

Lab 1: Requirements Elicitation & Modeling

This lab focuses on the initial steps of the Software Architecture process: understanding the business needs and modeling the system's external behavior using **Use Case Diagrams**. No complex programming is required in this foundational lab; the primary tools are documentation, modeling software, and analysis techniques.

Objectives

1. Identify and document **Architecturally Significant Requirements (ASRs)**.
2. Define the main **Actors** and **Use Cases** of the ShopSphere system.
3. Model the system context using **UML Use Case Diagrams**.

Technology & Tool Installation

Tool	Purpose	Installation/Setup Guide
Google Docs/Microsoft Word	Documentation of Requirements and ASRs.	Standard word processing software.
draw.io (Diagrams.net)	Creating professional UML Use Case Diagrams.	Access online via a web browser (no install needed) or download the desktop application.
Lucidchart / Visual Paradigm (Optional)	Alternative UML modeling tools.	Register for a free account or install the necessary software.

Activity Practice 1: Requirements Elicitation

Goal: Identify and document key functional, non-functional, and architecturally significant requirements (ASRs).

Step-by-Step Instructions

1. **Identify Actors:** Based on the ShopSphere scenario, list the primary entities that interact with the system.
 - o **Action:** List at least 3 actors. (*Hint: Web Customer, Administrator, Payment Gateway*).

2. **Document Functional Requirements (FRs):** List the specific services the system must provide to the users.
 - **Action:** Document the top 5 FRs related to the **Web Customer** (e.g., browse products, check out).
3. **Document Non-Functional Requirements (NFRs):** Define the quality attributes (how well the system performs, scales, or secures data).
 - **Action:** Document 3 critical NFRs related to **Performance, Security, and Availability** (refer to the initial scenario description for guidance on target metrics).
4. **Identify Architecturally Significant Requirements (ASRs):** Choose 3 requirements (from the FRs or NFRs) that will heavily influence the choice of architecture (e.g., Microservices vs. Monolith).
 - **Action:** For each chosen ASR, write a brief justification explaining *why* it drives architectural decisions.

Example ASR Documentation (ASR Card):

- **ASR:** High Scalability (NFR)
 - **Statement:** The system must be able to handle a surge of **10,000 concurrent active users** during peak sales events.
 - **Impact:** This drives the choice towards **distributed architectures** (Microservices, Event-Driven) and requires **stateless** processing in key services (like Cart/Order lookup) and effective **load balancing**.
-

Activity Practice 2: Use Case Modeling

Goal: Visually model the system's boundary and its interactions with the environment.

Step-by-Step Instructions

1. **Define the System Boundary:** In your chosen modeling tool (draw.io recommended), create a large rectangle labeled "**ShopSphere E-Commerce System.**" This represents the scope of your application.
2. **Place Actors:** Draw stick figures outside the system boundary to represent the primary **Actors** identified in Practice 1.

3. **Draw Top-Level Use Cases:** Inside the system boundary, draw ovals representing the main functionalities (the FRs you identified).
 - **Customer Use Cases:** *Make Purchase, Manage Profile, Search Catalog.*
 - **Admin Use Cases:** *Manage Products, Process Orders.*
4. **Connect Actors to Use Cases:** Draw solid lines connecting each Actor to the Use Cases they participate in.
 - *Example:* The **Web Customer** actor connects to the *Make Purchase* use case.
5. **Detail a Critical Use Case (Checkout):** Focus on the **Make Purchase** use case and apply generalization relationships (include and extend).
 - **Action:** Draw the main *Make Purchase* use case.
 - **Use <<include>>:** Draw the *Secure Payment* use case and link it to *Make Purchase* using an <<include>> relationship. This means Payment **must** happen as part of making a purchase.
 - **Use <<extend>>:** Draw the *Apply Coupon* use case and link it to *Make Purchase* using an <<extend>> relationship. This means applying a coupon is **optional** during the purchase process.

UML Use Case Diagram Example

Documentation & Submission

After completing the activities, organize your work into the following sections:

1. **ASR Documentation:** A table or list documenting the 3 chosen Architecturally Significant Requirements and their architectural impact.
2. **UML Use Case Diagram:** The final diagram created in the modeling tool, clearly showing the system boundary, actors, use cases, and the detailed relationships (include/extend) for the Checkout process.