Assignment 8: Team Write Up

Master of Science:

Information and Communications Technology

Michelle Agustin, Hamdi Ali, Kalika Browder, Jake Collins

University of Denver University College

November 2, 2025

Faculty: Nirav Shah, M.S , MBA

Director: Cathie Wilson, M.S

Dean: Bobbie Kite, PhD

Developing UML Diagrams

Throughout this assignment, our team collaborated on creating several UML diagrams to model the Vendor Interaction and Management software from Scenario 3. Each team member focused on a different type of diagram to practice applying object-oriented design concepts in a real-world context. Working together allowed us to understand how different diagrams connect to show the overall system flow, as well as represent the same scenario in different contexts. By dividing responsibilities and communicating regularly, we were able to combine our individual contributions into a complete and consistent model that reflected the requirements from our original use-case document. We also developed a small domain glossary with terms like Vendor, Representative, Address, Interaction, KPI, and PurchaseOrder. In addition, we created a quick CRUD matrix to clarify ownership and system boundaries. As we created our UML diagrams, these steps helped us stay consistent.

Working on the UML Activity Diagram for our team project was both challenging and rewarding. At first, we found it difficult to grasp exactly how to start the diagram and figure out the appropriate flows and actions between the shopkeeper and the overall system. Once we broke the process into small steps, it simply became easier to understand. We also found that keeping the actions simple with short verb phrases made the diagram clearer. The easiest part was organizing the actions into swimlanes because they helped us see which tasks belonged to the user and which belonged to the system. Visualizing the steps helped us understand how the software would work in real life, as well. Reviewing the original use-case document from our first assignment to make sure the diagram matched the original scenario and details also helped this diagram fall into place.

The activity diagram also illustrates the entire workflow of the scenario. Adding or selecting a vendor, logging interactions, updating responsiveness KPIs, checking inventory, creating or submitting purchase orders, and scheduling reviews are all included. Sequence diagrams depict realistic processes. A communication diagram shows how different system components interact. It shows message exchanges with VendorService, InventoryService, POService, TaskService, Messaging, and the external Vendor.

As sequence diagrams are already a type of interaction diagram, we decided to also create a communication diagram to show message flow and avoid repetition. It was also challenging to find the right level of detail. There was too much detail in the activity diagram, but too little detail missed key decisions. The solution was to keep the activity diagram general and add specific details, such as reorder points and message confirmations, to the sequence diagram. Collaboration and frequent reviews were required to maintain consistency across all diagrams.

We were able to make easier decisions once we had agreed on the key elements and services. The team identified processes such as duplicate-vendor detection, responsiveness thresholds, and inventory reorder rules. We kept things clear and professional by using the same names for actors and services across diagrams. The UML structure also helped us divide work efficiently and stay organized.

As a first step, we brainstormed domain events such as InteractionRecorded, KPIUpdated, FollowUpCreated, ReorderTriggered, POConfirmed, and ReviewScheduled. At least one diagram element was linked to each event. In this way, we were able to ensure consistency and traceability in our work. Separating system functions was also accomplished using the Single Responsibility Principle. An example is VendorService that manages vendor data, InventoryService that manages stock, and POService that manages purchase orders. Follow-ups were handled by TaskService, and communication was handled by Messaging. We were able to make our diagrams more understandable by doing this. To ensure clarity, accuracy, and proper UML notation, we reviewed every diagram before submitting.

The most important lesson was how the diagrams supported each other. Stakeholders can see the overall process through activity diagrams, while developers can visualize system interactions with sequence and communication diagrams. In addition, we learned how important consistency and clarity are in modeling. Each diagram must have the right amount of detail and use the same language. Our teamwork, communication, and technical accuracy were improved by this assignment, which strengthened our understanding of UML.

.

References

Anonymous. 2025. “Sequence Diagrams – Unified Modeling Language (UML).” GeeksforGeeks. https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-sequence-diagrams/

Jain, Ankit. 2025. “Unified Modeling Language (UML) Diagrams – Introduction.” GeeksforGeeks, August 28. https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-introduction/. geeksforgeeks.org

Object Management Group (OMG). 2017. *“What is UML?”* Accessed October 30, 2025. https://www.omg.org/uml/what-is-uml.htm. OMG+1