**Parking System Classes**

ICT 4305: Object Oriented Methods and Programming

Michelle Agustin

University of Denver College of Professional Studies

October 5, 2025

Faculty: Nirav Shah, M.S , MBA

Director: Cathie Wilson, M.S.

Dean: Michael J. McGuire, MLS

Creating the sequence diagrams for the parking system was both challenging and insightful. Understanding the logical flow of actions such as a customer registering, adding a car, and receiving a permit, was straightforward, but representing those interactions accurately in UML required careful thought. The hardest part was deciding which classes to include and ensuring that each message and activation bar reflected the real flow of the system.

Reviewing UML references, especially Fowler’s UML Distilled (2004), and examples from class materials helped clarify the distinction between the UI, control objects like ParkingOffice, and the entity classes such as Customer and Car. Sketching the diagrams before building them digitally made it easier to visualize message order and avoid unnecessary complexity. Having the use case from the previous assignment also gave clear foundation for designing the message flow.

One thing I wish I understood earlier was how critical message sequencing and activation are in showing system behavior. At first, I treated the diagram more like static structure rather than a dynamic interaction. Realizing that each message represents an actual system call helped me refine the design and improve consistency between the diagrams and updated class model.

For implementation, I chose to include the UI as a boundary object to show where user input begins. I also represented object creation explicitly, such as <<create>> Customer and <<create>> Car, to make the sequence of events clearer. When updating the classes, I ensured methods like addCharge() and register() aligned with the sequence diagrams.

This assignment deepened my understanding of how sequence diagrams bridge use cases and code structure, and how each interaction supports object-oriented design.

**References**

Fowler, Martin. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3rd ed. Boston: Addison-Wesley, 2004.