Use Cases for

Master of Science in Information Technology

Software Design and Programming

Joseph DiBiasi, Estell Moore, James McKenna, Anthony Martuscelli

University of Denver College of Professional Studies

9/13/2025

Faculty: Nirav Shah, M.S.

Director: Cathie Wilson, M.S.

Dean: Michael J. McGuire, MLS

Abstract

For the purposes of our scenario for developing object oriented methods for a shopkeeper who would like to manage, log, record, and interact with vendors. Our use cases will cover interactions with the vendors, updating and maintaining records and vendors, and the notifications and statuses that coincide with those vendors. We've developed multiple use cases that should cover the use cases in the industry, our difficulties with developing those use cases, and our overall conclusion in regards to the scenario.

CONTENTS

Introduction

For this project, a specific scenario for designing a software to manage vendors for a shopkeeper was selected to brainstorm use cases for. In a group, multiple use cases were defined and expanded on through different alternative flows and Object-Oriented Programming (OOP) methods applications. During the planning process, the team discussed their experiences with utilizing use cases in a professional setting, which directly correlates with what will be discussed in this paper.

Use Cases in the Industry

Not every team of software developers generate use cases when planning out products and features from stakeholders. While discussing with the team, we found it to be extremely beneficial to identify and define use cases, and all of the team members have experience with utilizing them in their professional careers. Organizations should be encouraged to invest time into defining use cases, particularly in settings that utilize frameworks such as Agile Methodology. The process of defining use cases helps break down a product into manageable features and reduces the likelihood that important elements will be overlooked.

For the overall use cases, the team decided to develop and include multiple tables and work flows that would not only cover all of the use cases but also lead into object oriented modeling and programming. The team made use cases for 5 specific parts to the shop keeper scenario:

- Be able to sort/search vendors by metadata
- When you select vendor, it displays metadata

- Able to log/record the data from the vendor
    - Also add the capability for personal notes, just a simple note section
    - Performance
    - Inactive/active
- Add/Remove vendors
- Handle scheduling and notifications
    - Calendar invites
    - Stock/inventory notifications (if the vendor updates their requirements)
    - Setup automated text reminders for vendor and shopkeep for invites

This also led into the team's work flow design that would help lay out the beginning foundations for object oriented modeling and programming that can develop in future assignments if needed. Once these use cases were split into five areas, the team then expanded on each one, as shown in Figure 1. This snippet of the table from the Whiteboard planning expanded on the Use Case, Actor, Basic Flow, Alternative Flow, and OOP Techniques.

Table 1: Use Case Overview

Use Case	Actor	Basic Flow	Alternative Flow	OOP Techniques
I want to view my vendors in a list through a UI/Web app	Shopkeeper	I login to the website. I navigate to the Vendor page.	I login to the website. I navigate to the Vendor page. I can select a Vendor to view their details	Vendor Object. Contains all metadata, can contain other smaller objects such as representatives
Selector Vendor, and have it display metadata	Shopkeeper	When I click a vendor on the website, I would like all the metadata to be associated with that vendor to be displayed.	I login to the website. I navigate to the Vendor page. I sort alphabetically by metadata fields	Vendor object that will contain all the appropriate metadata and representatives.
Able to log/record data from the vendor	Shopkeeper	I would like to record data from the vendor in notes during interactions.	I login to the website. I navigate to the Vendor page. I have the ability to update notes for the vendor somewhere in the UI.	Vendor Object containing data/ notes
Add./Remove Vendors	Shopkeeper	I want to the ability to add or remove vendors on my website.	This should be either to add, update, or delete a vendor.	Vendor class will have methods associated to adding, update or removing vendors.
Handle Scheduling and Notifications	Shopkeeper	I want to be able to set notifications based on inventory levels. When a vendor hits a certain amount of inventory I will be notified.	I can set this reminder to notify me additionally by email and/or phone.	Vendors contain Inventory. Many different objects can extend Inventory (polymorphism).

Figure 1. Use Case Overview

Assignment Difficulties

The common difficulty the group ran into was primarily the source of communication and utilizing Zoom as the primary tool for brainstorming. Only one member had extensive experience with Zoom and the Whiteboard tools, so most of the team needed to take the time to understand how things worked before the discussion could begin. Now, the team overall feels more comfortable with the tool and the Whiteboards it provides, so it can hopefully be an essential tool for the rest of the course.

Regarding the source of communication for the team, it felt that there were confusions between everyone's preference for contact. The team was unsure if the discussion posts should

be utilized, or if the Canvas Inbox was more beneficial, so coordinating meeting times was difficult. The resolution was to just utilize the direct Inbox messaging, and now there is a steady flow of communication that will hopefully work for the rest of the quarter.

During brainstorming, planning an initial flow of use cases and breaking it down step by step into alternative flows and methods helped significantly, as outlined in Figure 2. This was the first point of planning the team conducted before elaborating through the table provided back in Figure 1.

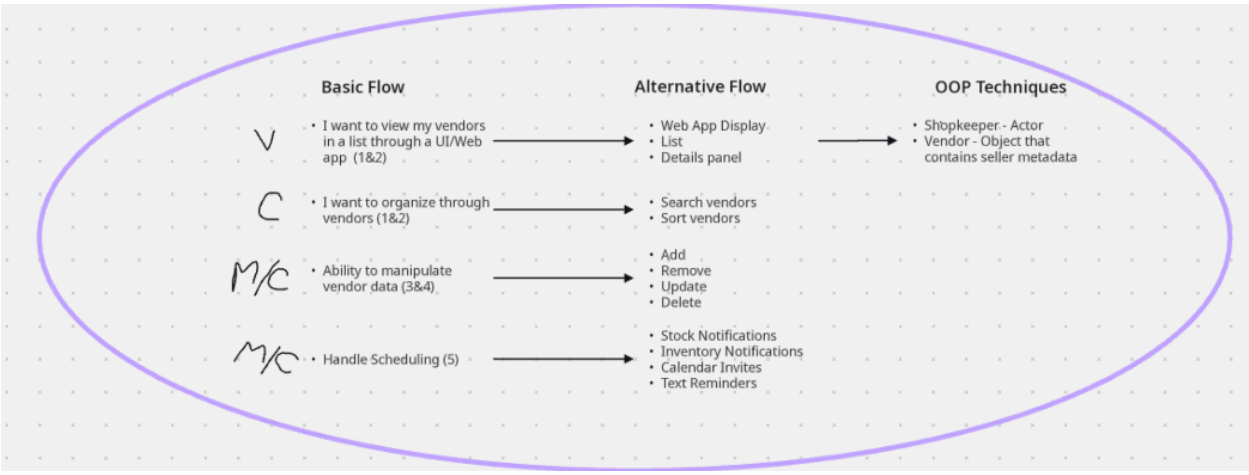

Figure 2. Use Case Breakdown

Once the flow design was complete, tables were utilized to expand on the use cases, and the Whiteboard is provided in Appendix A. This strategy enabled the team to take the scenario step-by-step, and plan together what the use cases could be broken down into, and how OOP methods are applied.

Conclusion

Overall, brainstorming use cases for a shopkeeper's vendor tool provided a comprehensive understanding of how the tool could be developed and implemented. Through team collaboration, it was reinforced that organizations benefit greatly from dedicating time to the planning of use cases prior to initiating product development. Despite minor communication challenges, the team successfully carried out the planning process, demonstrating the effectiveness of structured use case analysis. This exercise highlights how use cases not only guide technical execution but also promote clarity, alignment, and efficiency within a development team, ultimately supporting the delivery of well-designed and reliable software solutions.

Appendix A: Zoom Whiteboard Planning