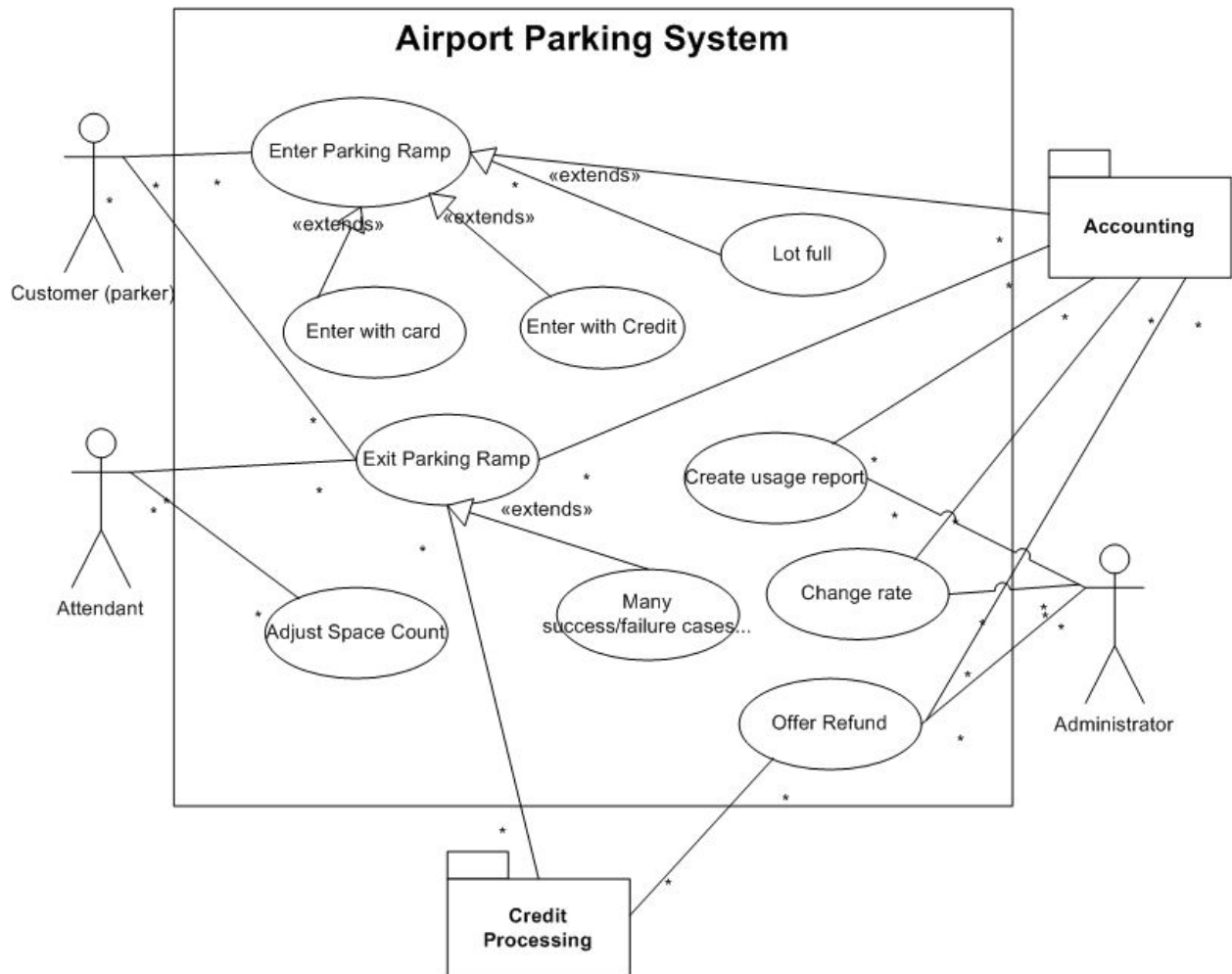# CSCE 742 - Functional View Activity
Name(s):

You have been asked to develop a new automated parking system at the CAE airport.

In this new system, a user can simply insert their credit or debit card into the card reader at the ramp entrance. This will record the time they entered airport parking. They then can use the same credit or debit card to pay at an exit lane. The system should be fully automated; there is no waiting in line for a cashier. The system should also support ticketed parking: where the user receives a ticket and pays either by credit card or cash on exiting.

You have been provided with the following Use Cases for the system

Airport Parking System

Enter Parking Ramp

«extends»

Lot full

«extends» «extends»

Customer (parker)

Enter with card

Enter with Credit

Accounting

Exit Parking Ramp

Create usage report

«extends»

Attendant

Adjust Space Count

Many success/failure cases...

Change rate

Offer Refund

Administrator

Credit Processing

General design principles:

- Encapsulate components likely to change
    - Hardware, policies (pricing, lot capacity)
- Define elements that individually and collectively have value
    - High coherence, low coupling

What do you need to track?

- On entry?
- On exit?
    - Where do you store completed transactions? In the system? Sent through interface to accounting system?
- For pricing?
- When performing manual overrides
    - Who can perform them?
    - How do you log them?

What do you need to control?

- Physical gates for entry / exit
- Entry kiosks
    - Credit card reader
    - Parking card dispenser
- Exit kiosks
    - Automated: credit card / parking card reader
        - Optional: cash input
    - Attendant kiosks
        - Point of sale device: in or out of system?
        - Allow manual override of charges?

How do you want to report?

- Revenue?
    - Partitioned by pricing type?
    - Current? Over time?
- Card stock levels per entry kiosk?
- Mechanical failures?
- Ramp usage? Utilization over time?

From the requirements, use cases, and other provided information:

- **Derive elements.**
  - **For each, briefly note the responsibilities and purpose of that element.**
  - **You do not need to design interfaces, but think about how you would implement important ones.**
- **Draw a UML Context Diagram depicting the system.**