
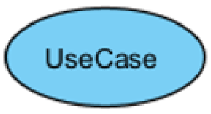
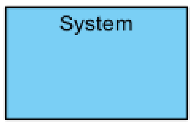
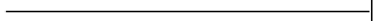
- precondition
- main flow
- alternative flow
- post-condition
- any additional requirements(optional)

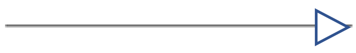
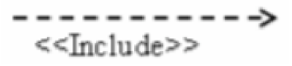
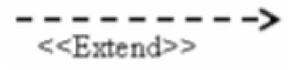
Use-case diagram might be changed during project development, as requirements might change or use-cases are defined better.

Process of creation use-case diagram

- during interview with the stakeholders of the system, text with all requirements is created. Stakeholder should clearly explain what the system should do and who will use the system
- identifying actors
- identifying use-cases
- identifying association between actors and use-cases
- identifying relationships between use-cases
- identifying system boundary

Use-case diagram components

Component	**Description**	**UML Representation**
Actor	External entity outside the system which will interact with the system	
Use-Case	Represents the functionality from the viewpoint of the user of the system	
System Boundary	Boundary between system and actors who interacts with the system	
Association	Interaction of an actor and a use-case	

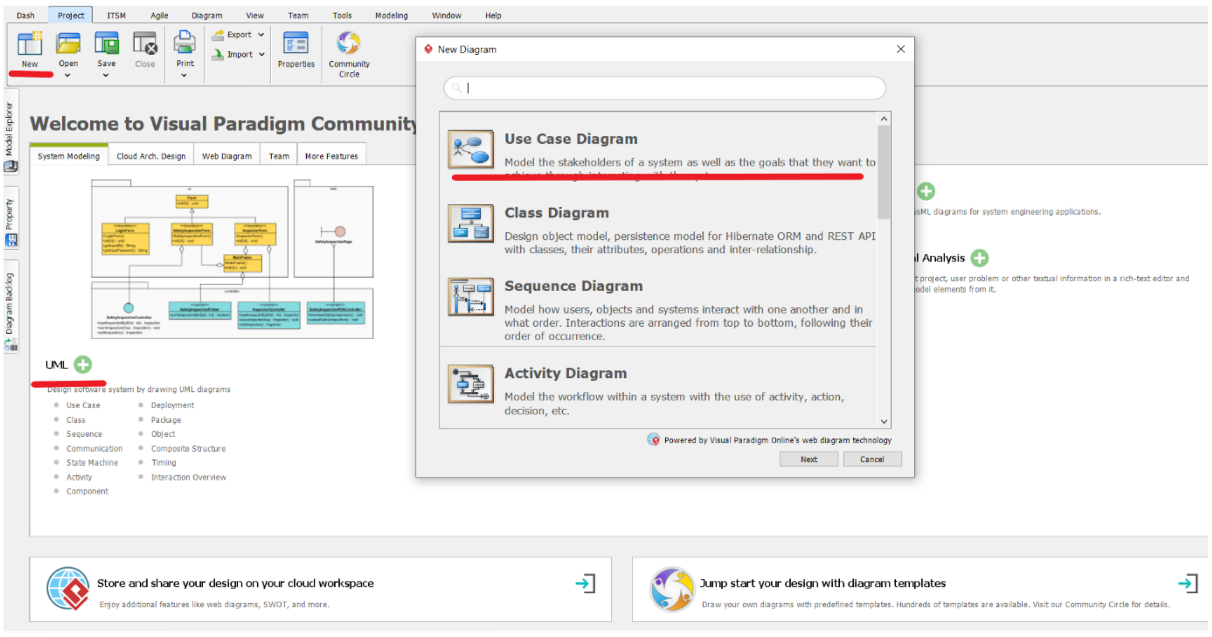
Generalization	Relationship in which one model is based on another model	
Include	Relationship in which one use-case includes functionality of another use-case	
Extend	Relationship which describes that one use-case extend behaviour of another use-case	

Development environment

To create UML use-case diagram we can use old fashion way of using pen and paper, but digitally documented file will benefit from development with easy sharing and it will be more secure to store process on several devices. Visual Paradigm is a great application for creating any type of diagrams and it has Communal Edition version which is free.

To create a new Use-Case diagram we should:

1. Select New button and give name to the project
2. Press UML +
3. Choose Use-Case diagram
4. Press Blank (or choose any other depends on the project)
5. Name our diagram
6. Select OK



Design software system by drawing UML diagrams

Use Case Diagram
Model the stakeholders of a system as well as the goals that they want to achieve through interacting with the system.

Class Diagram
Design object model, persistence model for Hibernate ORM and REST API with classes, their attributes, operations and inter-relationship.

Sequence Diagram
Model how users, objects and systems interact with one another and in what order. Interactions are arranged from top to bottom, following their order of occurrence.

Activity Diagram
Model the workflow within a system with the use of activity, action, decision, etc.

Powered by Visual Paradigm Online's web diagram technology

Store and share your design on your cloud workspace
Enjoy additional features like web diagrams, SWOT, and more.

Jump start your design with diagram templates
Draw your own diagrams with predefined templates. Hundreds of templates are available. Visit our Community Circle for details.

Figure 1. Creating a new project

Now we can start to create our Use-Case diagram. Toolbox located on the left-hand side, has all necessary features.

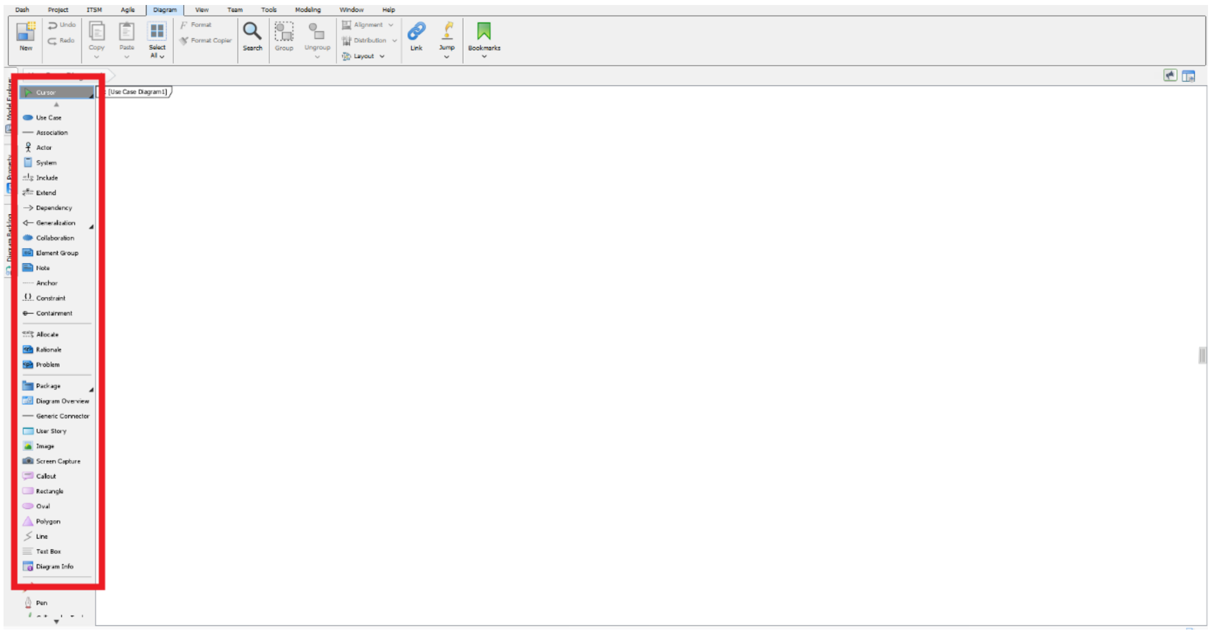


Figure 2.Use-Case diagram toolbox

XYZ Online web purchasing requirements

To use the system the customer must log in providing their user id and password. If a customer hasn't registered with the company, then the system registers the customer and records details on their name, address, telephone number and email address. Customers are corporate customers and individual customers. Corporate customers are billed monthly (rather than be charged by credit card). The credit card number of individual customers is recorded.

Customers manage their shopping basket by adding items and removing items. Each customer has exactly one shopping basket. The shopping basket consists of no items or many items that the customer may wish to purchase. Detailed held on the items are the quantity ordered and the line price. Each item refers to one product. An administrator records the number of products available, title, copyright and unit price. For books the author is recorded, for recording products the artist and running time and for software the product version is recorded.

When a customer completes their shopping basket and is ready to purchase, they proceed to checkout and an order is generated. Customers can place many orders, however an order is associated with just one customer. Details held on an order are the order number, order date, date shipped, the order status and total price. Orders can contain many items and a customer can add items, remove items from the order, print order details if they choose to do so and charge their credit cards or account depending on what type of customer they are. Authorisation is required by the credit card authority to make a payment if they are individual customers as opposed to corporate customers where they are billed monthly. The administrator produces sales reports for management meetings that reports on products that are selling well. Some products may be discontinued if not producing appropriate sales levels.

Identifying actors

Based on these requirements we make search in the text for actors which can be people or other systems who interact with the system.

I marked actors with turquoise colour (please see XYZ Online web purchasing requirements**)**. To add actors, we can simply drag and drop actor shape from left side toolbox in our diagram. Each actor should have appropriate name representing what his role is in our system (don't give your actor's real names). In current situation some actors might be related, like Individual customer and Corporate customer, both are Customers. We should model this relationship using generalization. There is another possible generalization between Administrator and Manager (both of them are XYZ Staff), but for diagram simplicity I will not use generalization in this case.

Actor Name	Additional Notes
Customer	Actor requesting service from XYZ online web shop
Individual Customer	Descendant of the Customer
Corporate Customer	Descendant of the Customer
Manager	Actor requiring system to perform tasks
Administrator	Actor requiring system to perform tasks
Credit Card Authority	Actor requiring system to perform tasks

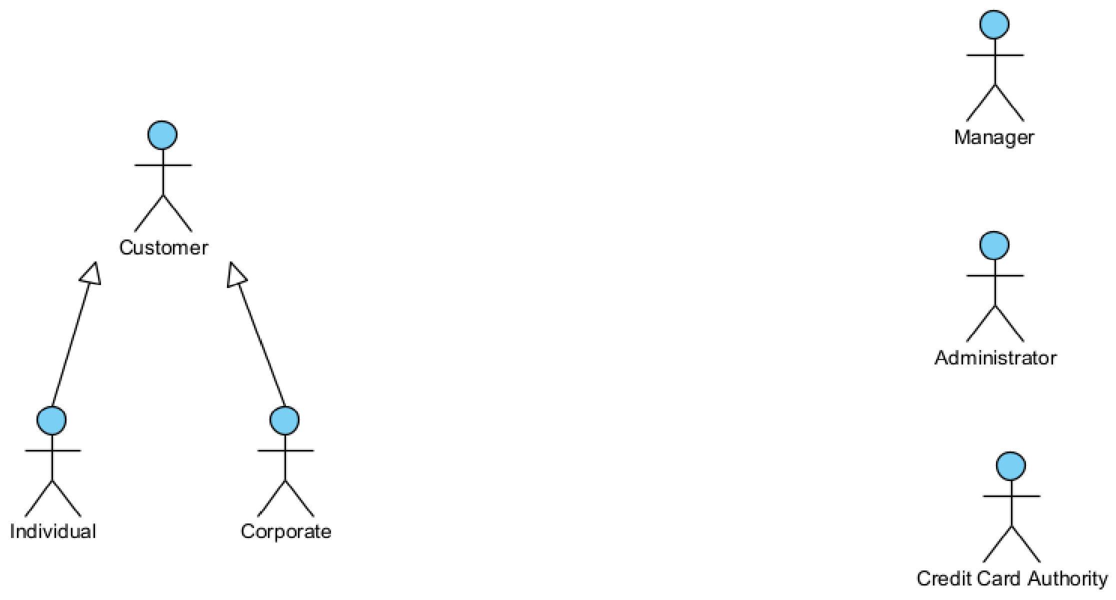


Figure 3. Diagram after all actors is added

Identifying use-cases

Use-cases represents the functionality from the viewpoint of the user of the system and they are adding value because they help explain how the system should behave.

Use cases name should begin with a verb and the name should be descriptive, for example “Add Item” or “Produce Sales Report”.

I marked use-cases with bright green colour in requirements (please see XYZ Online web purchasing requirements). Now we need to pick actors and their goals (use cases).

Actor	Goal
Customer	Login into System
Customer	Authenticate ID and password
Customer	Register New Customer Details
Customer	Manage Shopping Basket
Customer	Add Item
Customer	Remove Item
Customer	Proceed to Checkout
Customer	Generate / Place Order
Customer	Print Order Details
Corporate	Add to Corporate Bill
Individual	Pay by Card
Credit Card Authority	Authorize Card Payment
Administrator	Update Product Details
Administrator	Produce Sales Report
Administrator	Discontinue Product
Administrator	Charge Corporate Customer

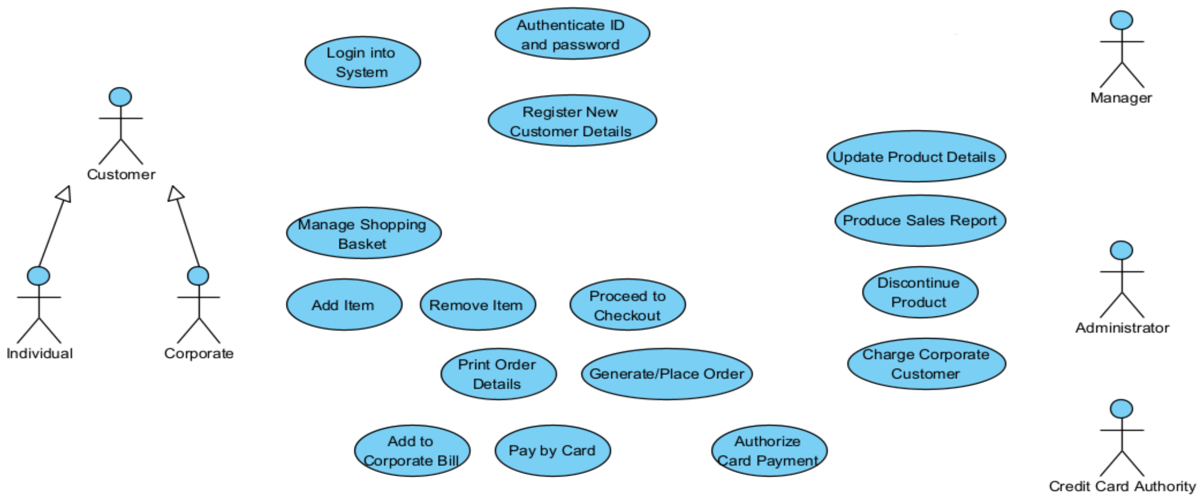


Figure 4. Diagram after all use-cases is added into the diagram

Use-case association

Use cases might be related. There are three types of relationships:

- Generalization relationship represents object inheritance where one use-case is based on another
- Include relationship that describes a situation in which a use case contains behaviour that is common to more than one use-case
- Extend relationship describes a situation in which one use case extends the behaviour of another use-case

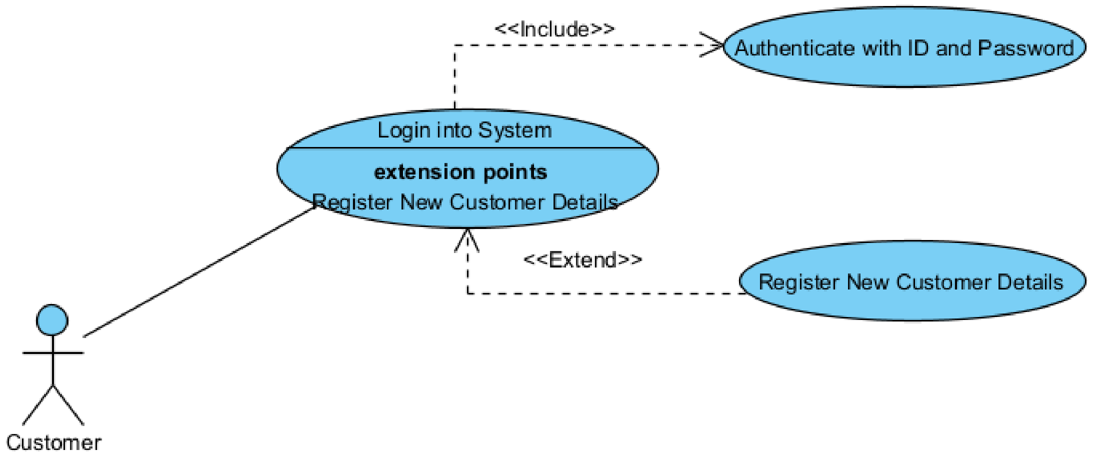
Now we need again to go through the requirements list and add appropriate associations for each use-case depending on requirements.

1. *To use the system the customer must log in providing their user id and password.*

   Customer has an association with Login into System use-case. Login into System use-case has <<include>> relationship with Authenticate with ID and Password use-case.

2. *If a customer hasn't registered with the company, then the system registers the customer and records details on their name, address, telephone number and email address.*

   Login into System has <<extend>> relationship with Register New Customer Details.



<<Include>>
Authenticate with ID and Password
Login into System
extension points
Register New Customer Details
<<Extend>>
Register New Customer Details
Customer

Figure 5 demonstrates step 1 and step 2 in use-case diagram

3. *Customers manage their shopping basket by adding items and removing items.*

Customer has association with Manage Shopping Basket use-case. Manage Shopping Basket has <<extend>> relationship with Add Item and Remove Item use-cases.

4. *<u>An administrator records the number of products available, title, copyright and unit price.</u>*

   Administrator has association with Update Product Details. Those product updates will be available for customer, so customer and administrator have association with Browse Product Catalogue.
   Browse Product Catalogue use-case is not mentioned in requirements, but I think this way is easier to understand that all changes made by administrator will be available for customer.

5. *<u>When a customer completes their shopping basket and is ready to purchase, they proceed to checkout and an order is generated.</u>*

   Customer proceeds to checkout from shopping basket, when he is happy with items, he added into his shopping basket. Proceed to Checkout has <<extend>> relationship with Manage Shopping Basket use-case.
   You can't generate order with zero items in a shopping basket, so logically for generating order at least one item should be in the shopping basket. Generate/Place Order use-case <<extends>> Proceed to Checkout use-case with one condition that in the shopping basket should be at least one item.

6. *<u>Orders can contain many items and a customer can add items, remove items from the order, print order details if they choose to do so and charge their credit cards or account depending on what type of customer they are.</u>*

   Generate/Place Order use-case has <<extend>> relationship with Add Item, Remove Item, Print Order Detail. In a use-case flow of events I documented that if an order was generated, system performs customer type check to choose designated payment type (Individual customer – Pay by Card, Corporate customer – Add to Corporate Bill).

7. *<u>Authorisation is required by the credit card authority to make a payment if they are individual customers as opposed to corporate customers where they are billed monthly.</u>*

   Corporate customer has association with Add to Corporate Bill use-case, also as administrator, who will know every time when corporate customer orders anything from a store and will record the purchase on a corporate account. Administrator will generate invoice on a due date for Corporate customer. Administrator has association with Charge Corporate Customer use-case.

Individual customer has association with Pay by Card use-case. And it has <<include>> relationship with Authorise Card Payment use-case, which has association with Credit Card Authority.

8. *The administrator produces sales reports for management meetings that reports on products that are selling well.*

   Administrator has association with Produce Sales Report use-case. If the report is produced for the manager, manager has the ability to get a sales report. Get Sales Report use-case has association with both manager and administrator.

9. *Some products may be discontinued if not producing appropriate sales levels.*

   Administrator have association with Discontinue Product use-case.

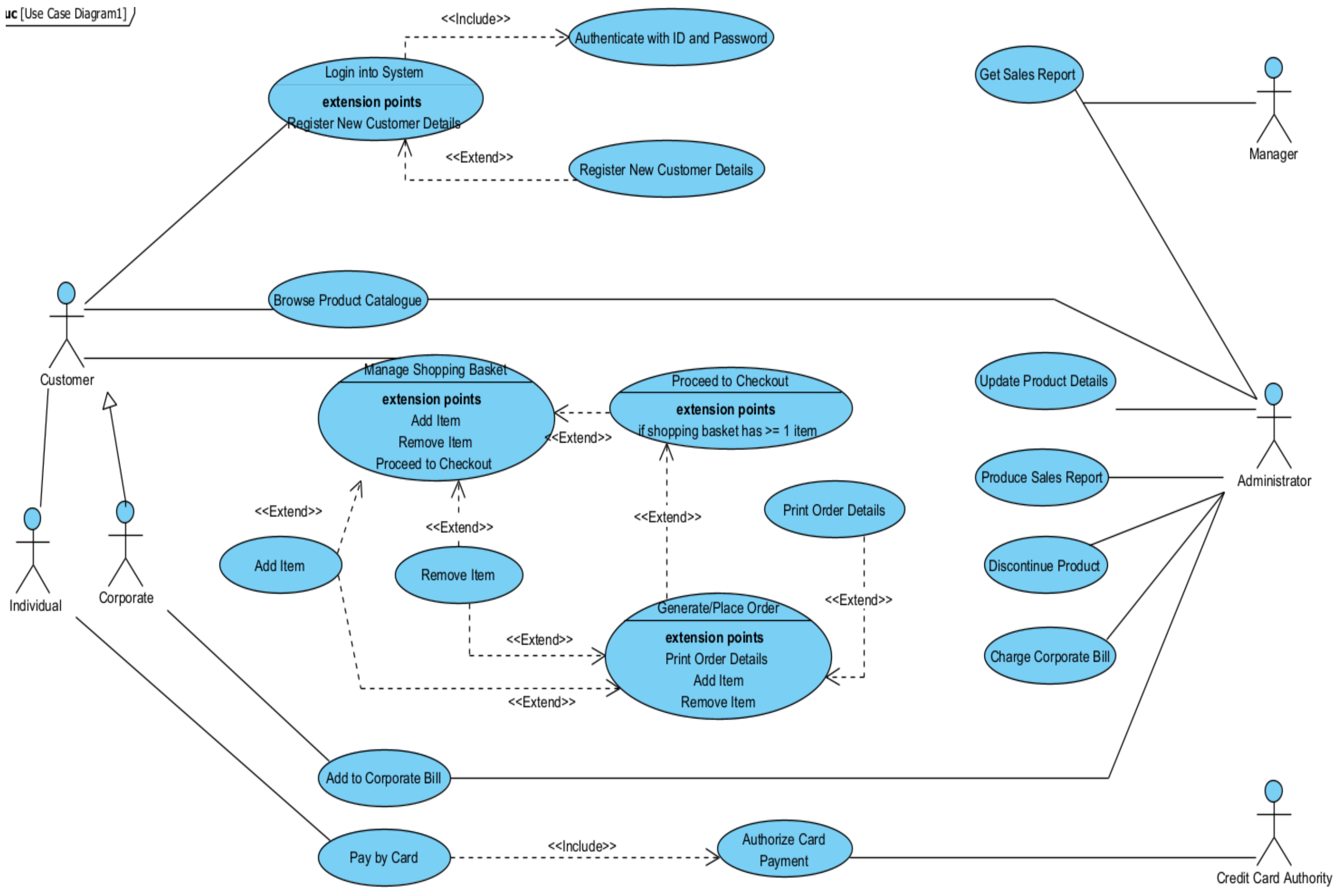


Figure 6. Diagram with all associations

Documenting the system boundary

From our progress in diagram, we can clearly see that our system might be divided in two functional parts:

- Product store where customer browsing products, create orders and purchase them
- Store Management where administrator and manager can perform their duties

From toolbox located on a left-hand side we can choose Package tool and highlight each area of functionality. Packages used to group elements and to provide a namespace for the grouped elements.

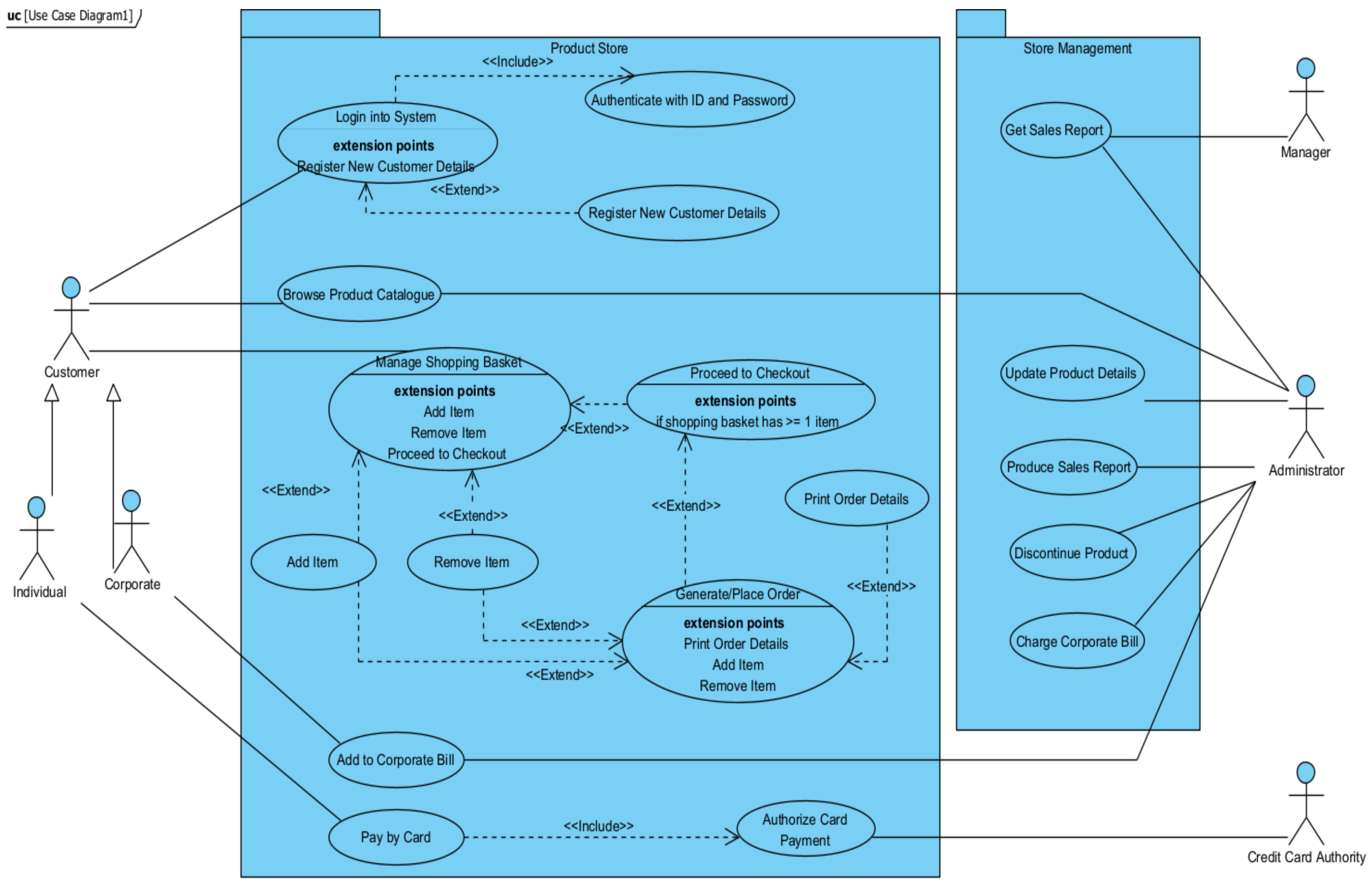


Figure 7. Complete use-case diagram

Use-case description

A text-based use case description can be used to provide additional information to support the use case definition. Use-case description provides details on the flow of processes, the expected results and how exceptions or alternative flow is handled. Each use-case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled.

It is good practise to document each use-case, but in this report only Generate/Place Order use-case is documented.

Use-case Flow of Events Template

Use Case Name	Generate/Place Order
Objective	Generate/Place Order is used to complete an online purchase
Precondition	Shopping basket should have at least one item and customer must proceed to checkout. System must be operable.
Main Flow	1. Order is placed from registered account. 2. Order confirmed by the customer. 3. System performs customer type check (corporate or individual) and chooses designated payment type. 4. Order information is provided on a customer screen.
Alternative Flow	At point 1 or above: • Customer should authenticate his account with ID and password. • If customer doesn't have a registered account, new account registration option is provided. • New customer should select an appropriate account type and fill all required details. • During the registration customer might press Cancel button and proceed to Login screen. At point 2 or above: • customer might change quantity on item or remove item. New total price will be calculated depending on customer modification. • customer might print order details. • customer might press Cancel button and proceed to product catalogue screen. At point 3 or above: • authorization performed by Credit Card Authority is unsuccessful and error message is displayed. Customer is advised to repeat Credit Card Authority authorization again.

	• customer might press Cancel button and proceed to product catalogue. At point 4 or above: • error message is displayed with suggestion to contact helpdesk@xyz.ie
Post-condition	Order is placed.
Non-Functional	System should be available on mobile devices.

Result/Discussion and Conclusions

Based on my research all types of UML diagrams **should be** made to deliver **valuable final product** for the customers. Developers will have a good UML view of the system based on UML diagrams, before they start actual coding. That might take more time for development, but it will pay off with less troubleshooting later in the project.

What happens if the delivered system doesn't have some important features for the customer? We code for people, and we want them to enjoy our final product. So, in that case, the system might need to be rebuilt again, and the total of our project price will definitely go up.

Use-case diagram is a starting point for any project based on UML model. It describes user perception of functionality in domain or system. It models use-cases and each use-case should be documented in a use-case description.

After use-case view is done, the next step will be Logical View, structured on produced use-case view.

Of course, requirements might change during development, but if the development is based on a UML model there is a big possibility that all necessary requirements will be found.

Building systems is not easy, and it should be taken very seriously. The structured process that UML model is providing, makes it easier to stay on track during development.