

**University Parking System Use Cases for**  
Master of Science in Information Technology  
Software Design and Programming

Estell Moore

University of Denver College of Professional Studies

9/20/2025

Faculty: Nirav Shah, M.S.

Director: Cathie Wilson, M.S.

Dean: Michael J. McGuire, MLS

### **Abstract**

This paper outlines the reflection of the overall experience developing use cases for a University Parking System. Challenges when designing are highlighted, as well as outward hindsight perspective of what should have been known and was instead learned. Each component of the assignment is reflected on, touching on implementation decisions and how the use case system was overall defined.

## CONTENTS

1. Introduction	3
2. Reflection	3
The Diagram	
Use Case Elements Table	
3. Conclusion	4

## **Introduction**

When developing use cases, there are certain challenges to face when thinking of an overarching perspective of a system. Use cases are meant to be broad so they can later be broken down into smaller elements in design, but with this experience, I managed to do the opposite and define use cases based on unorganized ideas for the system. Though backwards, it was still effective.

## **Reflection**

### **The Diagram**

Creating a use case for the university parking system was relatively straightforward, but when generating the use case diagram, there were some challenges when deciding what to include and what not to include. Approaching the process of designing a system with many nuances, external factors, and software requirements could be made easier if all thoughts could be communicated prior to creating concrete use cases; the thoughts could be organized under common categories (use cases), which then actors can be identified to accompany them. Though it seems backwards, this is the overall thinking process that led to the design of the diagram.

During the creation of the diagram, identifying the actors came from the perspective of who is using the system, along with what other systems are affected by it. Something I wish I knew before defining use cases regards the definition of an actor. Recognizing inanimate entities as actors was initially unintuitive, but reframing the parking lot gate as a collaborating software component that depends on permit-validation results clarified its role. When planning the use

cases, it appeared important to keep the overarching ideas broad and simple. These could later be broken down with alternate use cases, but it was important to encapsulate the basic use case functionality that could later be expanded upon in the elements table. Use case diagrams are supposed to be informative, yet simple and easy to read. Initially, I have a habit of expanding too much on one subject, and that was presented as a challenge when deciding what to include in the diagram itself.

### Use Case Elements Table

Filling in use case information for the elements table was easier to generate than the use case diagram, since it requires full in-depth expansion on a single use case. The original thinking process that was utilized to create the diagram was utilized directly in the development of the table, and expanded on. From this experience, it seems this table benefits from extra data and information, so there was no necessity for wondering on whether or not to include or restructure something. With this in mind, there were some complications when figuring out constraints and assumptions. When applied to the use case chosen, it was difficult to think of anything applicable for these fields. Considering external inclusions of other use cases and actors ultimately clarified what to include in those sections, though the process was initially more challenging than expected.

### Conclusion

Given prior experience in professional software engineering, the nuances and expected code structure made the elements table simpler to complete than the diagram's overarching framework. In hindsight, I wish I understood how to create use cases from a generic and simple

standpoint instead of backwards engineering with organizing by category. Although it sounds more difficult, starting by laying out unstructured ideas, then later organizing them is still an effective way to identify use cases.