Declaration on Plagiarism and Assignment Submission Form

This form must be filled in and completed by the student(s) submitting an assignment

Student Name(s):	Tomas Baltrunas
Student Number(s):	17350793
Programme:	BSc. Computer Applications
Assignment Title:	Systems Analysis SSADM Project
Module Code:	CA214
Module Coordinator:	Renaat Verbruggen
Submission Date:	2018/12/10

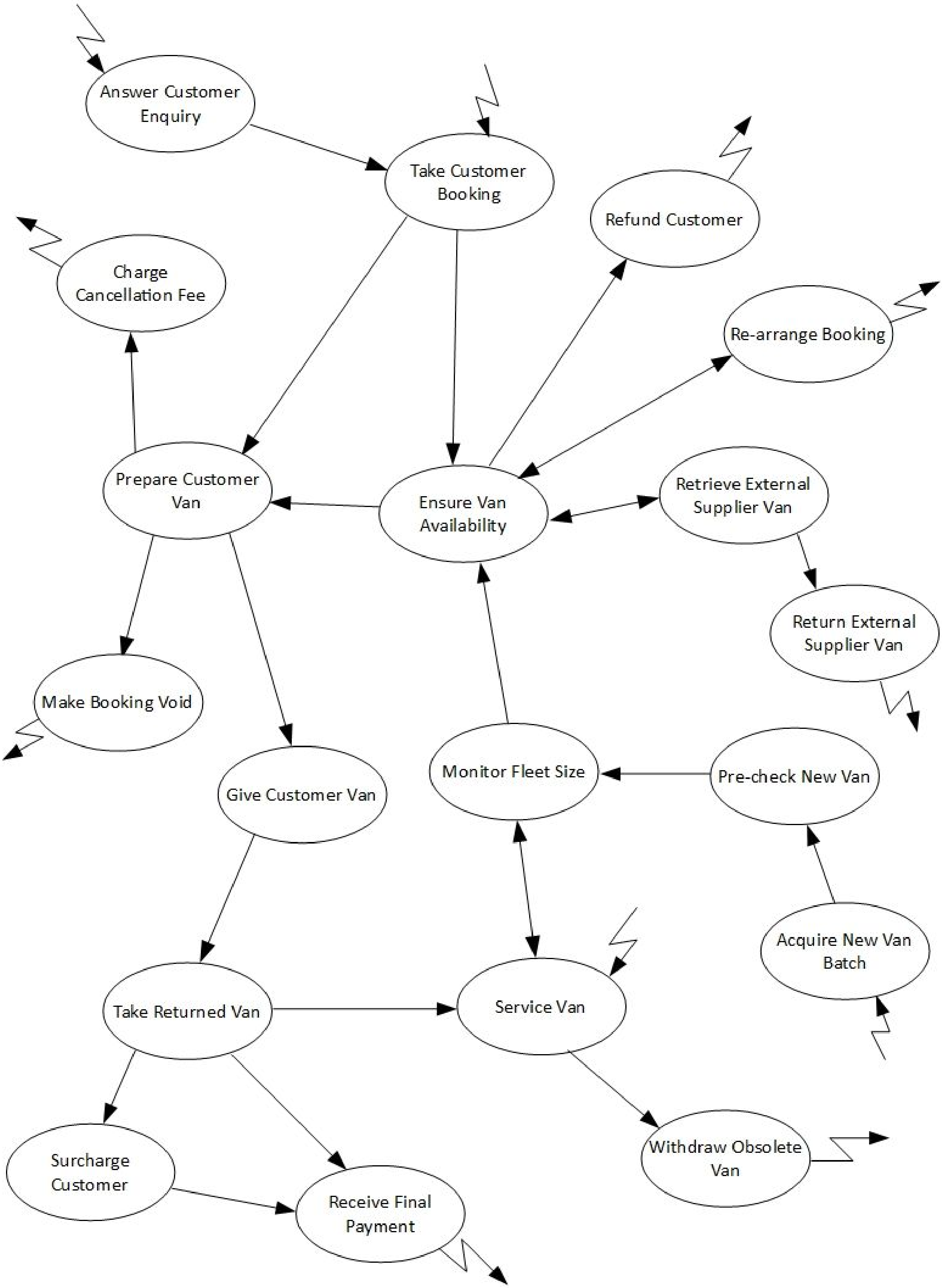
I/We declare that this material, which I/We now submit for assessment, is entirely my/our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my/our work. I/We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I/We have read and understood the Assignment Regulations. I/We have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me/us or any other person for assessment on this or any other course of study.

I/We have read and understood the referencing guidelines found at
http://www.dcu.ie/info/regulations/plagiarism.shtml,
https://www4.dcu.ie/students/az/plagiarism
and/or recommended in the assignment guidelines

Name(s): ________Tomas Baltrunas Date: ________2018/12/07

Take the systems description in the case study and produce:

A Business Activity Model (Give it at the Task level approx 15 tasks)

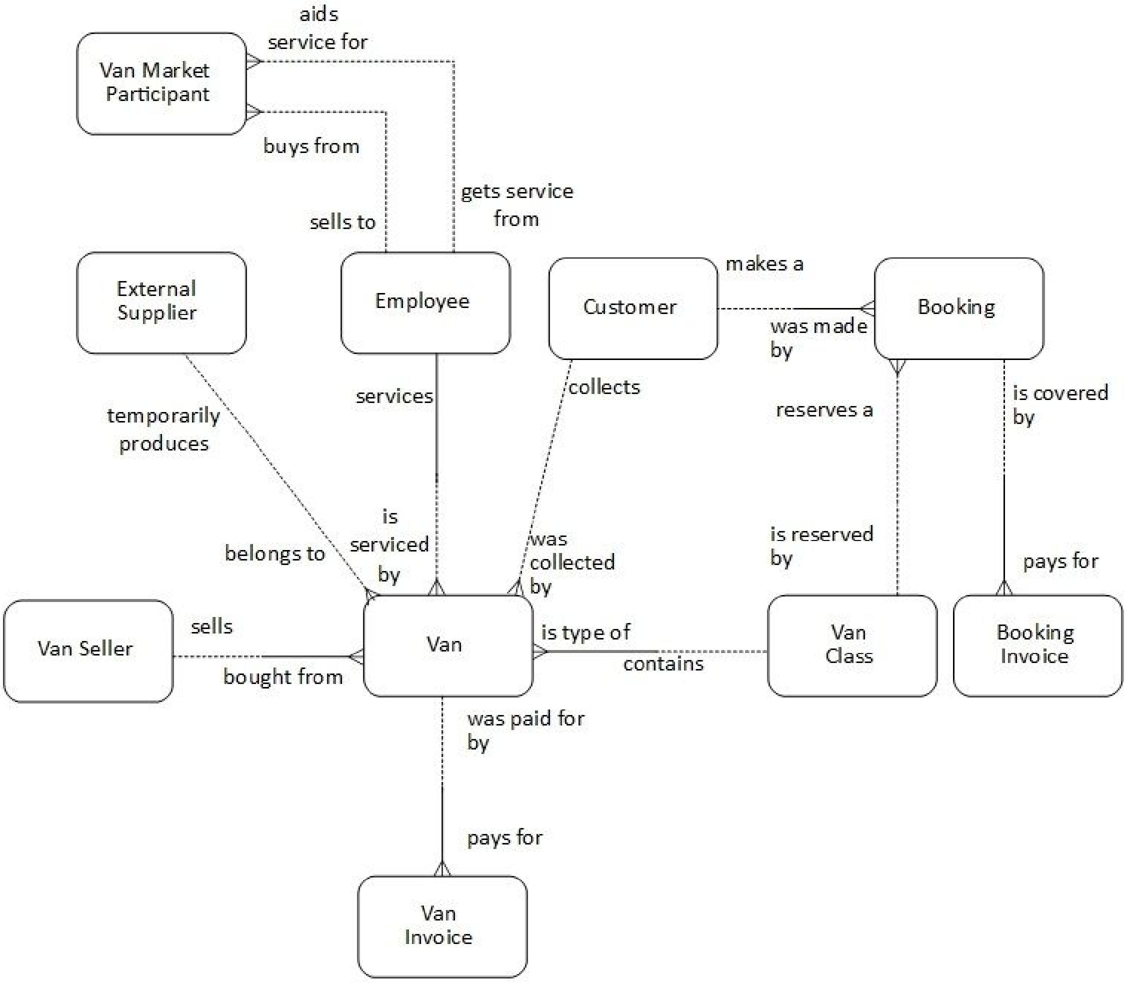

Answer Customer Enquiry
Take Customer Booking
Refund Customer
Charge Cancellation Fee
Re-arrange Booking
Prepare Customer Van
Ensure Van Availability
Retrieve External Supplier Van
Return External Supplier Van
Make Booking Void
Give Customer Van
Monitor Fleet Size
Pre-check New Van
Acquire New Van Batch
Take Returned Van
Service Van
Surcharge Customer
Receive Final Payment
Withdraw Obsolete Van

A Logical Data model. (Entities and relationships - be careful on: naming both sides; optionality; degree and cardinality)

Logical Data Structure (LDS)

The following is the *'Current Environment'* LDS.

To get 1:m relationships I assume that employees are 'assigned' to vans and van market participants, so an occurrence of a van is serviced by exactly one assigned employee (one mechanic), and a van market participant deals with exactly one assigned employee (a 'representative').



Logical Data Key

I assume that a single van seller provides exactly one same warranty for its vans.

VAN
License Plate Number
*Van Class
*Assigned Employee
Status
*Collected by Customer
Collection Time
Return Time
*External Supplier
Insurance Number
NCT Number
Manufacturer
Make
Model
Engine Power
Fuel System
Last Service Point
*Van Seller
Date of Purchase
Comment

CUSTOMER
Customer Number
Full Name
Phone Number
Email
Home Address

VAN CLASS
Van Class Name
Description

BOOKING
Booking Number
*Customer Number
*Van Class Reserved
Hire Start Date
Hire End Date
Status
Comment
Last Modified

EXTERNAL SUPPLIER
Supplier Number
Supplier Business Name
Supplier Phone Number
Supplier Email
Supplier Premises Address

EMPLOYEE
Employee Number
Full Name
Role
Phone Number
Email
Address

VAN SELLER
Seller Number
Company Name
Phone Number
Email
Premises Address
Warranty Provider
Warranty Phone Number
Warranty Address

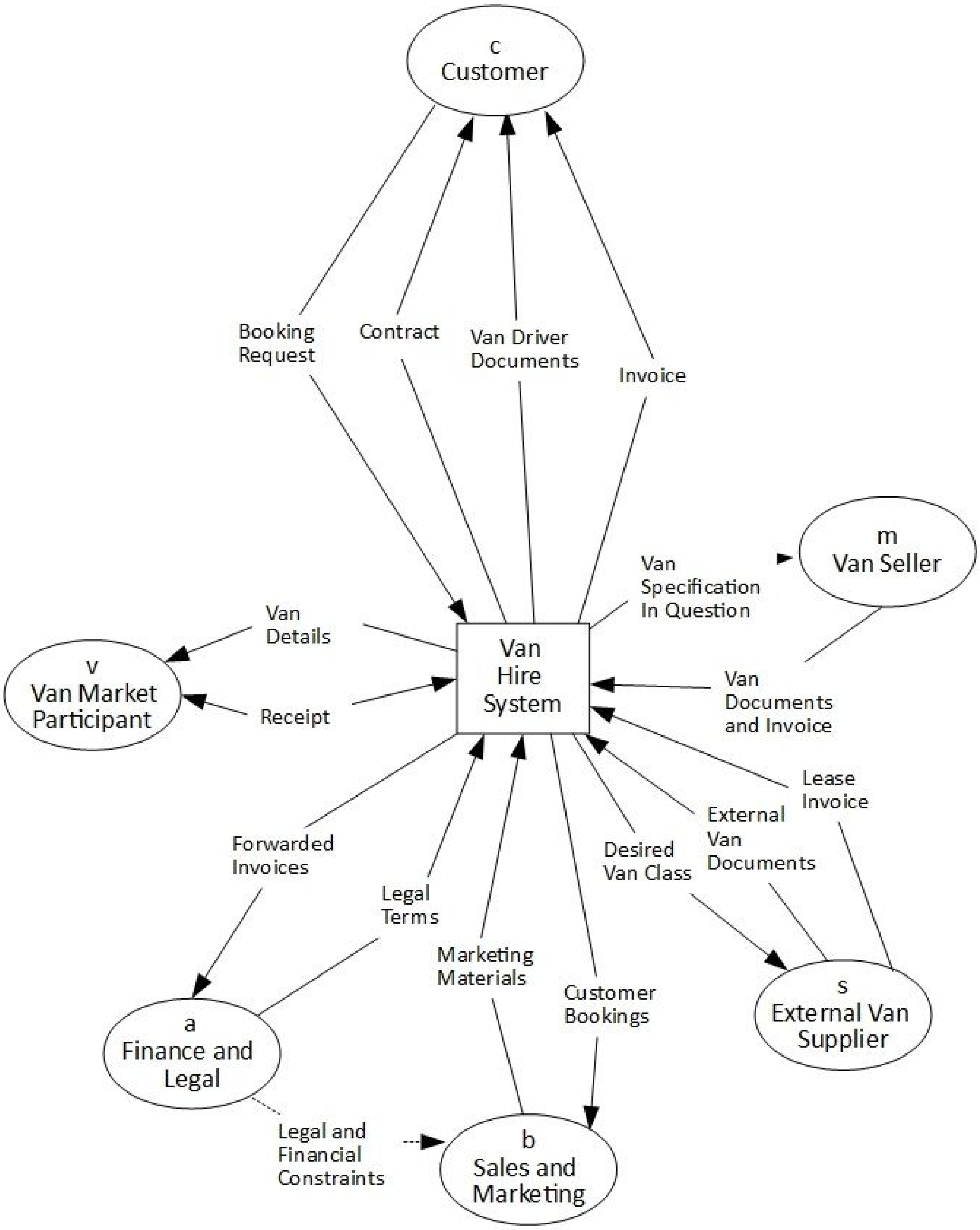
VAN MARKET PARTICIPANT
Name
Type
Phone Number
Email
Last Interaction Date
Last Interaction Comment

BOOKING INVOICE
Booking Invoice Number
*Booking Number
Incoming/Outgoing
Receiver/Sender
Status
Purpose
Amount
Comment

VAN INVOICE
Van Invoice Number
*Van License Plate Number
Incoming/Outgoing
Receiver/Sender
Status
Purpose
Amount
Comment

A Top level context diagram for the Data Flow Diagrams

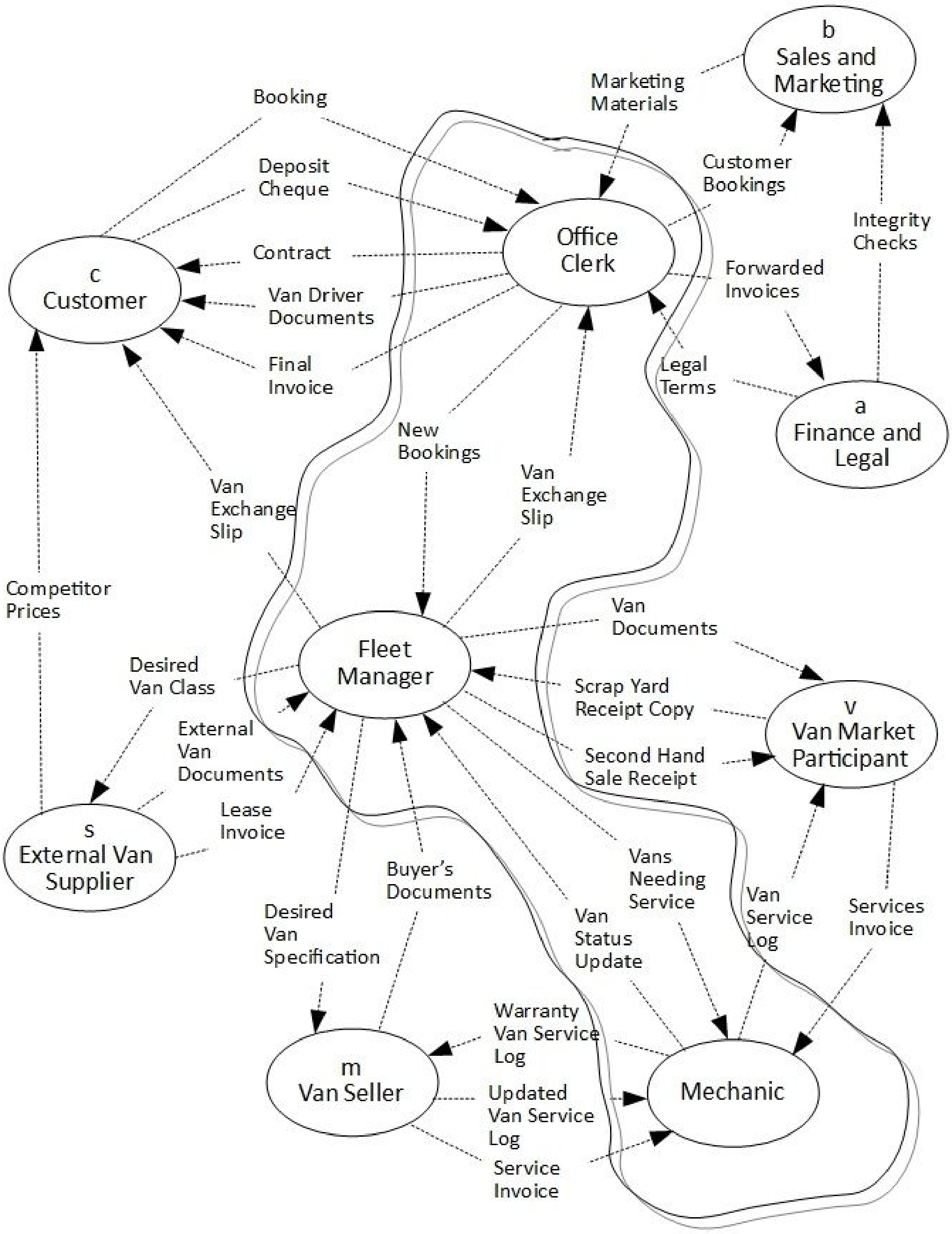
I assume the business is large enough (‘small-to-medium sized’) to have a small department dedicated to non-van hire operations, eg: payroll, finance, legal marketing, etc.


c
Customer
Booking Request
Contract
Van Driver Documents
Invoice
m
Van Seller
Van Specification In Question
Van Details
Van Hire System
v
Van Market Participant
Receipt
Van Documents and Invoice
Lease Invoice
External Van Documents
Desired Van Class
Forwarded Invoices
Legal Terms
Marketing Materials
Customer Bookings
s
External Van Supplier
a
Finance and Legal
Legal and Financial Constraints
b
Sales and Marketing

A Document Flow Diagram

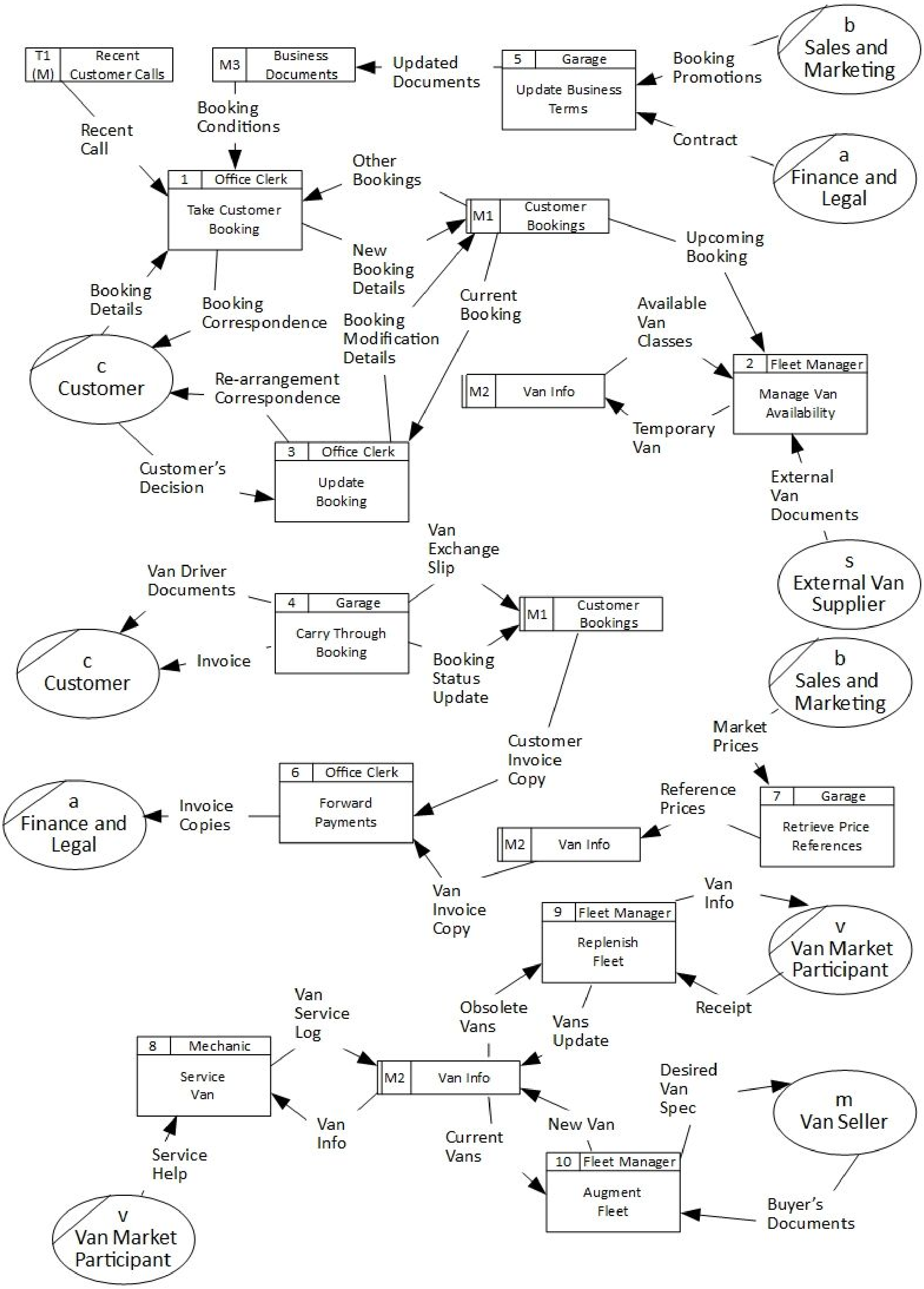
I assume that the company has three major employee roles in its van hire department - office clerk, fleet manager, and mechanic.

I assume that the current physical system does not involve a lot of internal communications because there are many ‘common’ data stores, so many users can access the same documents.


b
Sales and Marketing
Marketing Materials
Booking
Deposit Cheque
Customer Bookings
Integrity Checks
Office Clerk
c
Customer
Contract
Van Driver Documents
Forwarded Invoices
Final Invoice
Legal Terms
a
Finance and Legal
New Bookings
Van Exchange Slip
Van Exchange Slip
Competitor Prices
Van Documents
Fleet Manager
Desired Van Class
Scrap Yard Receipt Copy
v
Van Market Participant
External Van Documents
Second Hand Sale Receipt
Lease Invoice
s
External Van Supplier
Vans Needing Service
Buyer's Documents
Van Service Log
Services Invoice
Desired Van Specification
Van Status Update
Warranty Van Service Log
m
Van Seller
Updated Van Service Log
Mechanic
Service Invoice

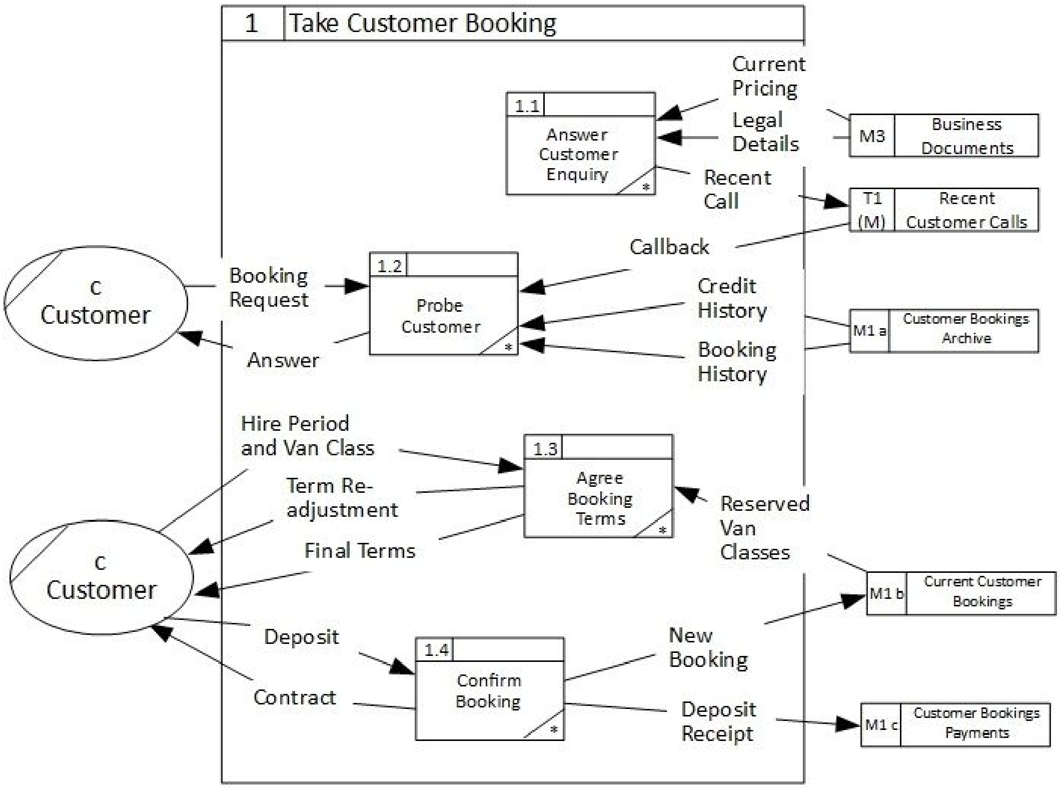
A Physical Level One DFD c. 10 processes

This is the *current* physical level 1 DFD.

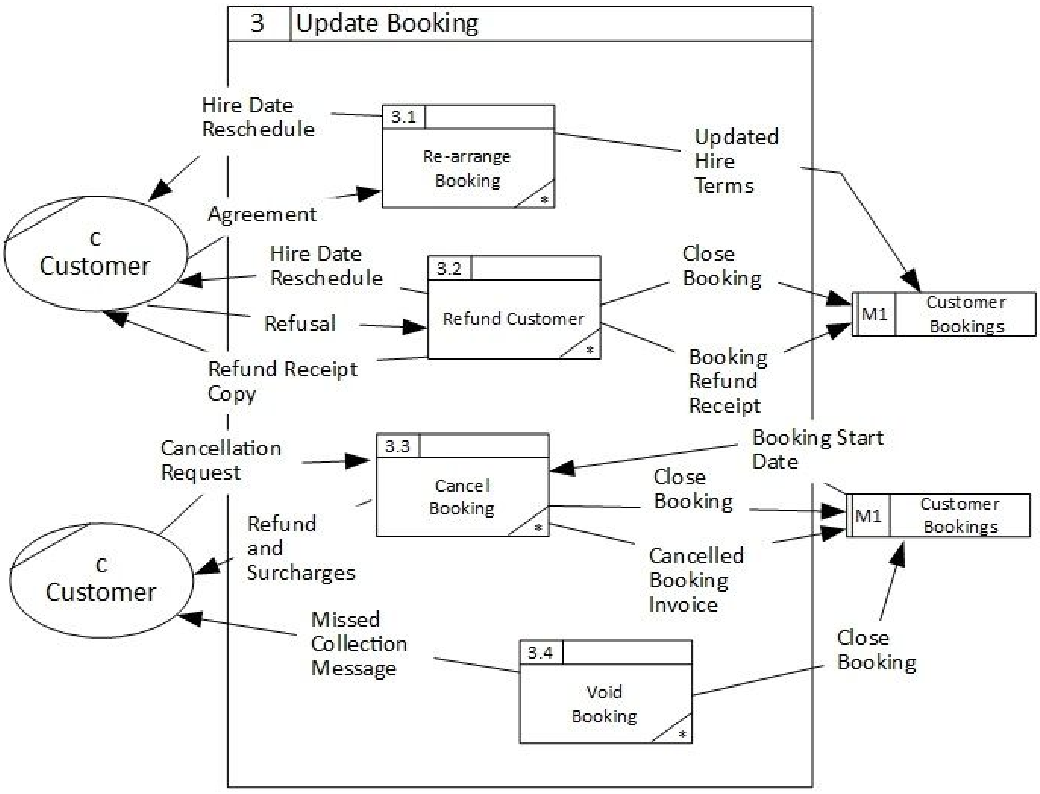

T1 (M) Recent Customer Calls
M3 Business Documents
Updated Documents
5 Garage Update Business Terms
Booking Promotions
b Sales and Marketing
Contract
a Finance and Legal
Recent Call
Booking Conditions
1 Office Clerk Take Customer Booking
Other Bookings
M1 Customer Bookings
New Booking Details
Upcoming Booking
Booking Details
Booking Correspondence
Booking Modification Details
Current Booking
Available Van Classes
c Customer
2 Fleet Manager Manage Van Availability
M2 Van Info
Re-arrangement Correspondence
Temporary Van
Customer's Decision
3 Office Clerk Update Booking
External Van Documents
s External Van Supplier
Van Exchange Slip
Van Driver Documents
4 Garage Carry Through Booking
M1 Customer Bookings
c Customer
Invoice
Booking Status Update
b Sales and Marketing
Market Prices
Customer Invoice Copy
6 Office Clerk Forward Payments
Reference Prices
7 Garage Retrieve Price References
a Finance and Legal
Invoice Copies
M2 Van Info
Van Invoice Copy
Van Info
9 Fleet Manager Replenish Fleet
v Van Market Participant
Receipt
Obsolete Vans
Vans Update
Van Service Log
8 Mechanic Service Van
M2 Van Info
Desired Van Spec
m Van Seller
Van Info
New Van
Current Vans
10 Fleet Manager Augment Fleet
Service Help
Buyer's Documents
v Van Market Participant

Take **two** of these physical processes from your DFD and explode them into physical level 2 DFDs.

One



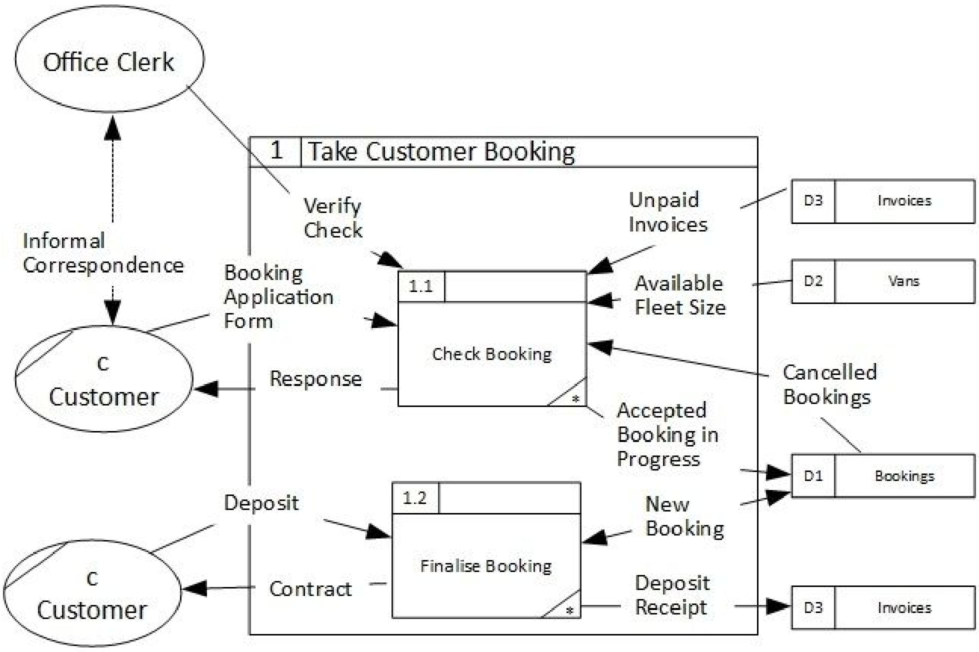
Two



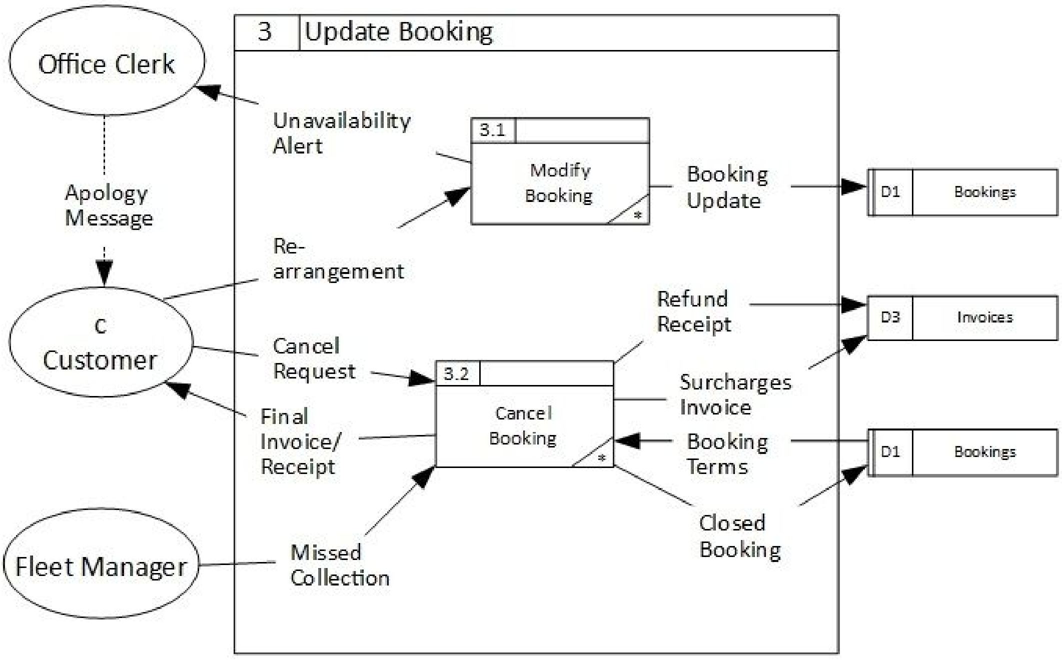
3
Update Booking
Hire Date Reschedule
3.1
Re-arrange Booking
*
Updated Hire Terms
c
Customer
Agreement
Hire Date Reschedule
3.2
Refund Customer
*
Close Booking
M1
Customer Bookings
Refusal
Refund Receipt Copy
Booking Refund Receipt
Cancellation Request
3.3
Cancel Booking
*
Booking Start Date
Close Booking
M1
Customer Bookings
c
Customer
Refund and Surcharges
Cancelled Booking Invoice
Missed Collection Message
3.4
Void Booking
*
Close Booking

Give the Logical DFDs for the latter two physical level 2 DFDs.

One



Two


Office Clerk
Apology Message
c Customer
Fleet Manager
3
Update Booking
Unavailability Alert
Re-arrangement
Cancel Request
Final Invoice/ Receipt
Missed Collection
3.1
Modify Booking
*
Booking Update
3.2
Cancel Booking
*
Refund Receipt
Surcharges Invoice
Booking Terms
Closed Booking
D1
Bookings
D3
Invoices
D1
Bookings

Give two functions from any of the DFDs and explain their purpose using a structured English.

I assume that *structured english* is pseudocode.

1. UPDATE BOOKING

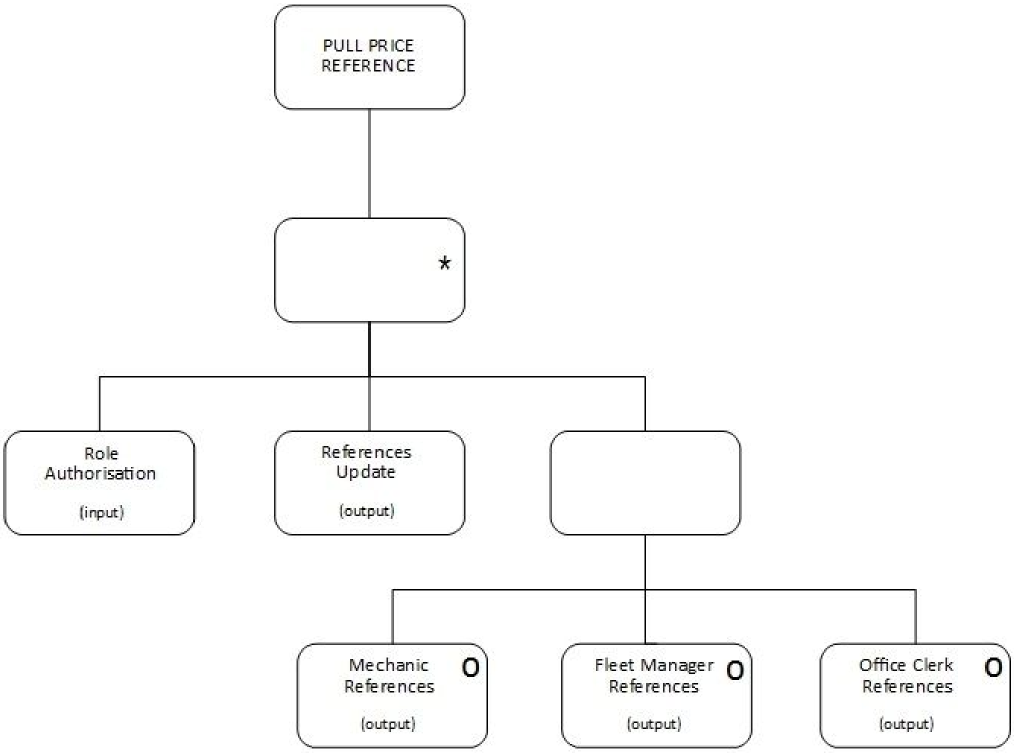
```
* Change booking details or booking state, with appropriate consequences

Input Office Clerk authorisation
IF Customer wants to rearrange Booking THEN
   Input Customer new start date preference
   Update start date value for Booking in Customer Booking database
ELSE IF Customer wants to refund Booking THEN
   Close Booking in Customer Booking database
   Get Customer deposit amount for Booking from Payments database
   Re-input Office Clerk authorisation
   Display electronic receipt
   Transfer deposit refund payment to Customer
   Add electronic receipt to Payments database
   Send electronic receipt to Customer
ELSE IF Customer wants to cancel Booking THEN
   Close Booking in Customer Booking database
   IF time until start date < two days THEN
      Cancellation_fee = Customer deposit amount
   ELSE
      Cancellation_fee = Customer deposit amount * Cancellation sliding scale
   Add Cancellation_fee to Customer invoice
   Re-input Office Clerk authorisation
   Display electronic invoice
   Transfer remainder to Customer
   Send electronic invoice to Customer
ELSE IF Customer missed collection THEN
   Close Booking
   Send alert to Customer
ELSE
   Input Customer new start date, end date, van class preference
   Update Booking start date, end date, van class preference
ENDIF

EXIT
```

2. PULL PRICE REFERENCE

```
* Show most recent market prices for comparison to various users
Authorise:
    Get Authorisation

IF Authorised and Price Reference not most recent THEN
   Pull updates from External Marketing Department Database
   * This updates references for all users, regardless of who is the calling user

IF Authorisation is Fleet Manager THEN
    Show price references for seller vans
    Show price references for external supplier vans
    Show price references for scrap yards
    Show price references for second hand sales
    IF want to see more THEN
       Show Mechanic References
    ENDIF
ELSE IF Authorisation is Mechanic THEN
    Mechanic References:
       Show price references for van services
       Show price references for van parts
ELSEIF Authorisation is Office Clerk THEN
    Show all price references
ENDIF

IF want to re-authorise again THEN
   GOTO Authorise
ENDIF

EXIT
```

Give the I/O diagram for one of the Functions within this DFD.

FUNCTION TWO



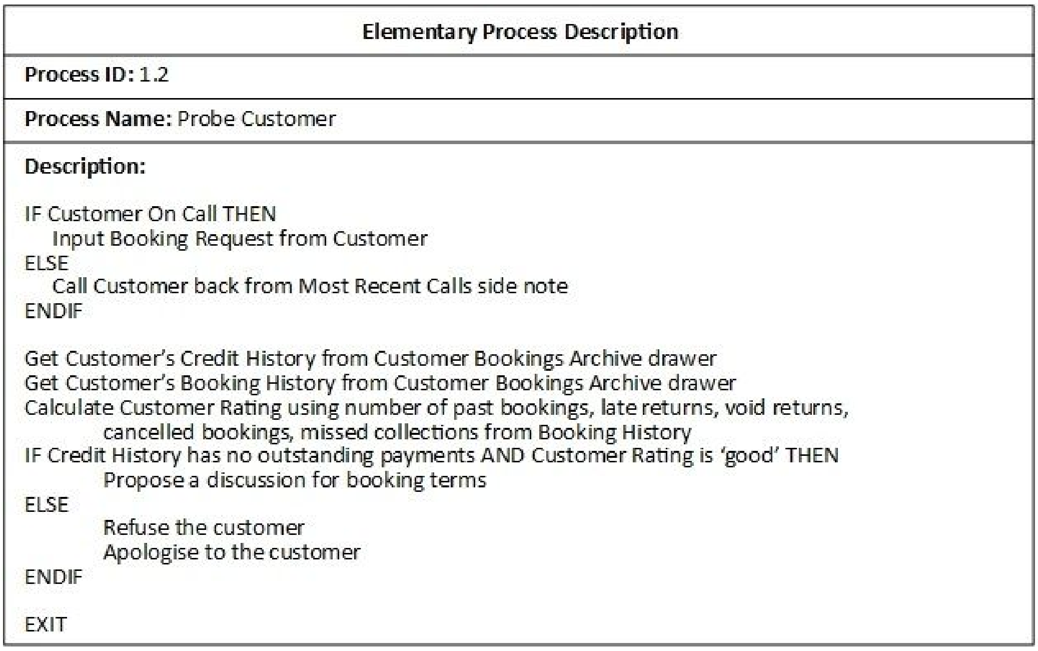
Produce 5 data dictionary entries as described below. (Use the data dictionary forms examples on Loop as a guide)

Five descriptions down below

2 process descriptions (Structured English)

I assume that the descriptions are for *elementary* processes (from level 2 physical DFD), and that *structured english* is the same structured english as in the function explanations.

One

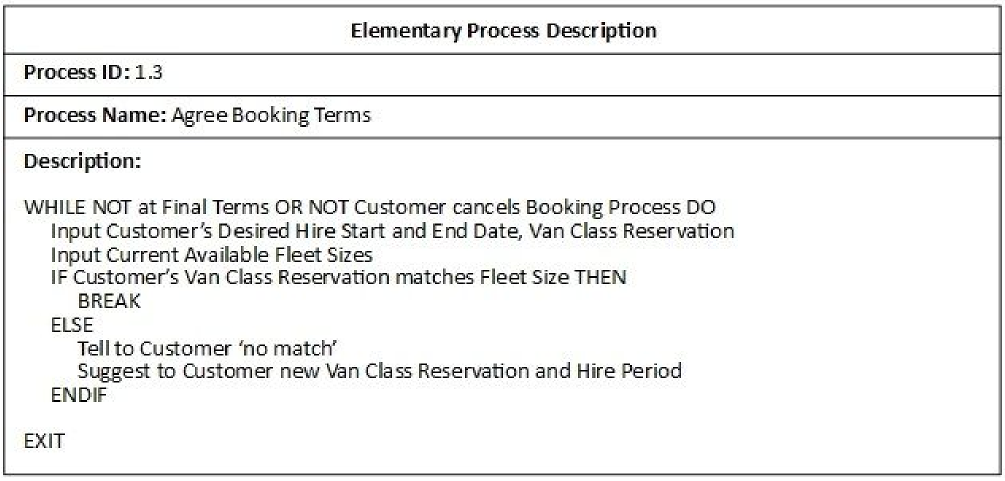


Elementary Process Description
Process ID: 1.2
Process Name: Probe Customer

Description:

```
IF Customer On Call THEN
    Input Booking Request from Customer
ELSE
    Call Customer back from Most Recent Calls side note
ENDIF

Get Customer’s Credit History from Customer Bookings Archive drawer
Get Customer’s Booking History from Customer Bookings Archive drawer
Calculate Customer Rating using number of past bookings, late returns, void returns,
          cancelled bookings, missed collections from Booking History
IF Credit History has no outstanding payments AND Customer Rating is ‘good’ THEN
          Propose a discussion for booking terms
ELSE
          Refuse the customer
          Apologise to the customer
ENDIF

EXIT
```

Two



Elementary Process Description

Process ID: 1.3

Process Name: Agree Booking Terms

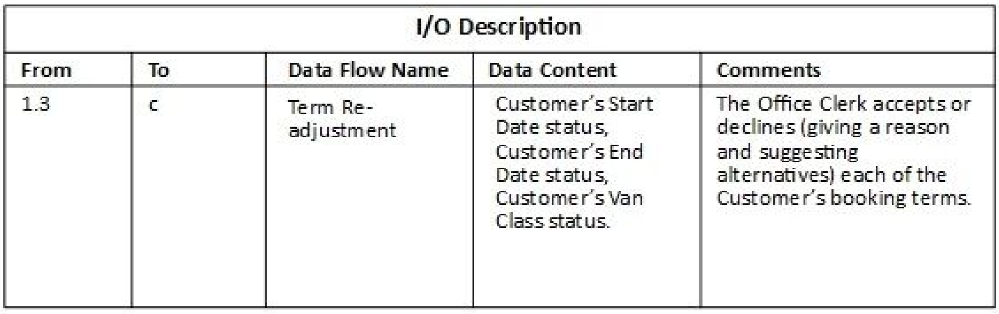
Description:

```
WHILE NOT at Final Terms OR NOT Customer cancels Booking Process DO
    Input Customer's Desired Hire Start and End Date, Van Class Reservation
    Input Current Available Fleet Sizes
    IF Customer's Van Class Reservation matches Fleet Size THEN
        BREAK
    ELSE
        Tell to Customer 'no match'
        Suggest to Customer new Van Class Reservation and Hire Period
    ENDIF

EXIT
```

1 Data flow (its components, location etc. p139)

I assume that I am documenting the lowest level data flow.

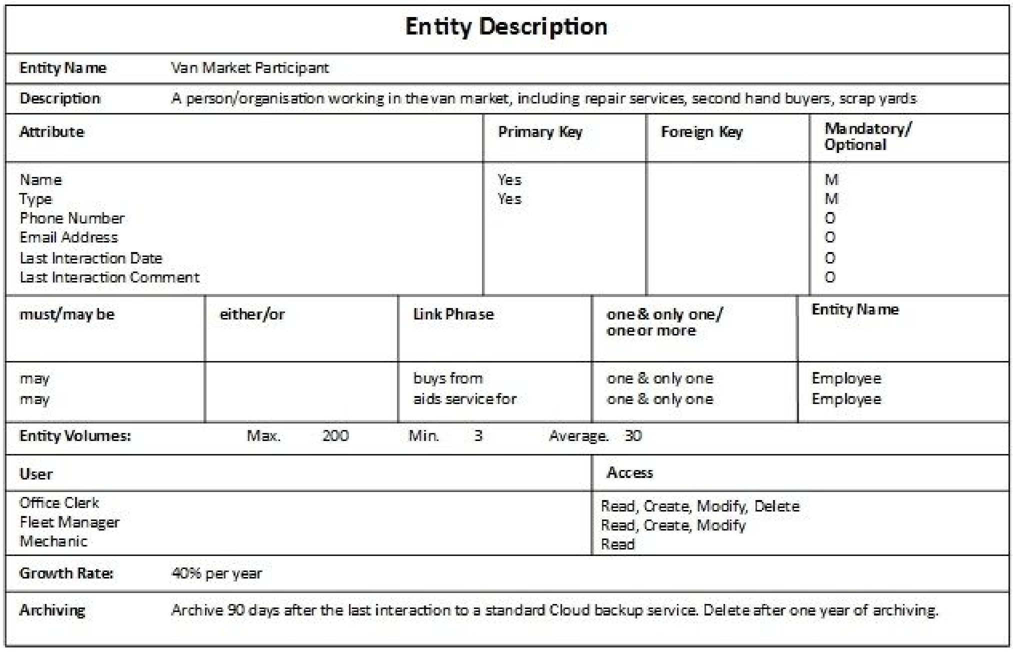
I assume that the *components* and *location* of the data flow are covered by the Data Content/Comment and the From/To headers in the I/O Description.



I/O Description

From	To	Data Flow Name	Data Content	Comments
1.3	c	Term Re-adjustment	Customer's Start Date status, Customer's End Date status, Customer's Van Class status.	The Office Clerk accepts or declines (giving a reason and suggesting alternatives) each of the Customer's booking terms.

1 Entity Description (see Weaver page 108)



Entity Description

Entity Name	Van Market Participant
Description	A person/organisation working in the van market, including repair services, second hand buyers, scrap yards

Attribute	Primary Key	Foreign Key	Mandatory/ Optional
Name	Yes		M
Type	Yes		M
Phone Number			O
Email Address			O
Last Interaction Date			O
Last Interaction Comment			O

must/may be	either/or	Link Phrase	one & only one/ one or more	Entity Name
may		buys from	one & only one	Employee
may		aids service for	one & only one	Employee

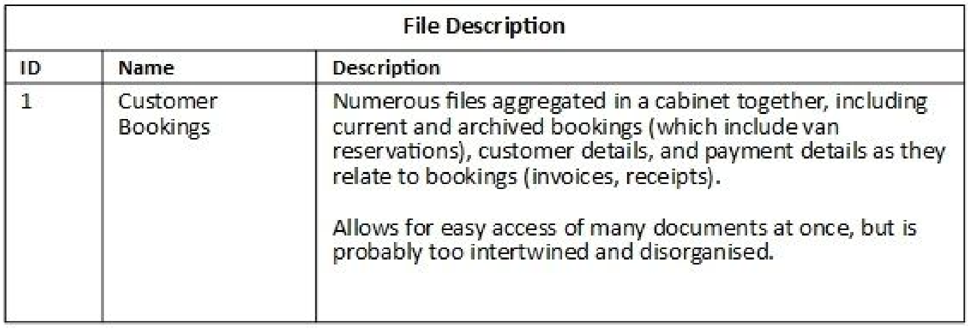
Entity Volumes:	Max.	200	Min.	3	Average.	30

User	Access
Office Clerk	Read, Create, Modify, Delete
Fleet Manager	Read, Create, Modify
Mechanic	Read

Growth Rate:	40% per year
Archiving	Archive 90 days after the last interaction to a standard Cloud backup service. Delete after one year of archiving.

1 Data Store (File Description)

Based on the project help file, I assume this is the format required for the data store's file description.



File Description

ID	Name	Description
1	Customer Bookings	Numerous files aggregated in a cabinet together, including current and archived bookings (which include van reservations), customer details, and payment details as they relate to bookings (invoices, receipts). Allows for easy access of many documents at once, but is probably too intertwined and disorganised.

FINALLY THE END OF THIS TERRIFIC THING