

DCU School of Computing Assignment Submission

Student Name(s): Muhammad Umar
Student Number(s): 17313893
Programme: BSc in Computer Applications
Project Title: SSADM
Module code: CA214
Lecturer: Renaat Verburgen
Project Due Date: 10 December 2018

Declaration

I declare that this material, which I now submit for assessment, is entirely my 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 work. I understand that plagiarism, collusion, and copying is a grave and serious offence in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion, or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, 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.

I have not copied or paraphrased an extract of any length from any source without identifying the source and using quotation marks as appropriate. Any images, audio recordings, video or other materials have likewise been originated and produced by me or are fully acknowledged and identified.

This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. I have read and understood the referencing guidelines found at <http://www.library.dcu.ie/citing&refguide08.pdf> and/or recommended in the assignment guidelines.

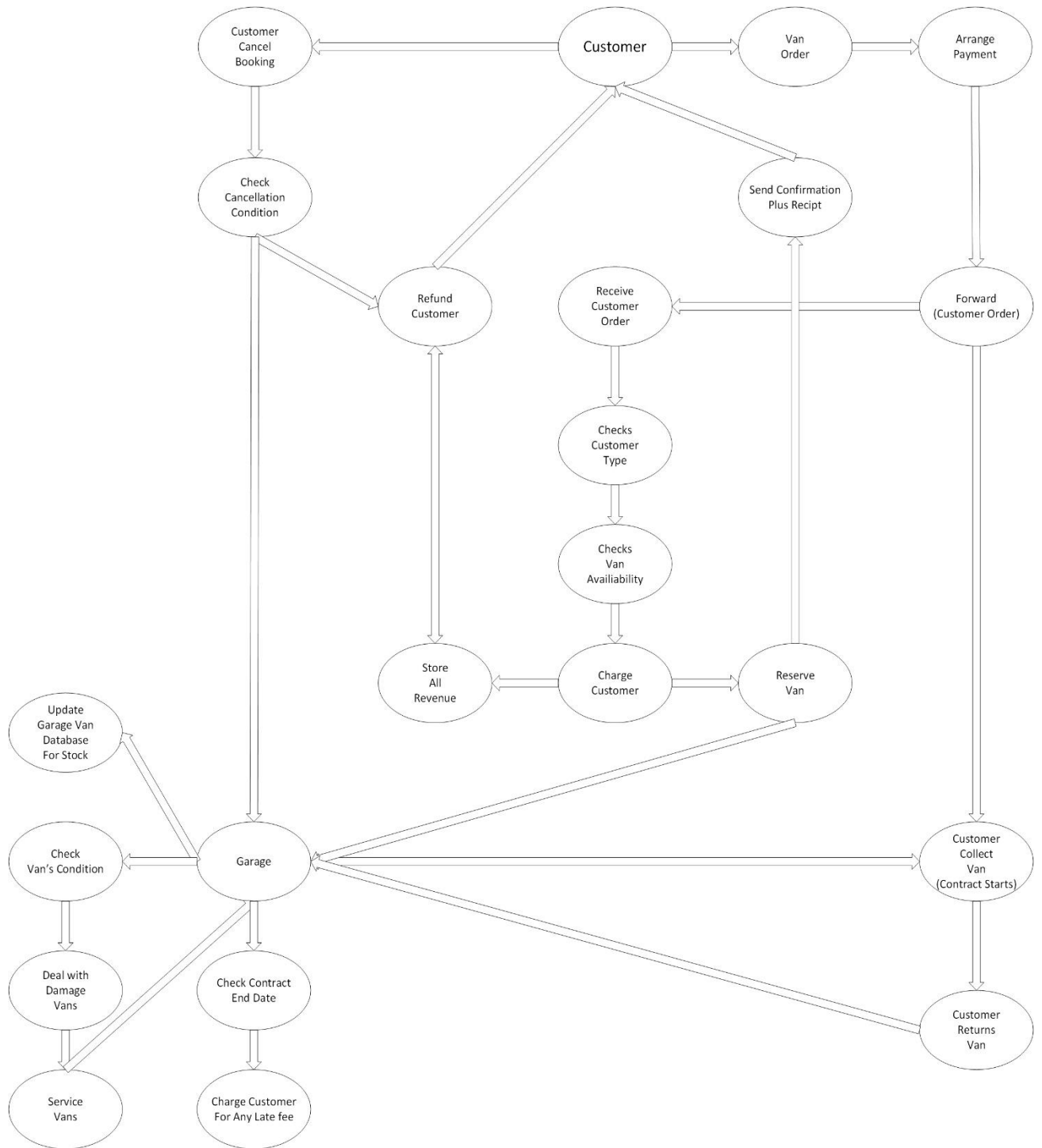
I understand that I may be required to discuss with the module lecturer/s the contents of this submission.

I/me/my incorporates we/us/our in the case of group work, which is signed by all of us.

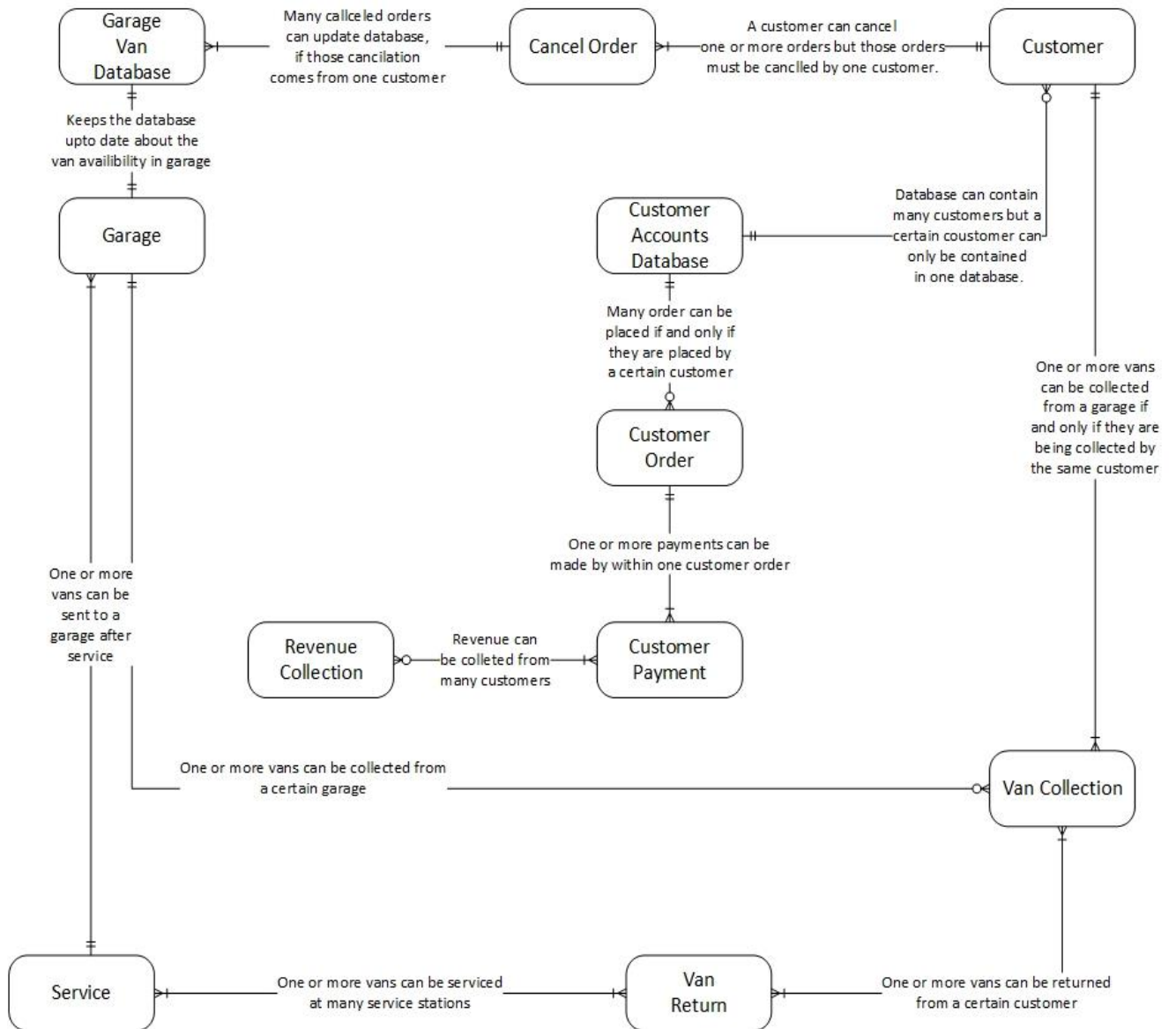
Signed: Muhammad Umar

Submission Date: 07 Dec 2018

Business Activity Modelling



Logical Data Model



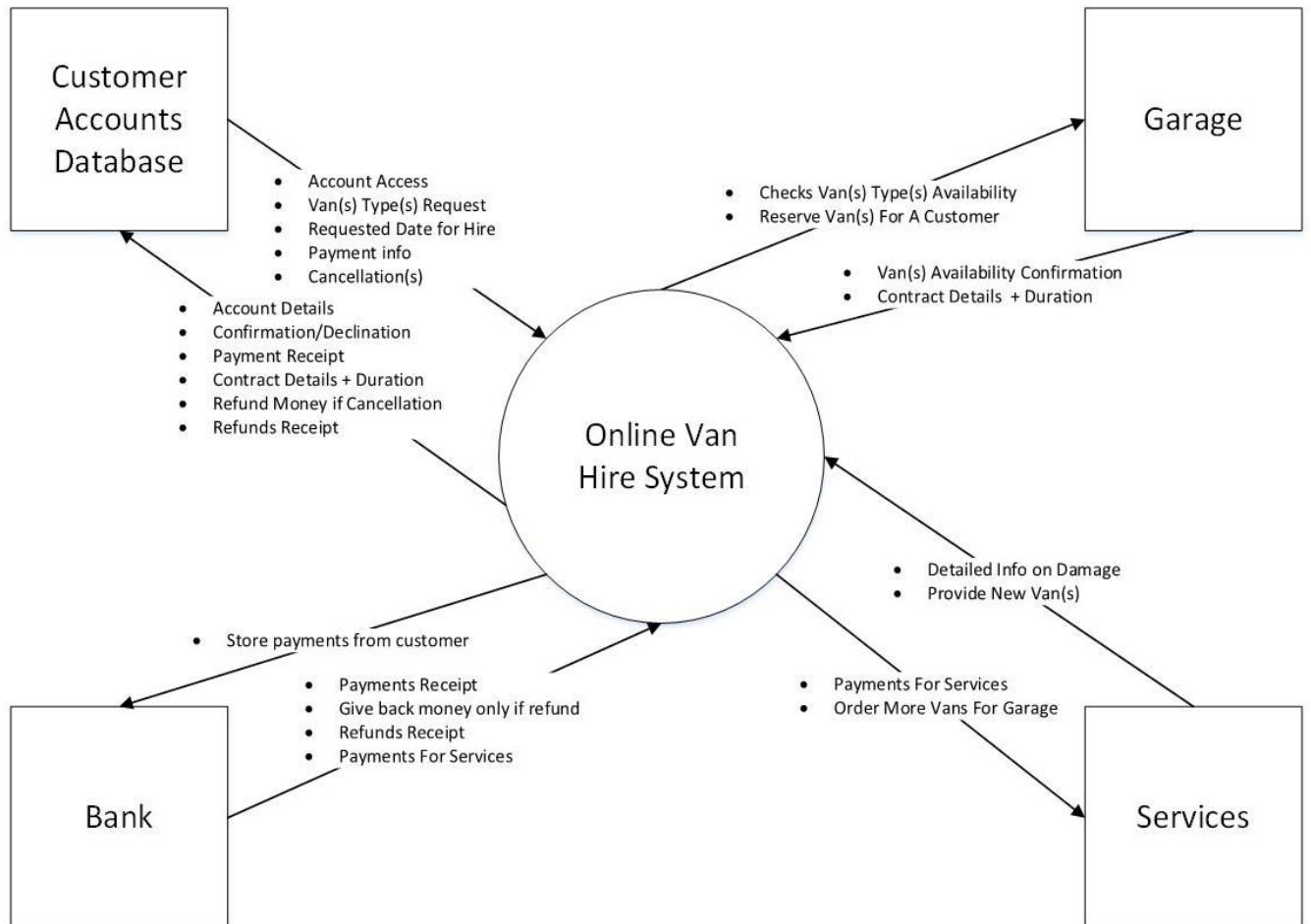
LDM Relationships

	Customer	Customer Account Database	Customer Order	Customer Payment	Revenue Collection	Van Collection	Van Return	Service	Garage	Cancel Order	Garage Van Database
Customer		X				X				X	
Customer Account Database	X		X								
Customer Order		X		X							
Customer Payment			X		X						
Revenue Collection				X							
Van Collection	X						X		X		
Van Return						X		X			
Service							X		X		
Garage						X		X			X
Cancel Order	X										X
Garage Van Database							X		X	X	

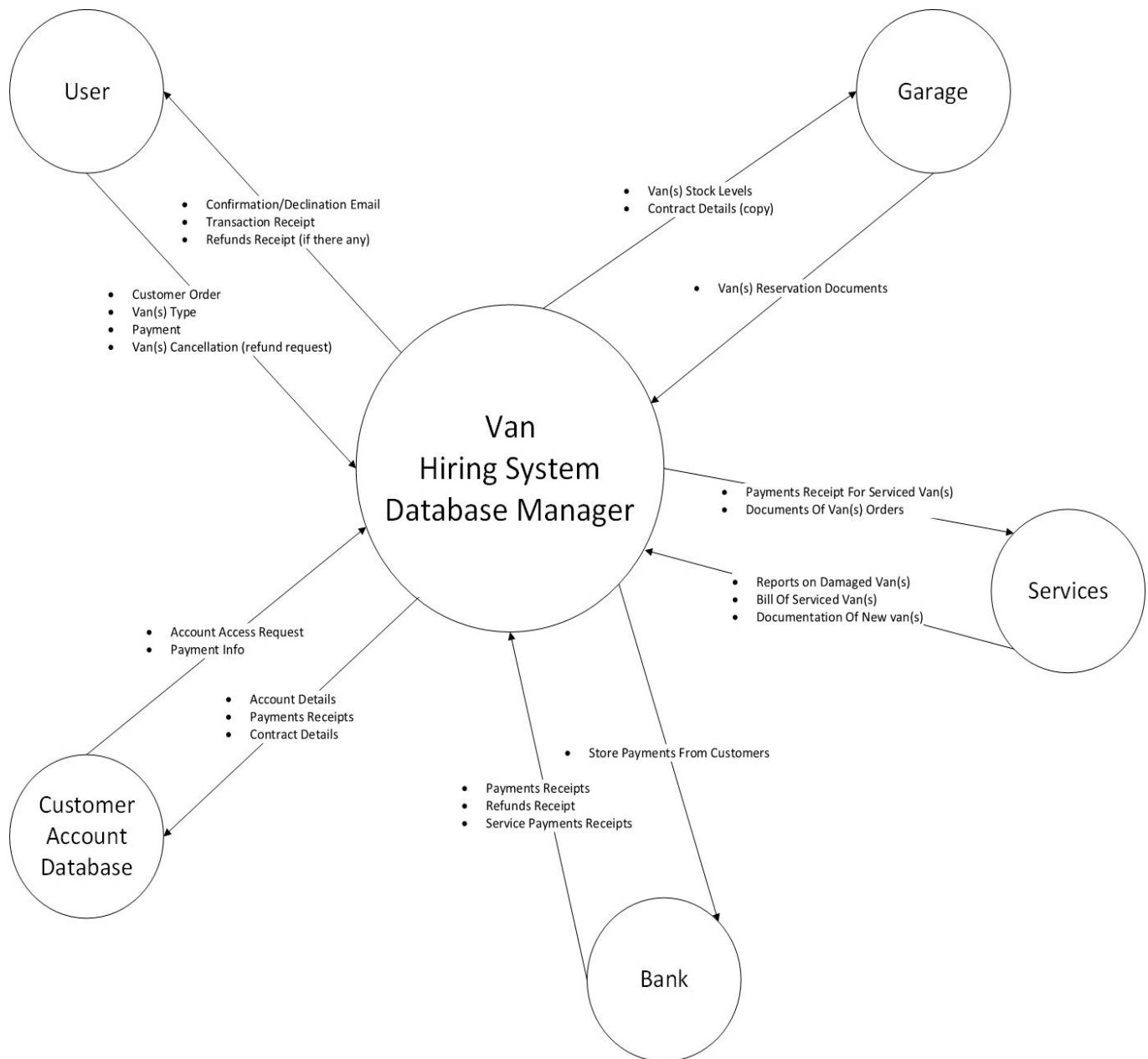
Relationship Degrees:

- Customer to Customer Account Database: N:1
- Customer Account Database to Customer Order: 1:M
- Customer to Cancel Order: 1:M
- Customer Order to Customer Payment: 1:M
- Customer Payment to Revenue Collection: M:M
- Customer to Van Collection: 1:M
- Van Collection to Van Return: M:M
- Van Collection to Garage: M:M
- Van Return to Service: M:M
- Service to Garage: 1:M
- Garage to Garage Van Database: 1:1
- Cancel Order to Garage Van Database: 1:M

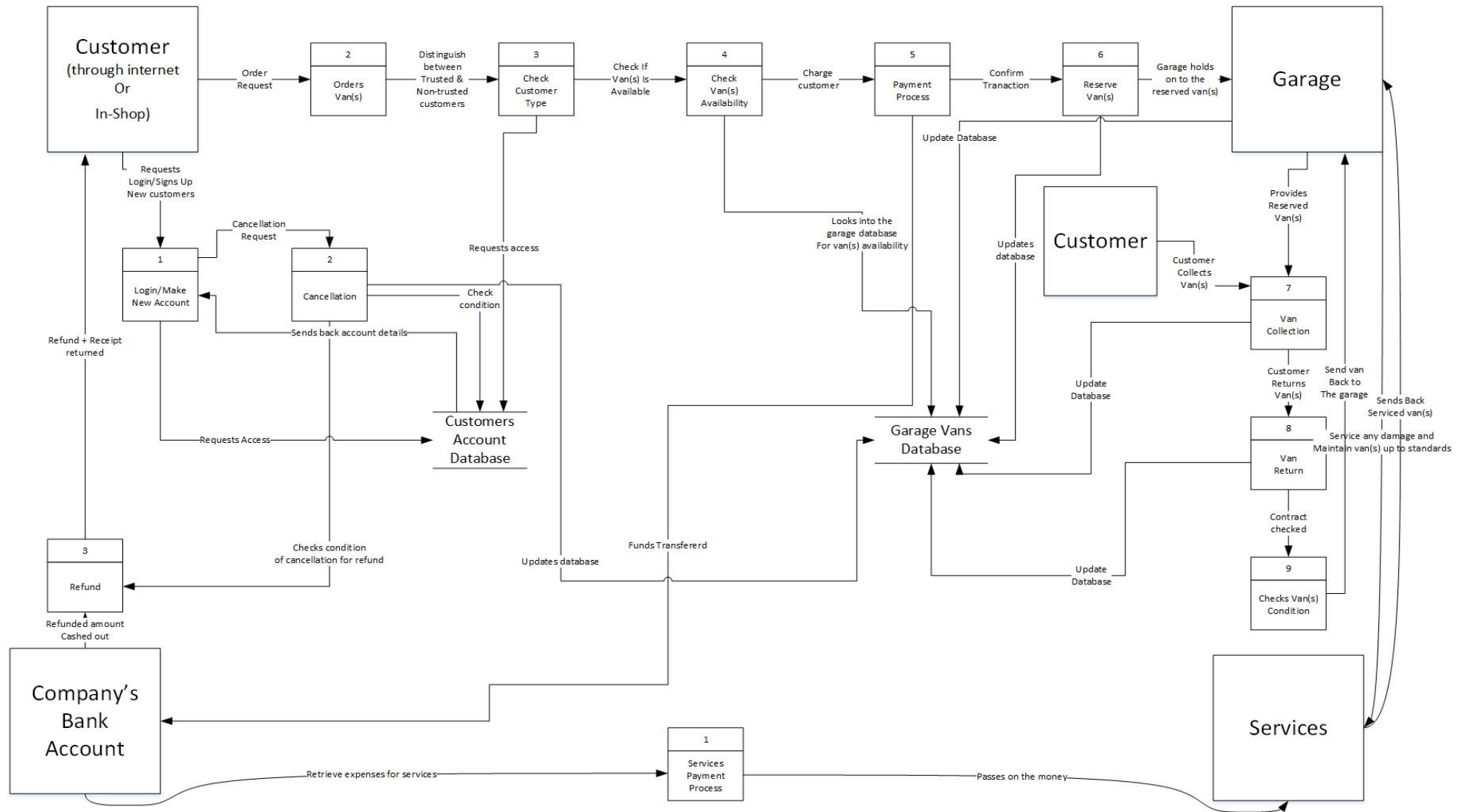
Top-level context diagram for the Data Flow Diagrams



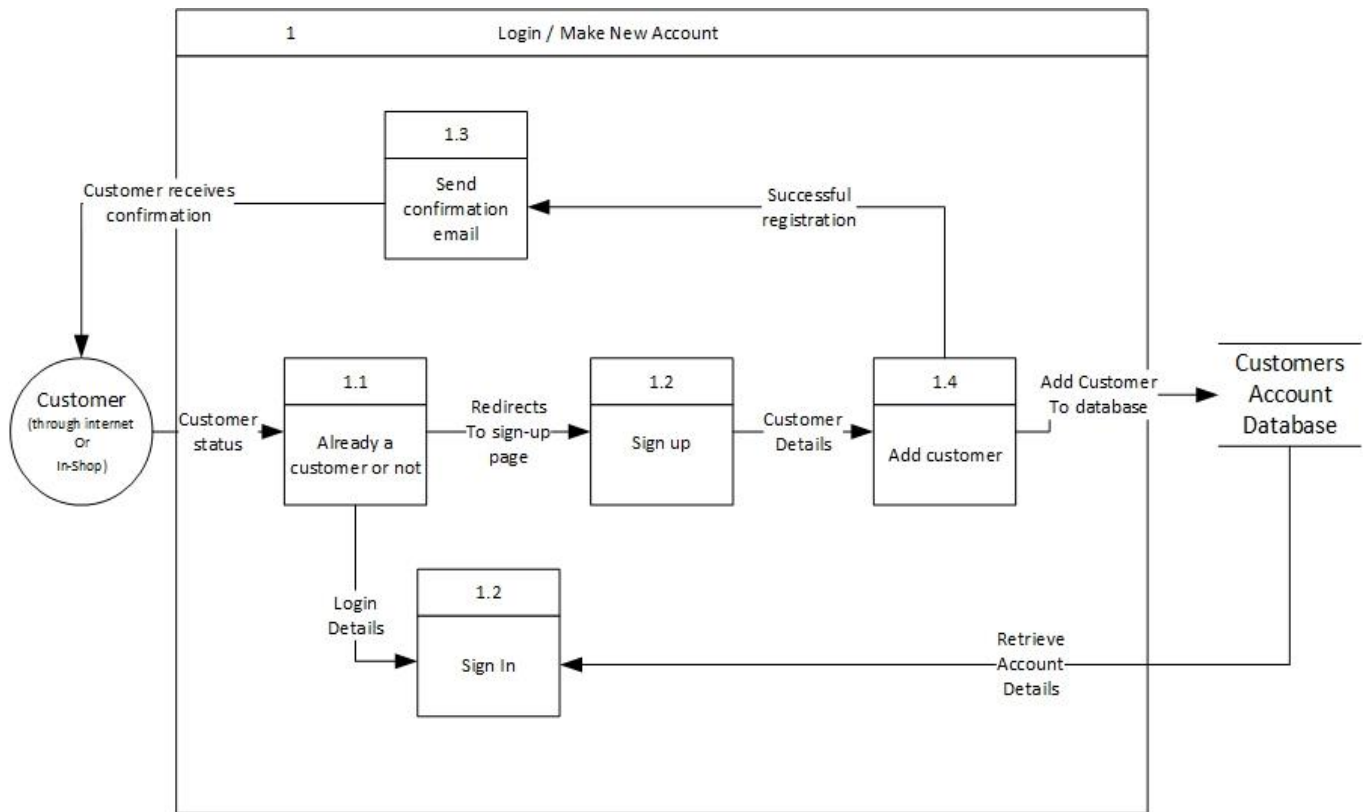
Document Flow Diagram



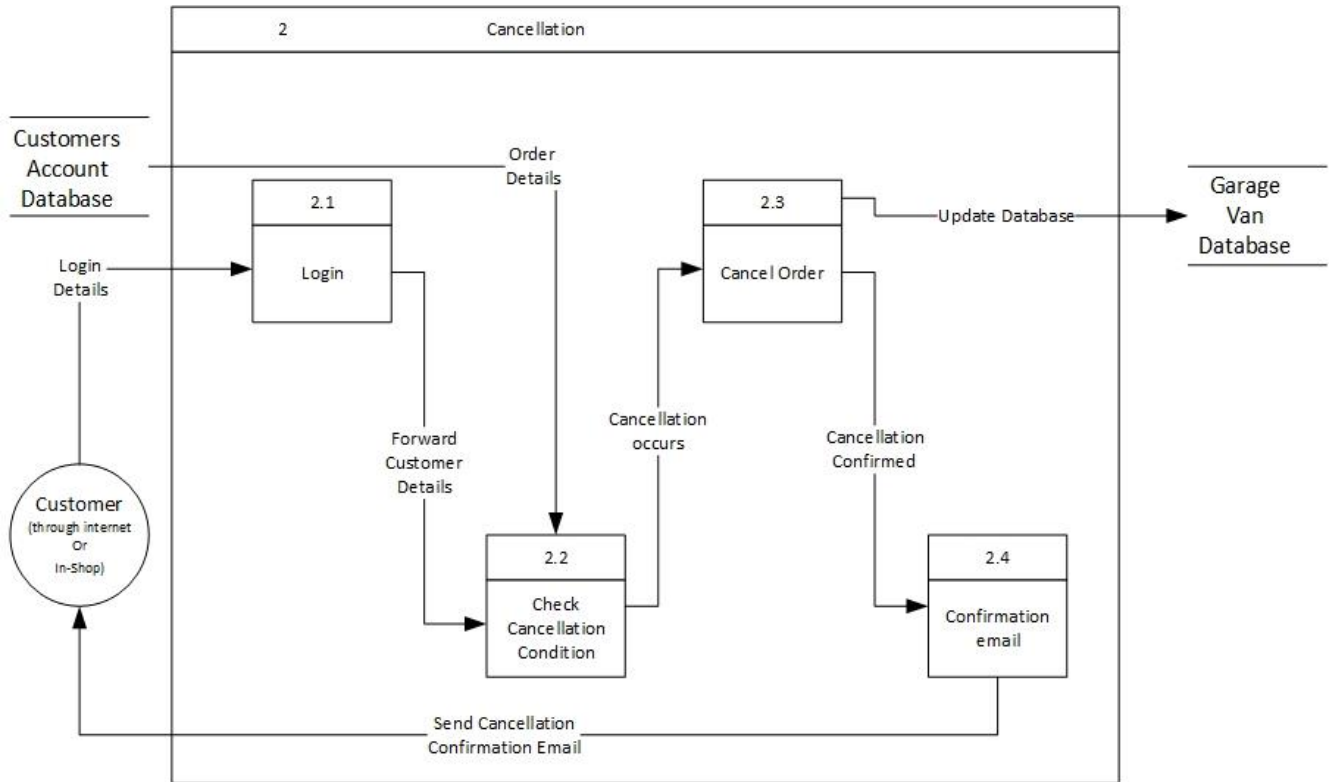
Physical Level One DFD



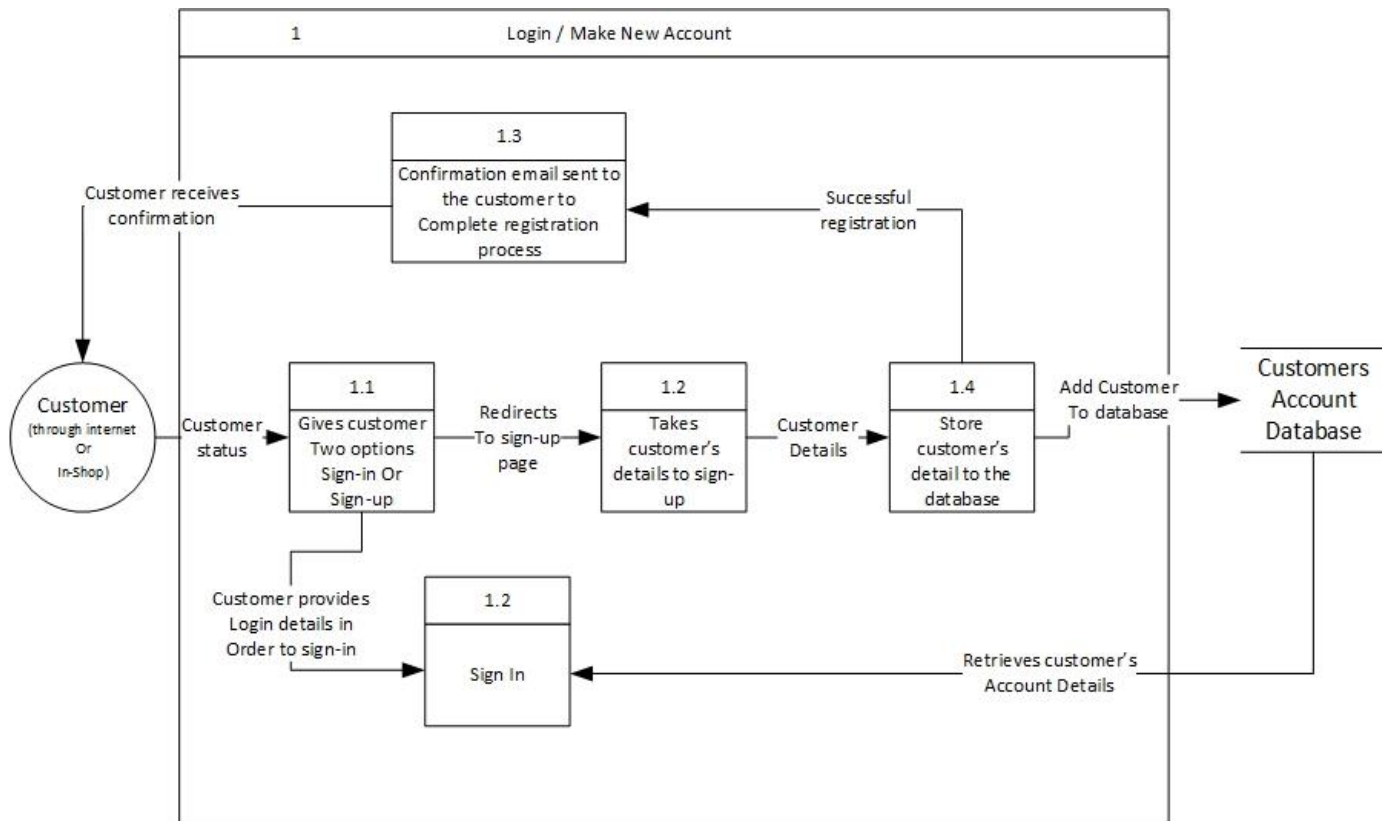
Physical level 2 DFD



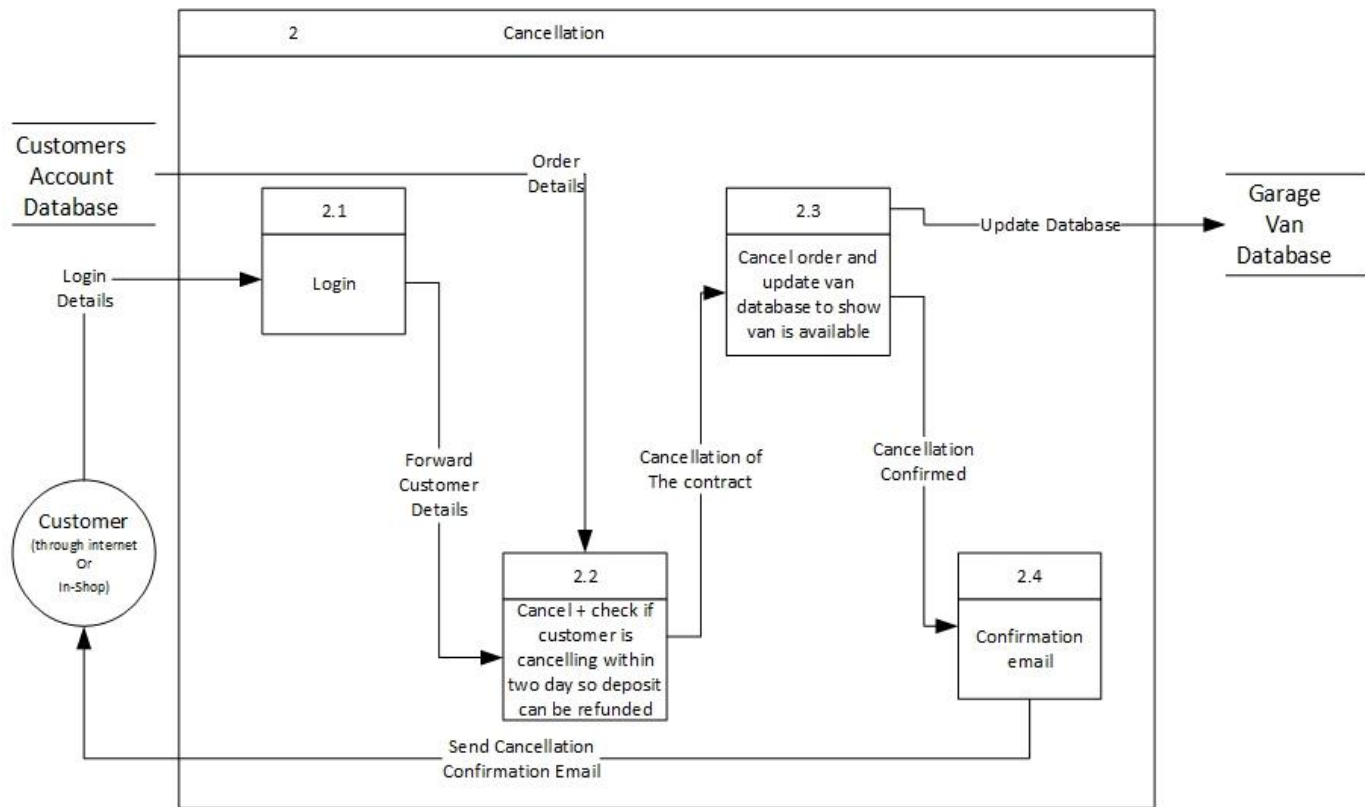
Physical level 2 DFD



Logical DFD Of Physical level 2 DFD



Logical DFD Of Physical level 2 DFD



1: Function to Check Cancellation Condition

Explanation in Structured English pseudocode:

START

CHECK CANCELLATION CONDITION FUNCTION:

CHECK customer's account details from customers account database

CHECK customer's order details

IF cancelled within certain days/hours after ordering:

 CHANGE customer's refund status to *TRUE*

ELSE:

 CHANGE customer's refund status to *FALSE*

EXIT

2: Function to Cancel Order

Explanation in Structured English pseudocode:

START

CANCEL ORDER FUNCTION:

CHECK customer's refund status

IF TRUE:

 CANCEL customer's order

 SEND customer a confirmation email for successful cancellation

 UPDATE garage van database

 DO refund

ELSE:

 CANCEL customer's order

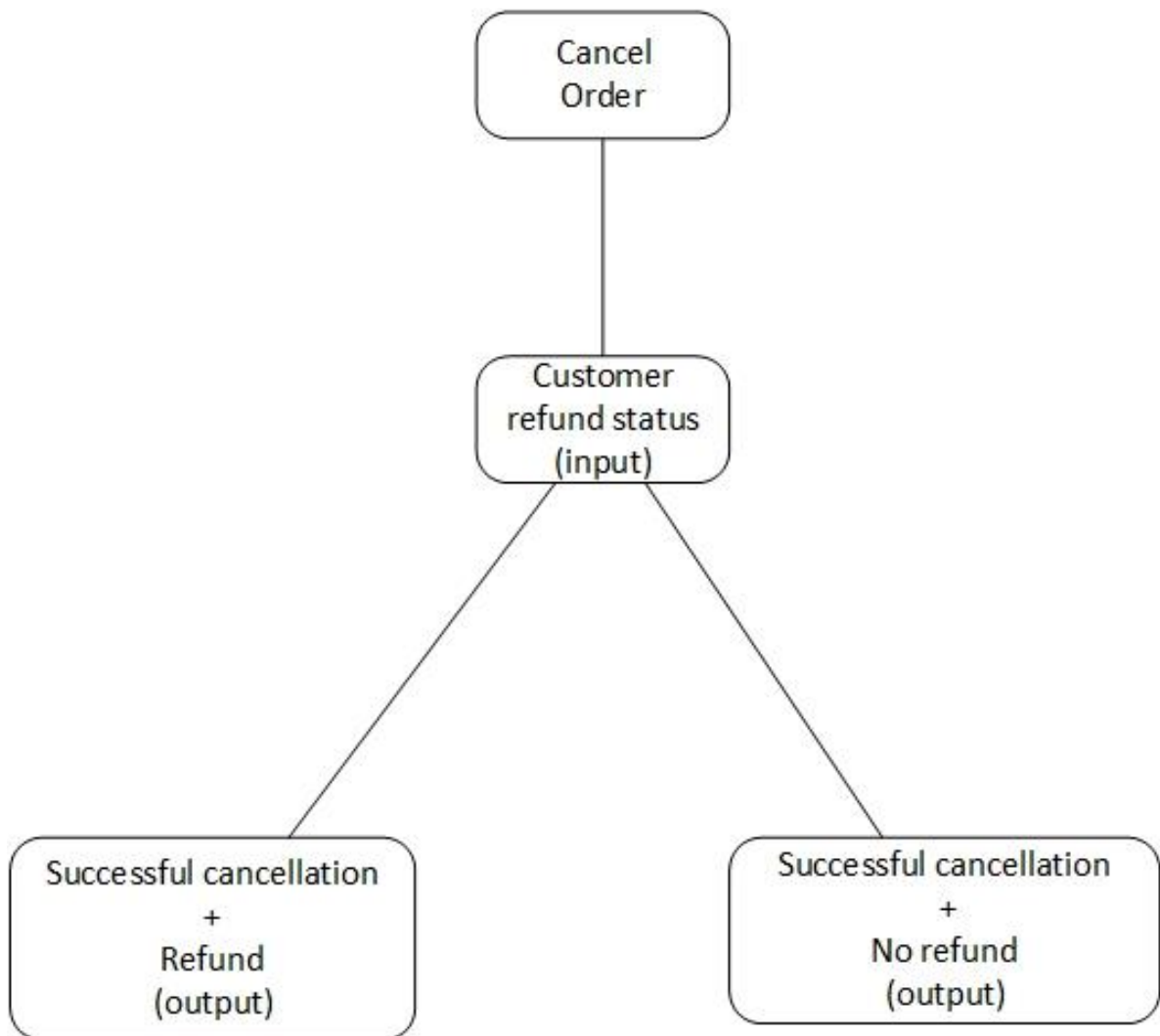
 SEND customer a confirmation email for successful cancellation

 UPDATE garage van database

 DON'T DO refund

EXIT

I/O diagram for Cancel Order



1: Process Description – Login/Make New Account

Note: Website home-page generates with two options either login or new customer which will proceed customer to sign-up and then login.

Explanation in Structured English pseudocode:

START

LOGIN/SIGN-UP NEW CUSTOMERS FUNCTION:

#WAIT FOR CUSTOMER TO CHOOSE THE OPTION BETWEEN LOGIN & NEW CUSTOMER

IF login:

 GET customer's login details

 SEND login details to customers account database

 RETRIEVE customer's account details from customers account database

ELSE IF new customer:

 PROCEED to sign-up page

 GET customer's details

 SEND customer a confirmation email

 ADD customer to customers account database

EXIT

2: Process Description – Cancellation

Note: Cancellation can cancel customer's order but can also do refund depending how long the customer has taken to cancel the order from the date he has made the order.

Explanation in Structured English pseudocode:

START

CANCELLATION FUNCTION:

GET customer's login details

SEND login details to customers account database

RETRIEVE customer's account details from customers account database

RETRIEVE customer's order details

IF cancelled within certain days/hours after ordering:

 CANCEL customer's order

 SEND customer a confirmation email for successful cancellation

 UPDATE garage van database

 DO refund

ELSE:

 CANCEL customer's order

 SEND customer a confirmation email for successful cancellation

 UPDATE garage van database

 DON'T DO refund

EXIT

Entity Description (external)

Entity Name	Entity ID	Entity Description
Customer/User	c	Customer/user is a person is interested in hiring vans and he/she sign's up to our website. After successful registration he/she can hire/reserve/cancel van(s) order(s) in our company

Data Flow Diagram

Data Flow Name	Origin	Destination	Data Content	Comments
Sign-up Request	c	Customers Account database	<ul style="list-style-type: none">• Full Name• DOB• Address• Payment Methods• Username• Password• Customer ID	All fields in this sign-up process are compulsory
Confirmation Email	Customers Account Database	c	Automated Email	An automatically Generated email Which confirms successful sign-up

ENTITY DESCRIPTION

Entity Name: Van(s) Order				
Description: Customer’s request for reservation of van(s) from this company’s supplier				
ATTRIBUTE	PRIMARY KEY	FOREIGN KEY	MANDATORY/OPTIONAL	
Customer ID	YES	YES	M	
Supplier ID			M	
Order Date			M	
Order Status			O	
Contract Start Date			M	
Contract End Date			M	
MUST/MAY BE	EITHER/OR	LINK PHRASE	ONE & ONLY ONE / ONE OR MORE	ENTITY NAME
Must be May Must		Placed with Order Result in	One & Only one One or more One or more	Customer Van(s) Payment(s)
Entity Volumes: Max = 10 Min = 1 Avg = 5				
USER:		ACCESS:		
Customer Supplier (Company’s Employee)		Read, Create Read, create, modify, delete		
Archiving: Van orders from each customer should be archived for a year after last van order has been made.				

DATA STORE

ID of Data Store	1
Data Store Name	Customers Account Database
Data store Description:	This data store contains all the information about the customer and the order they have placed. It also contains those order's start and end date. It also helps hiring company to distinguish between their trusted & new customers.