

# UML Use-case Diagrams

EECS 348 Lab 5 — 2/27/2025

Harlan Williams

Electrical Engineering and Computer Science  
University of Kansas



# What are use-cases for?

- A method of modeling interactions with the system at a high level
- Document *functional* requirements of software
  - ▶ Focus on the broad picture of what people will do with the software, not how it will be implemented
  - ▶ In other words, the “what”, not “how”
- Visualization to communicate the purpose of a software to non-programmers



# Definitions

- **Actors** take on **roles** and use the system to fulfill **use-cases**
  - ▶ An *actor* is someone who uses the system, typically in some *role* depending on who the system is being marketed to
  - ▶ The system then provides services to fulfill *use-cases*
  - ▶ For example, an actor in the *Student* role might use our system to fulfill the use-case *Enroll in a Class*
- **Relations** in the diagram are represented as lines
  - ▶ Contextualize how a role or a use-case relates to other roles/use-cases
  - ▶ A solid line with no arrowhead is an **association**
  - ▶ A solid line with a solid arrowhead is a **generalization**
  - ▶ A dashed line represents either an **extends** or **includes** relation
- Use-cases are placed inside a **System**



# Relations

- An *association* relation between an actor and a use-case implies that the actor has some goal that is fulfilled by that use-case
- A *generalization* takes an existing actor or use case and derives a more specific version from it
  - ▶ The thing being generalized has the arrowhead pointing to it
  - ▶ For example, a *Grad Student* is a more specific version of a *Student*
- An *include* relation means that one use-case includes the functionality of another
  - ▶ The use-case being included has the arrowhead pointing to it
  - ▶ To achieve one use-case, the use-case being included must also happen
- An *extends* relation means that one use-case adds functionality to another
  - ▶ The use-case having functionality added to it has the arrowhead pointing to it



# Identifying use-cases

- Think from the perspective of the user—what do they want to get done?
- Think from a high-level—what should the system offer?
- Think about what other entities the system interacts with—payment processors, *etc.*



## Assignment 3 hint

Assignment 3 asks you to write informal and formal requirements. One good type of informal requirement to use would be a user story:

### User story

As a ...,  
I want to ...,  
So that ...



For example, if we had a blog with a newsletter, two user stories could be:

*As a blog visitor, I want to subscribe to the newsletter, so that I can receive email updates when new content is published on the blog.*

*As the blog author, I want to send an email to all my subscribers, so that I can notify them when new content is published.*

Source: *Zero to Production in Rust* by Luca Palmieri



# Lost? See...

- <https://geeksforgeeks.org/use-case-diagram/>
- <https://online.visual-paradigm.com/>

