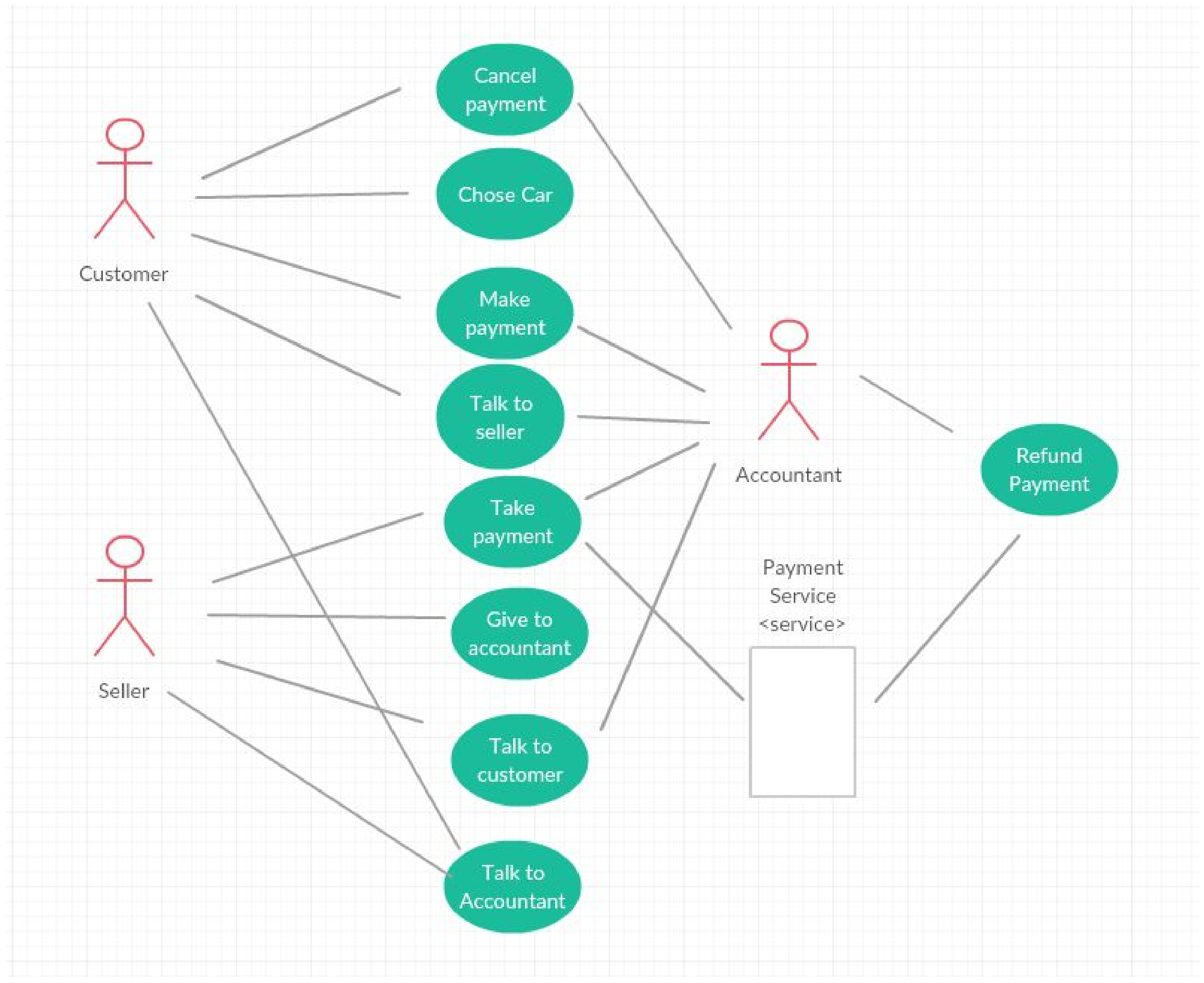
Michael Fernando Mendez Jimenez and Troy Simpkins
Lab 04z

UML Use Case:



The main use cases are the customer, seller, and accountant. The customer will have access to talk to the seller, chose the car, make a payment which will connect them them to accountant, seller, or the payment service. They can cancel payment. They also have access to the checkout service and access to all the cars in stock. Seller will be able to talk to the customer, give the payment they received from the customer to the accountant. The accountant will have access to talk to the seller, cancel payment, make payments for the customer, take payment, talk to the customer and give refund.

Michael Fernando Mendez Jimenez and Troy Simpkins
Lab 04z

ER Diagram

3t

Car
Manufacturer
Model
Car ID
Stock
Transmission
Fuel Efficiency
Price

Seller
Seller ID
Name
Phone
Address
Cust ID

Accountant
Name
Order #
Acct ID

Payment Service
Pay ID

Michael Fernando Mendez Jimenez and Troy Simpkins
Lab 04z

Payment Type
Company

Checkout Service
Checkout ID
Total Cost
Payment Type

Customer
Cust ID
Name
Phone
Order #
Address

Refund
Accountant(Order #)
Payment Service(Pay ID)

Take Payment
Seller(Seller ID)
Customer(Cust ID)

Our Brainstorming at Beginning

Actors:

Customer should talk to seller to initiate the buying process. Seller (better word for this) should talk to accounting. Accounting deals with the payment service.

This allows accounting <> Payment Service to have a many:many relationship I think??

Customer (name, id, phone, address,order number)
- talk to seller
- choose car
- checkout - includes payment - payment -> payment company
  - cancel payment -> refund payment

Seller (name, id, phone, address, Cust ID)
- take payment info
- give to accountant
- talk to customer

Accountant(name, id,order number) - could also make it accounting??
- Create transaction / payment
- Cancel transaction / payment

Payment Service (company, payment Type)
- process payment info
- verify
- send confirmation
- refund payment

Supplier (name, id) - We don't need a supplier name since we have the manufacturer in cars
- supplys cars - many:one relationship

Cars (Manufacturer, Model, Price, Fuel Efficiency, Stock, Transmission)
- Buy - do we need sell car as well?
- Test-drive?
- Show/Look at????

Checkout Service (ID, payment Type, Total Cost)
- Customer -> Checkout
- Seller -> Checkout
- Help extends checkout
- Checkout includes payment?

Payment -> Payment Service

Relationships:

Cars <> Customer - 1:many

Payment Service <> Checkout? - Many:many if we can have this relationship

Payment <> Accounting/Accountant - Many:Many or 1:many depending on if accounting or a single accountant in accounting

Customer <> Seller - 1:many

Customer<>Accountant - Many:Many Can pay the accountant, (accountant can take payments from any customer, customer can make a payment through any accountant)