**Use Cases for a Parking System**

ICT 4305: Object Oriented Methods and Programming

Michelle Agustin

University of Denver College of Professional Studies

September 21, 2025

Faculty: Nirav Shah, M.S , MBA

Director: Cathie Wilson, M.S.

Dean: Michael J. McGuire, MLS

PART 2: USE CASE ELEMENTS

|  |  |
| --- | --- |
| **Field** | **Description** |
| ID | UC-03 |
| Name | Purchase Permit |
| Short Description | Customer purchases one or more parking permits (for a specific lot or campus-wide). The system gathers vehicle information if needed, calculates price (including any compact-car discounts or lot-specific rules), processes payment, and issue the permits. |
| Goal | Allow a registered customer to obtain valid parking permits and complete payment, so they can legally park in target lots. |
| Preconditions | Customer has an active account (or can register during purchase). At least one registered vehicle exists, or the customer is prepared to register a vehicle during checkout. Payment method available. |
| Success End Condition | Permits created, linked to customer (and vehicle if applicable), payment successfully processed, permit status Active (or scheduled to start at configured start date). Receipt generated. |
| Fail End Condition | Payment fails and permits not issued, or validation detects policy violation (e.g., limit exceeded), purchase aborted, and no permit created. Customer informed of failure and next steps. |
| Stakeholder | Primary: Customer  Secondary: Payment Gateway, Parking Office (admin for approvals or special permits), Gate/Hardware (downstream consumer of permit) |
| Trigger | Customer selects “Purchase Permit” from UI or mobile app (often after selecting lot and permit type) |
| Normal Flow (Primary) | 1. Customer opens Purchase Permit UI and selects lots/permit type and start/end dates 2. System displays pricing, permit rules (entry-only or entry and exit), taxes/fees, and discount eligibility (compact-car). 3. Customer selects target vehicle or chooses to register a new vehicle during checkout 4. System validates vehicle information and applies any discounts (e.g., compact-car discount if vehicle type qualifies). 5. System calculates total price and displays summary. 6. Customer confirms purchase and provides payment details (or uses store method). 7. System sends transaction to Payment Gateway and awaits confirmation. 8. On payment success, system issue permit, links them to the customer/vehicle, updates permit status to Active (or Pending if admin approval required), and sends receipt. |
| Alternative Flows | A1 – *Vehicle not registered*: During step 3, the customer chooses to add vehicle details, system validates and continues.  A2 – *Payment declined:* System informs customer, offers retry or alternate payment method. If unresolved, purchase is canceled or no permit issued.  A3 – *Administrative approval required:* System places permit in Pending state and notifies customer, admin later approves or rejects. |
| Includes | UC-02 (Register Vehicle) – included if customer registers vehicle during purchase.  UC-09 (Charge Account / Payment) – for processing payment. |
| Frequency of Use | High – depends on term cycles. For example, many customers at the start of the semester. Daily for short-term or visitor permits. |
| Constraints & Special Requirements | * Must support multiple permit types such as daily, monthly, term, visitor * Compact-car discounts must be applied when vehicle type is compact * PCI (Payment Card Industry)-compliant payment integration or campus billing interface required * Support for immediate issuance or scheduled start date * Audit logs for transactions and permit issuance |
| Assumptions | * Customer information is valid, sand address and payment information is current * The Payment Gateway has sufficient uptime and a defined API * Vehicle classification (compact vs. standard) is determined from user input or a maintained list |
| Notes and Issues | * Define business rule: when is admin approval required? (e.g., special permits or capacity limits) * Need refund and cancellation policy and UI flow for refunds * Consider concurrency control for limited permit quotas per lot * Owner: Parking Office – for policy decisions * Due date / next action: confirm discount rules and approval thresholds with stakeholders |

**References**

Cockburn, Alistair. 2001. *Writing Effective Use Cases*. Boston: Addison-Wesley.

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

Kulak, Daryl, and Eamonn Guiney. 2012. *Use Cases: Requirements in Context*. 2nd ed. Boston: Addison-Wesley.

Object Management Group (OMG). 2017. *Unified Modeling Language (UML) Version 2.5.1*. Needham, MA: Object Management Group. https://www.omg.org/spec/UML/2.5.1/.