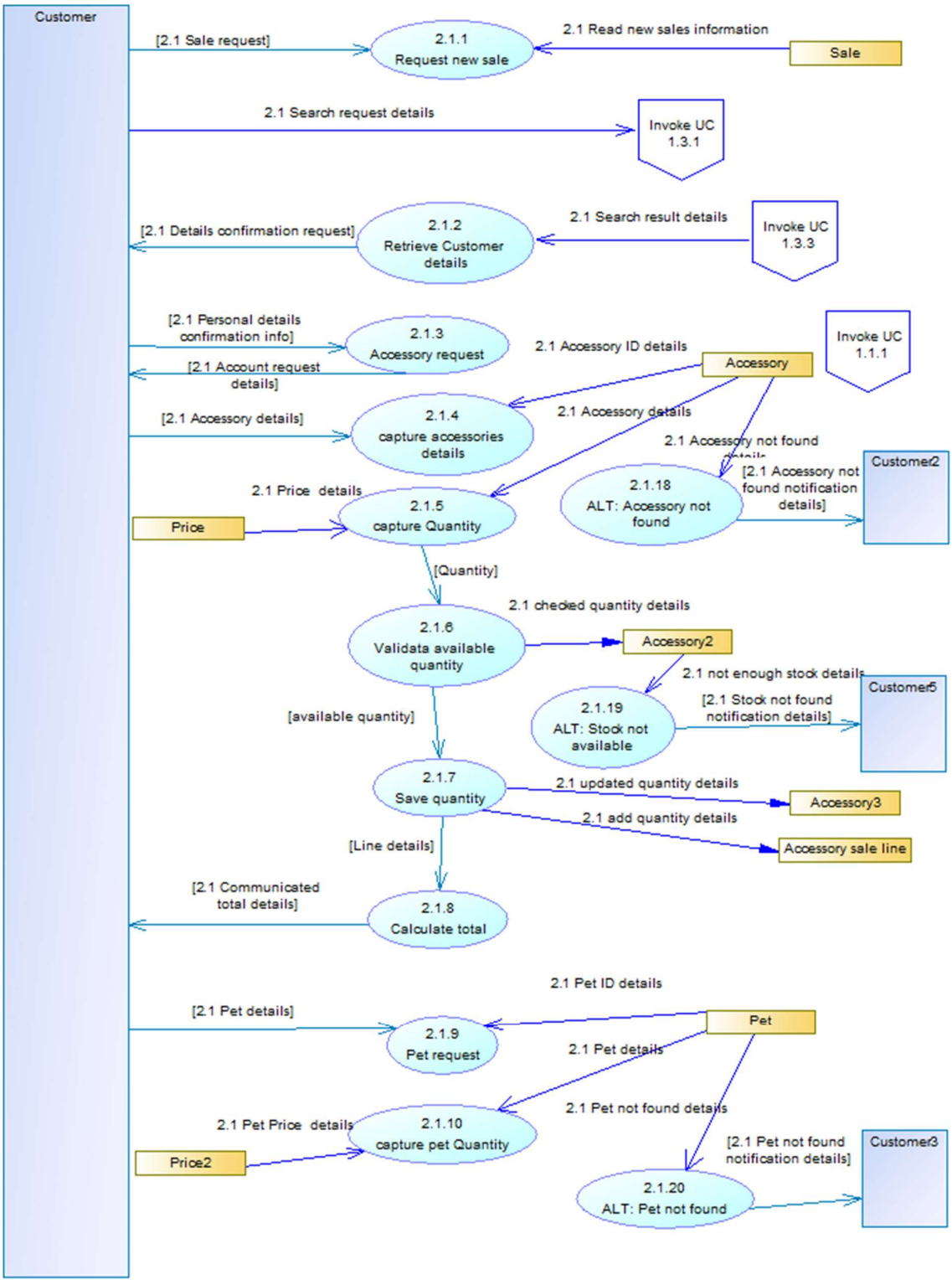
Bootcamp day 2 Memo

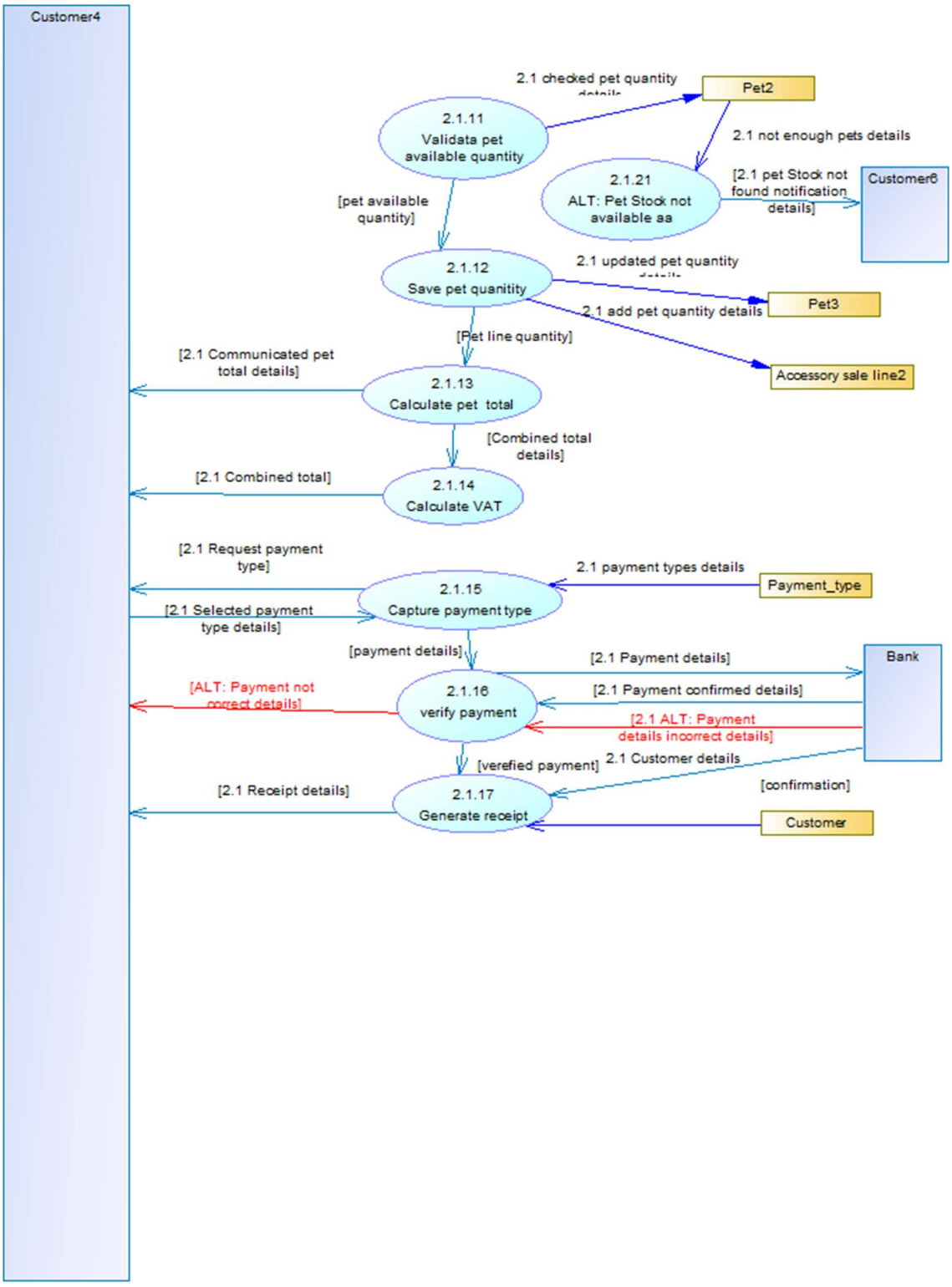
System Name: Soapy Suds		
Author: Your Name	Date: 2022/27/07	Version: 1.0.0

Use case name:	Add Customer	**USE CASE TYPE**	
Use case id:	1.1	Business Requirements:	☐
Priority:	High	System Analysis:	☒
Source:	Case Study provided	System Design:	☐
Primary business actor	Customer		
Primary system actor	Employee		
Other participating actors:	None		
Other interested stakeholders:	None		
Description:	This use case describes the event of an employee adding a new customer to the system. The customer provides his/her details and the employee captures the details on the system. This use case concludes when the system displays a confirmation message that the customer has successfully been added.		
Pre-condition:	The employee must be logged into the system.		
Trigger:	A new customer requests to be added to the system.		
Typical course Of events:	**Actor Action**	**System Response**	
		Manual Action	**Automated Action**
	Step 1: Customer requests to be added to the system.	**Step 2**: The Cashier navigates to the 'Create Customer' Screen.	**Step 3**: The system loads the 'Create Customer' screen using MVC. The screen contains the following: 'Register Customer' heading. The following labels: ● Name ● Surname ● Contact Number ● Email Address ● Postal Address

			● Delivery Address For each label, there is a corresponding text box containing example input. Two buttons are labelled 'Submit' and 'Cancel'. The system prompts the user to enter the customer details.
		Step 4: The cashier requests the customer's details.	
	Step 5: The customer provides the following information: ● Name ● Surname ● Contact Number ● Email Address ● Postal Address ● Delivery Address	**Step 6**: The Cashier inputs the details and clicks on the 'Submit' button.	**Step 7**: The system captures the customer details and submissions and validates the information by using SQL_Read to check if the customer doesn't already exists in the Customer entity. [ALT]
			Step 8: The system retrieves the last Customer_ID from the Customer entity using SQL_Read. The system increases the Customer_ID by one and stores the new customer record with the ● Name ● Surname ● Contact_Number ● Email_Address ● Physical_Address ● Delivery_Address using SQL_Write. The system displays the 'Success Notification' stating "The customer has successfully been added to the system"

			with a button labelled 'OK'.
		Step 9: The cashier informs the customer that they have been registered successfully.	
Alternate courses:	**Alt Step 7a**: The system verifies that all fields are completed and if not displays the 'Error Notification' stating "ERROR: All fields need to be completed!" with a button labelled 'OK'.		
	Alt Step 7b: The system verifies the customer already exists in the Customer entity using SQL_Read and displays the 'Verification Notification' stating "ERROR: Customer already in the system!" with a button labelled 'OK'.		
Conclusion:	A new customer is added to the Customer table.		
Post-condition:	A new customer was added to the system. The customer received his customer number.		
Business Rules:	None		
Implementation, Constraints And Specifications	None		
Assumptions:	None		
Open Issues:	None		


Customer
[2.1 Sale request]
2.1.1
Request new sale
2.1 Read new sales information
Sale
2.1 Search request details
Invoke UC
1.3.1
[2.1 Details confirmation request]
2.1.2
Retrieve Customer
details
2.1 Search result details
Invoke UC
1.3.3
[2.1 Personal details
confirmation info]
2.1.3
Accessory request
[2.1 Account request
details]
2.1 Accessory ID details
Accessory
Invoke UC
1.1.1
[2.1 Accessory details]
2.1.4
capture accessories
details
2.1 Accessory details
2.1 Accessory not found
2.1.18
ALT: Accessory not
found
[2.1 Accessory not
found notification
details]
Customer2
2.1 Price details
Price
2.1.5
capture Quantity
[Quantity]
2.1 checked quantity details
2.1.6
Validata available
quantity
Accessory2
2.1 not enough stock details
[available quantity]
2.1.19
ALT: Stock not
available
[2.1 Stock not found
notification details]
Customer5
2.1.7
Save quantity
2.1 updated quantity details
Accessory3
2.1 add quantity details
Accessory sale line
[Line details]
[2.1 Communicated
total details]
2.1.8
Calculate total
2.1 Pet ID details
[2.1 Pet details]
2.1.9
Pet request
Pet
2.1 Pet details
2.1 Pet not found details
2.1 Pet Price details
Price2
2.1.10
capture pet Quantity
[2.1 Pet not found
notification details]
Customer3
2.1.20
ALT: Pet not found


Customer4
2.1 checked pet quantity
Pet2
2.1.11
Validata pet
available quantity
2.1 not enough pets details
2.1.21
ALT: Pet Stock not
available aa
[2.1 pet Stock not
found notification
details]
Customer6
[pet available
quantity]
2.1.12
Save pet quanitity
2.1 updated pet quantity
Pet3
2.1 add pet quantity details
[Pet line quantity]
Accessory sale line2
[2.1 Communicated pet
total details]
2.1.13
Calculate pet total
[Combined total
details]
[2.1 Combined total]
2.1.14
Calculate VAT
[2.1 Request payment
type]
2.1 payment types details
Payment_type
2.1.15
Capture payment type
[2.1 Selected payment
type details]
[payment details]
[2.1 Payment details]
Bank
[ALT: Payment not
correct details]
2.1.16
verify payment
[2.1 Payment confirmed details]
[2.1 ALT: Payment
details incorrect details]
[verefied payment]
2.1 Customer details
[confirmation]
[2.1 Receipt details]
2.1.17
Generate receipt
Customer