Business Activity Model

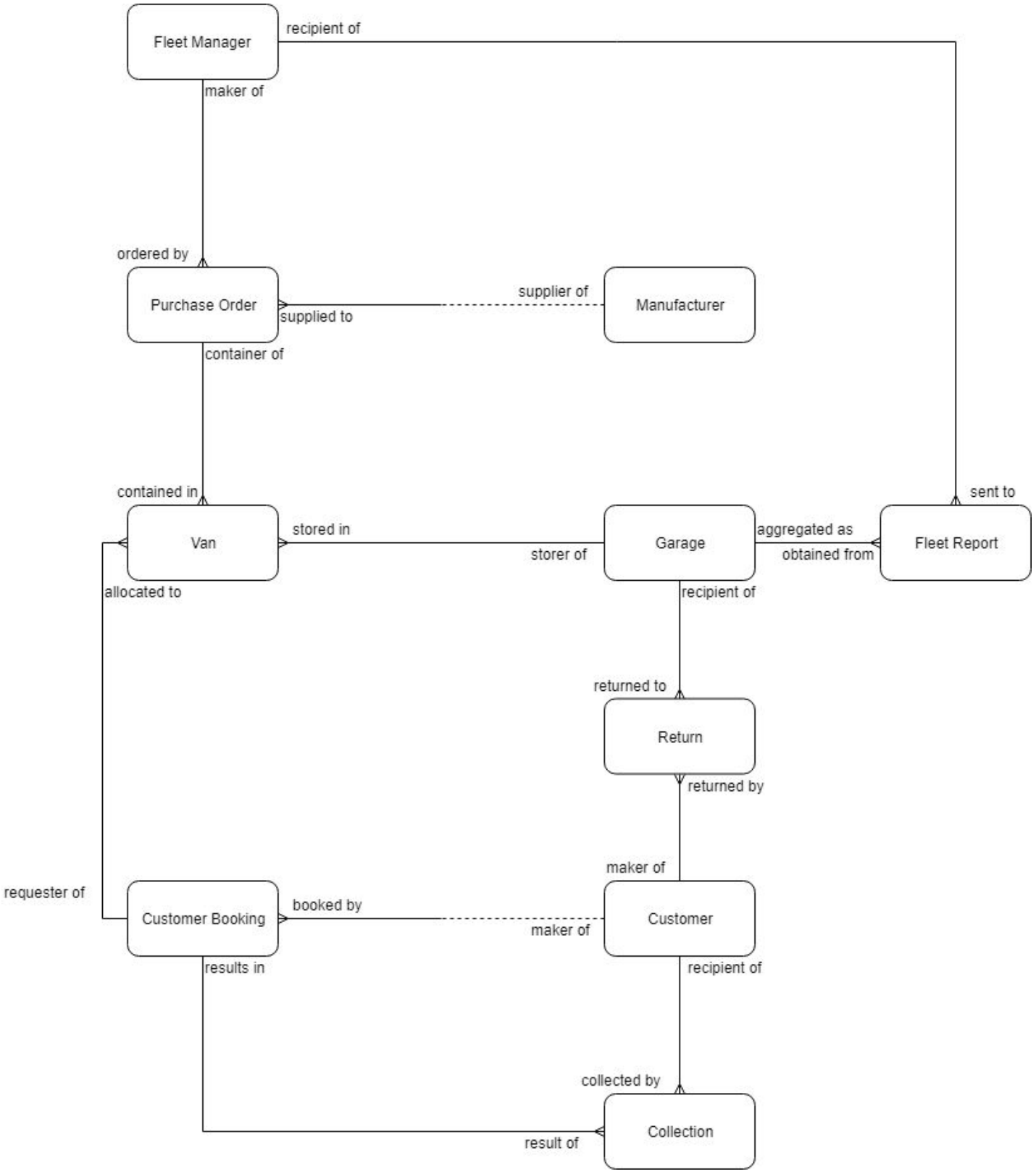
As in the BAM above, I assumed that the deposit is separate from the actual payment, and that the customer would receive their deposit back, providing they followed the terms of the contract, i.e. they show up to collect van and return van on time. The deposit acts as an incentive for the customer to respect the conditions of the hire.

Logical Data Model

LDR Entity Relationship Matrix

	Manufacturer	Purchase Order	Van	Garage	Customer Booking	Customer	Collection	Return	Fleet Manager	Fleet Report
Fleet Report				X					X	
Fleet Manager		X								
Return				X		X				
Collection					X	X				
Customer					X					
Customer Booking			X							
Garage			X							
Van		X								
Purchase Order	X									
Manufacturer										

LDR

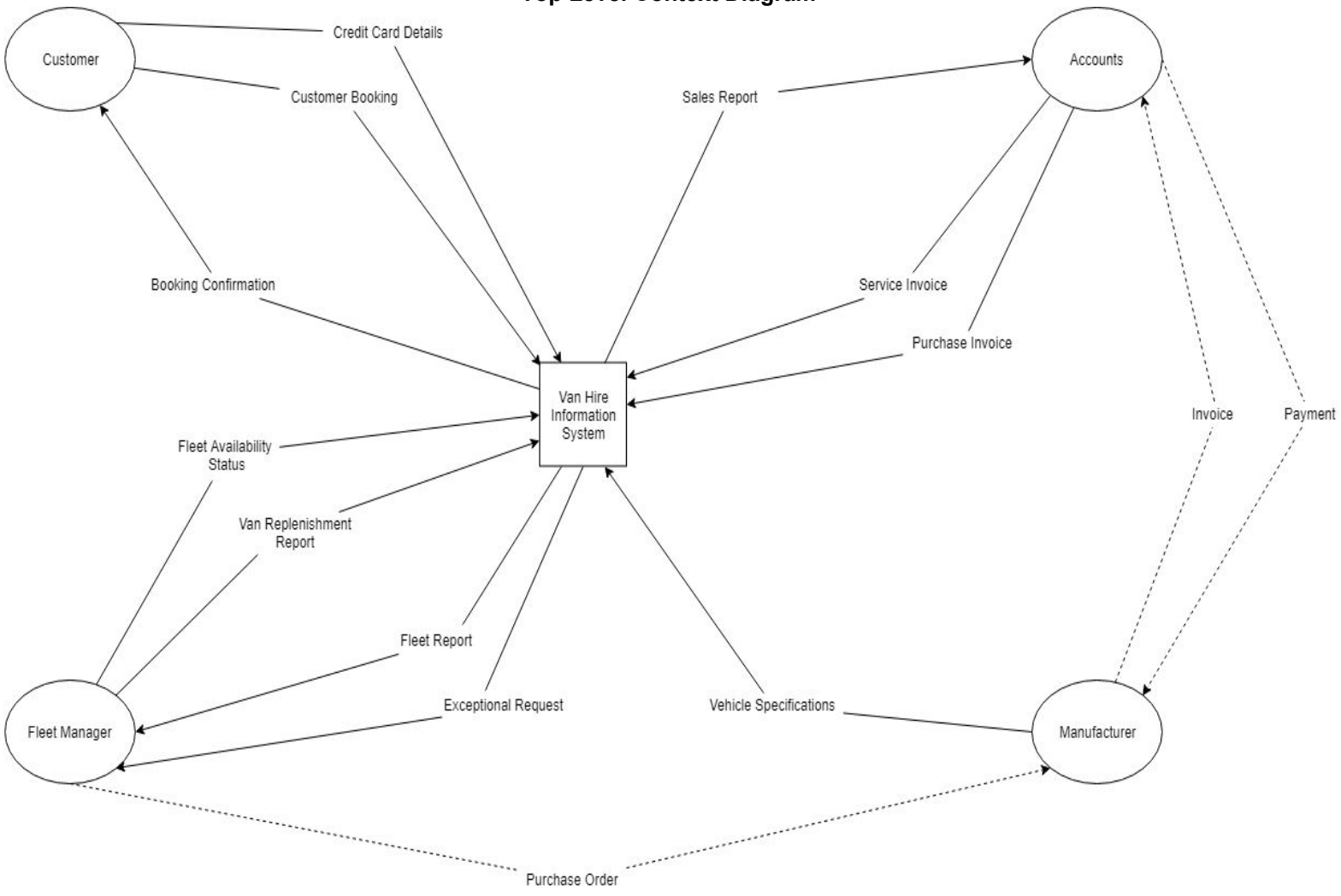

Fleet Manager
recipient of
maker of
ordered by
Purchase Order
supplier of
supplied to
Manufacturer
container of
contained in
sent to
Van
stored in
Garage
aggregated as
obtained from
Fleet Report
storer of
allocated to
recipient of
returned to
Return
returned by
maker of
requester of
Customer Booking
booked by
maker of
Customer
results in
recipient of
collected by
Collection
result of

Top Level Context Diagram

External Entities Context Table

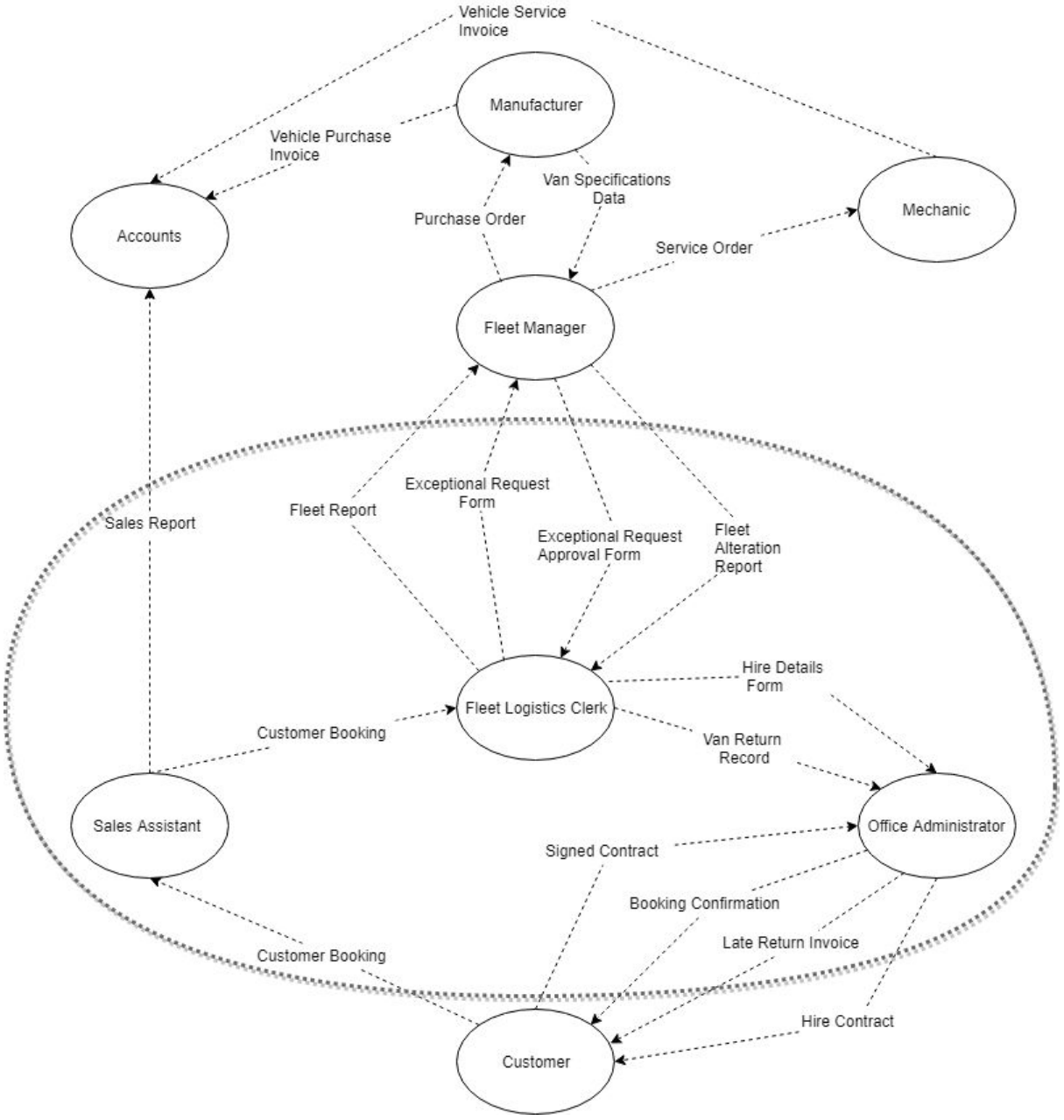
External Entity	Source or Recipient	Data Flow
Manufacturer	S	Vehicle Specifications
Accounts	S	Purchase Invoice
	S	Service Invoice
	R	Sales Report
Fleet Manager	S	Fleet Availability Status
	R	Fleet Report
	S	Van Replenishment Report
	R	Exceptional Request
Customer	S	Credit Card Details
	S	Customer Booking
	R	Booking Confirmation

Top Level Context Diagram



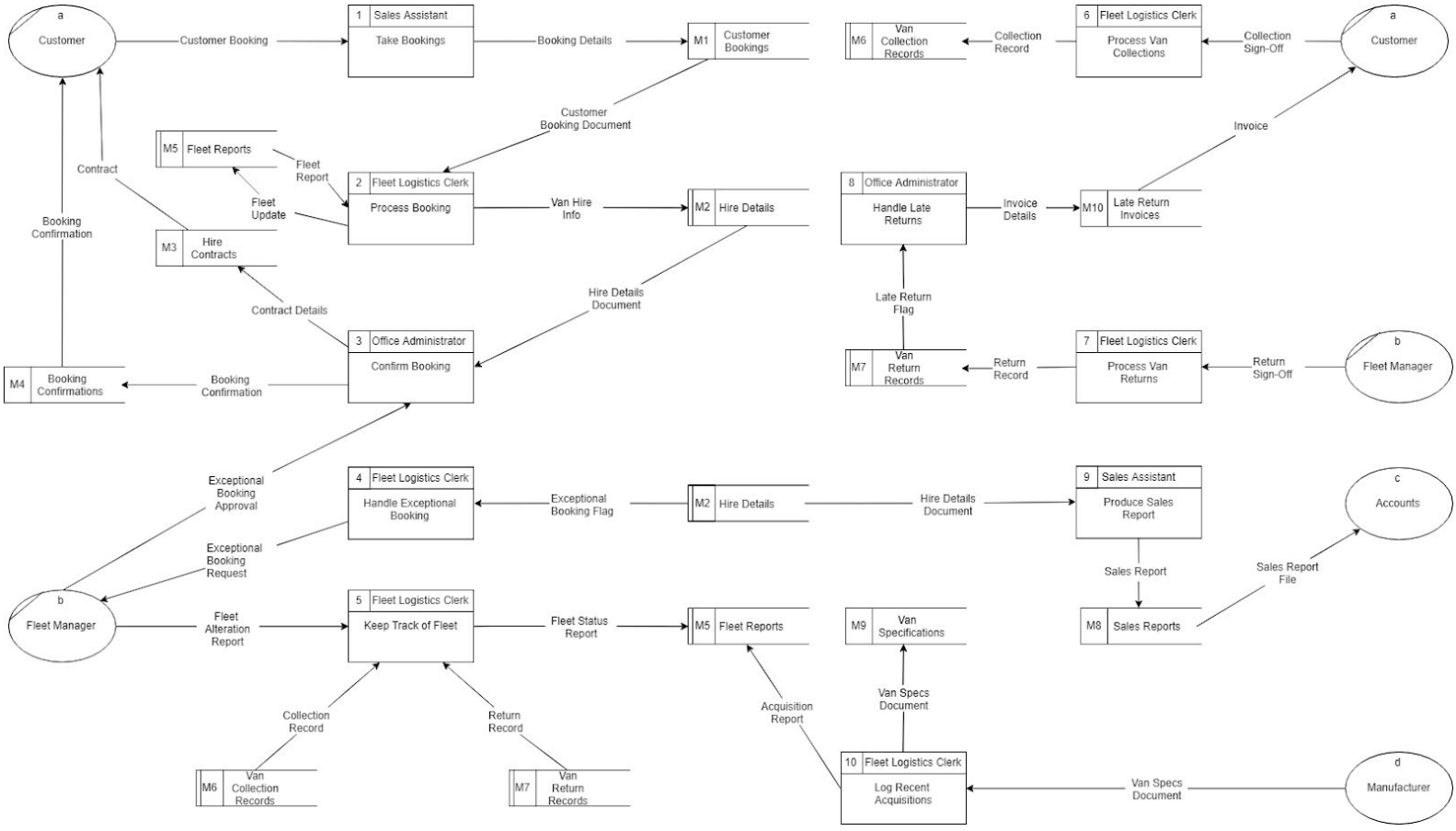
In this diagram and subsequent diagrams, I will define an ‘Exceptional Request/Booking’ as a booking that may push the limits of what the company allows in a hire, such as booking for the maximum hire period, or multiple vans.

Document Flow Diagram



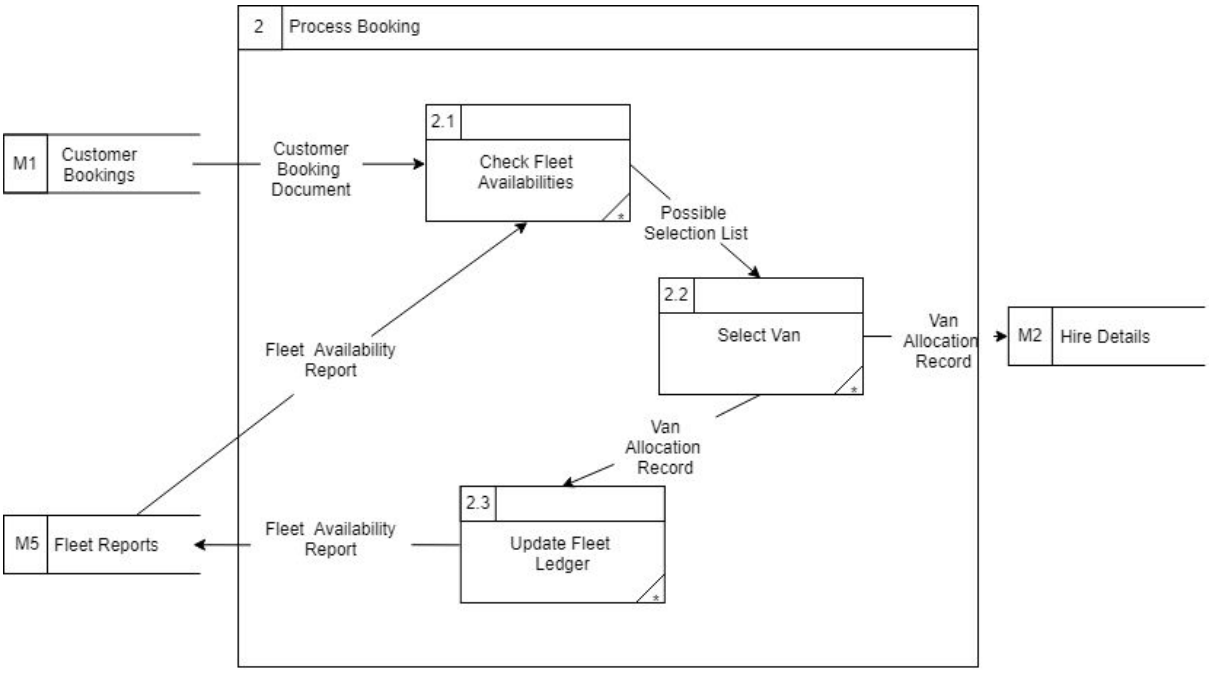
I decided that the fleet manager should be outside the boundaries of the system, as the decisions (e.g. ordering new vans) made by the fleet manager have a significantly greater weight than the fleet logistics clerk and hence they should not be left to automation.

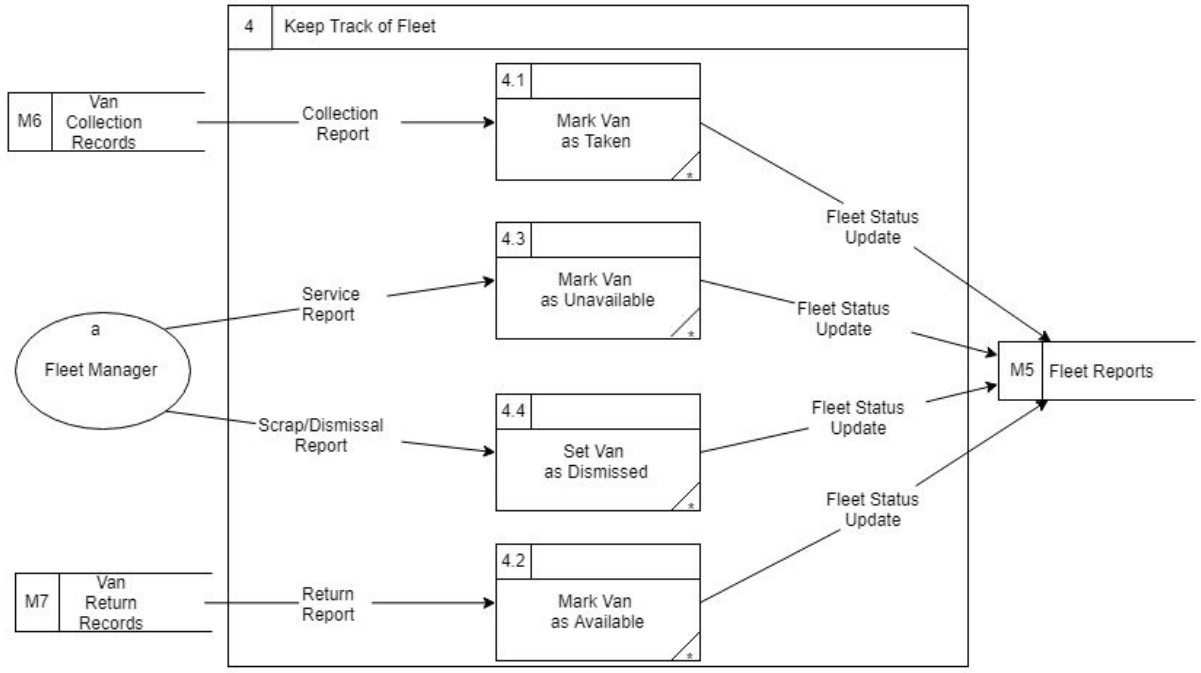
Current Physical (Level 1) DFD



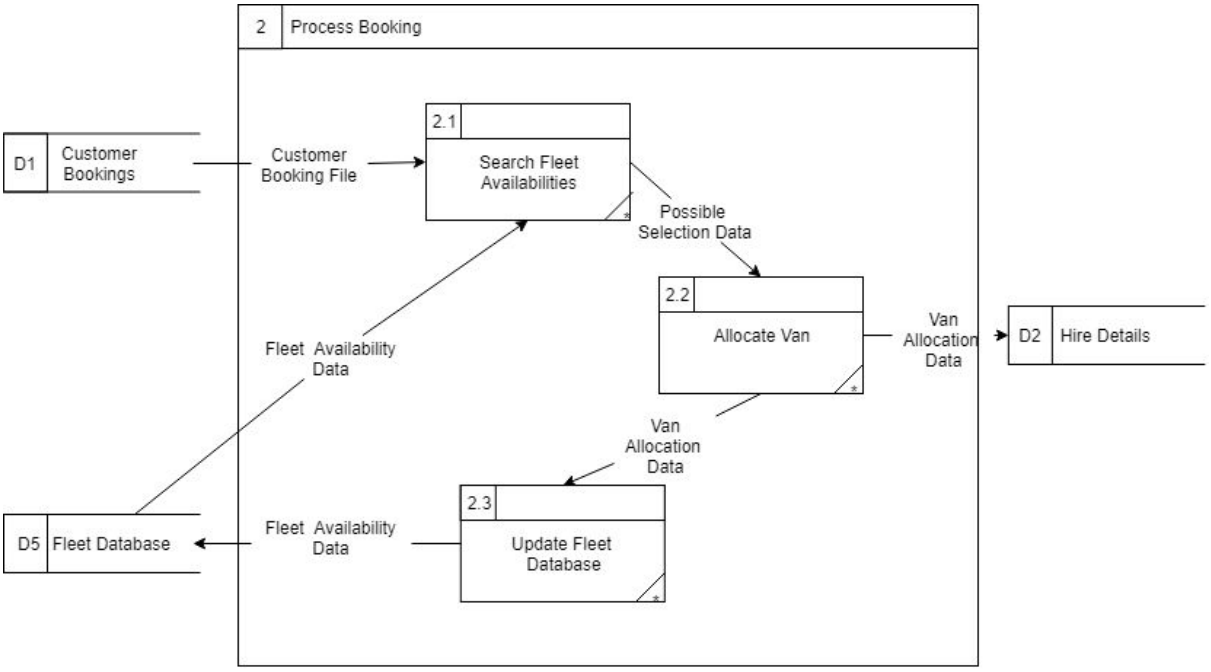
Note: You may have to zoom to properly see the contents of this DFD.

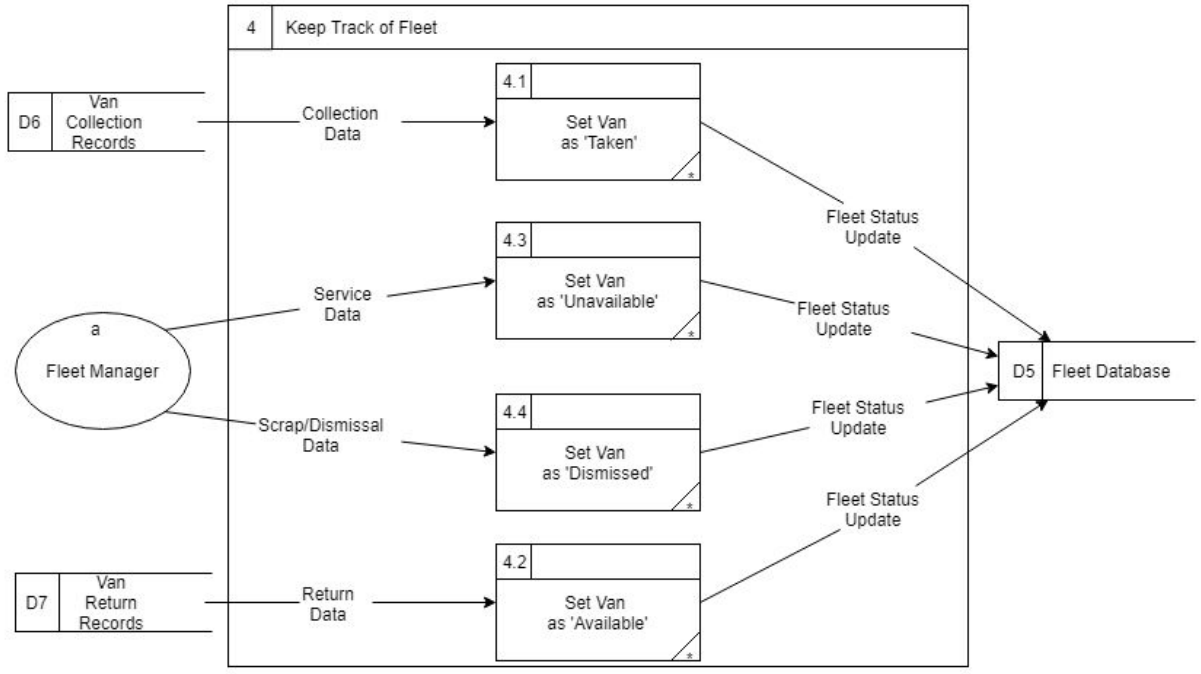
Physical (Level 2) DFDs





Logical (Level 2) DFDs





Function 1 - Search Fleet Availabilities

(Logical Level 2 DFD)

```
FUNCTION SearchAvailabilities

    ORDER fleet BY most recently returned

    FOR EVERY van IN fleet WHERE van type IS requested type
        GET van status
        IF van status IS 'available'
            PUSH van TO stack

    END
```

Note: The fleet was ordered by most recently returned, and then pushed to a stack. This is to ensure that all vans are used equally so as not to wear a select few vans out unnecessarily; the least recently returned van would be pushed to the top of the stack, and hence it would be the next one to be allocated.

Function 2 - Confirm Booking

(Physical Level 1 DFD)

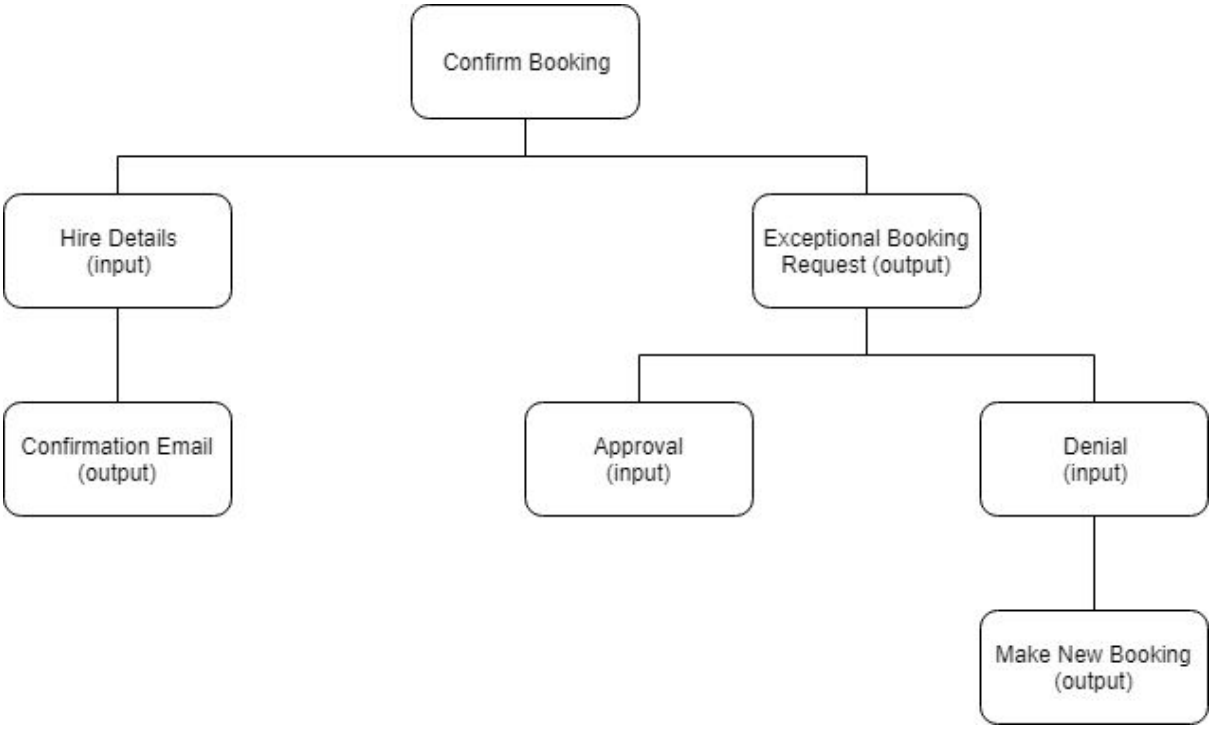
```
FUNCTION ConfirmBooking

    GET customer, hire info FROM hire details

        IF exceptionalRequestFlag NOT TRUE
            GENERATE confirmation email
            SEND confirmation email TO customer address
        ELSE
            SEND  customer, hire info TO fleet manager address
            IF response IS 'approved'
                GENERATE confirmation email
                SEND confirmation email TO customer address
            ELSE
                GENERATE make new booking email
                SEND make new booking email TO customer address

    END
```

I/O Diagram



Process Description 1 - Log Recent Acquisitions

IF van IS new
    MOVE van TO fleet

    GET van specs FROM manufacturer
    ADD van specs TO van details
    SET van status AS ‘available’

END

Process Description 2 - Process Van Returns

IF van status IS ‘taken’ AND return sign-off RECEIVED
    SET van status AS ‘available’
    IF lateReturnFlag NOT TRUE
        RETURN deposit TO customer
        SEND thank you email TO customer address
    ELSE
        SEND invoice TO customer address
        IF no response

DEDUCT late fee FROM deposit

END

Data Flow

From	To	Data Flow	Data Content	Comments
M1	2	Customer Booking File	Customer I.D. Date of Collection Period of Hire Van Type	The raw booking data is sent to be processed. We only need the customer I.D. and booking details to continue
2	M2	Van Hire File	Van I.D. Customer I.D. Date of Collection Period of Hire	When the booking is processed, a van is allocated to the booking hence why Van Type has been replaced by Van I.D.

Entity Description

Entity Description				
Entity Name	Collection			
Description	The collection of a hired van by a customer			
Attribute		**Primary Key**	**Foreign Key**	**Mandatory/ Optional**
Collection Number		Yes		M
Collection Date				M
Hire Period				M
Booking Number			Yes	M
Customer Number			Yes	M
must/may be	**either/or**	**Link Phrase**	**only one/ one or more**	**Entity Name**
must be		collected by	only one	Customer
must be		the result of	only one	Customer Booking
User			**Access**	
Customer			Read	
Logistics Clerk			Read, Write, Create	
Fleet Manager			Read, Write, Create, Delete	

Data Store

Data Store	
Data Store I.D.	D5
Data Store Name	Fleet Database
Data Store Description	A database containing all fleet related information. It would contain the status of every van (is it available? is it taken?), the specifications, and the records of every van (how long has it been in the fleet? How many times has it been used? Mileage?), along with other information.