

TRACEABILITY INFORMATION FOR USE CASES

Coffee Carts Rewards Management System Android App

TABLE OF CONTENTS

PURPOSE

TRACEABILITY INFORMATION FOR ADDING USER TO COFFEE CART REWARDS MANAGEMENT SYSTEM ANDROID APP:

- [Business requirement](#)
- [Use case](#)
- [Design model](#)
 - [Server side](#)
 - [Client side](#)
- [Code](#)
- [Test cases](#)
 - [Add VIP Server](#)
 - [Add VIP Client](#)

TRACEABILITY INFORMATION FOR RECORD VIP PURCHASE TO COFFEE CART REWARDS MANAGEMENT SYSTEM ANDROID APP:

- [Business requirement](#)
- [Use case](#)
- [Design model](#)
 - [Server side](#)
 - [Client side](#)
- [Code](#)
- [Test cases](#)
 - [Record VIP Purchase Server](#)
 - [Record VIP Purchase Client](#)

PURPOSE:

This document aims at showing the relationships between Business Requirements, Use cases, System architecture, Design specification and Test cases for the Coffee Carts Rewards Management System Android App.

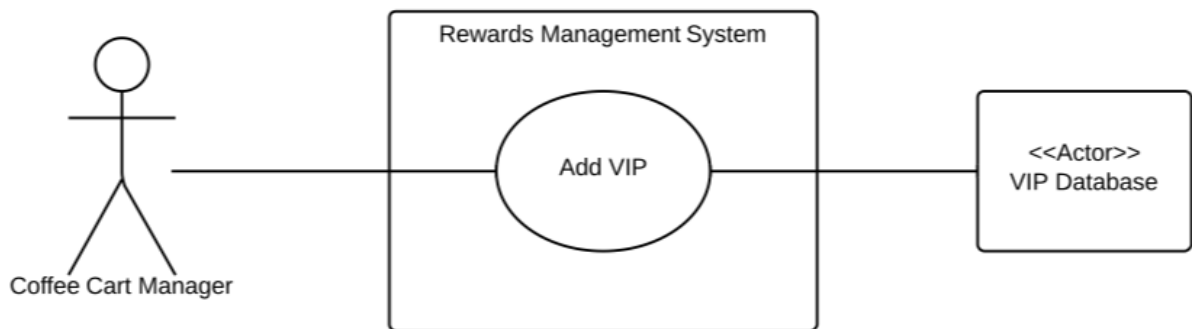
TRACEABILITY INFORMATION FOR ADDING USER TO COFFEE CART REWARDS MANAGEMENT SYSTEM ANDROID APP:

1. Business requirement

As per the Project Specifications Document, the coffee cart manager must be able to add VIP customers to the system by knowing their name, phone number and birth date. A unique VIP card number is assigned to each VIP customer. Related requirements are 1, 2, and 3.

2. Use case

The Use case corresponding to the above requirement is Add VIP documented in the Use Case Model document. The details of this use case are elaborated below:



Use Case Name:	Add VIP
Related Requirements:	1, 2, and 3
Goal:	Allow new VIP accounts to be created for customers.
Precondition:	None
Successful End Condition:	A new VIP account with a unique card number is created for the customer.

Failed End Condition:	No VIP account is created.
Primary Actors:	Coffee Cart Manager
Secondary Actors:	VIP Database
Trigger:	The Coffee Cart Manager starts the VIP account creation process for a customer.

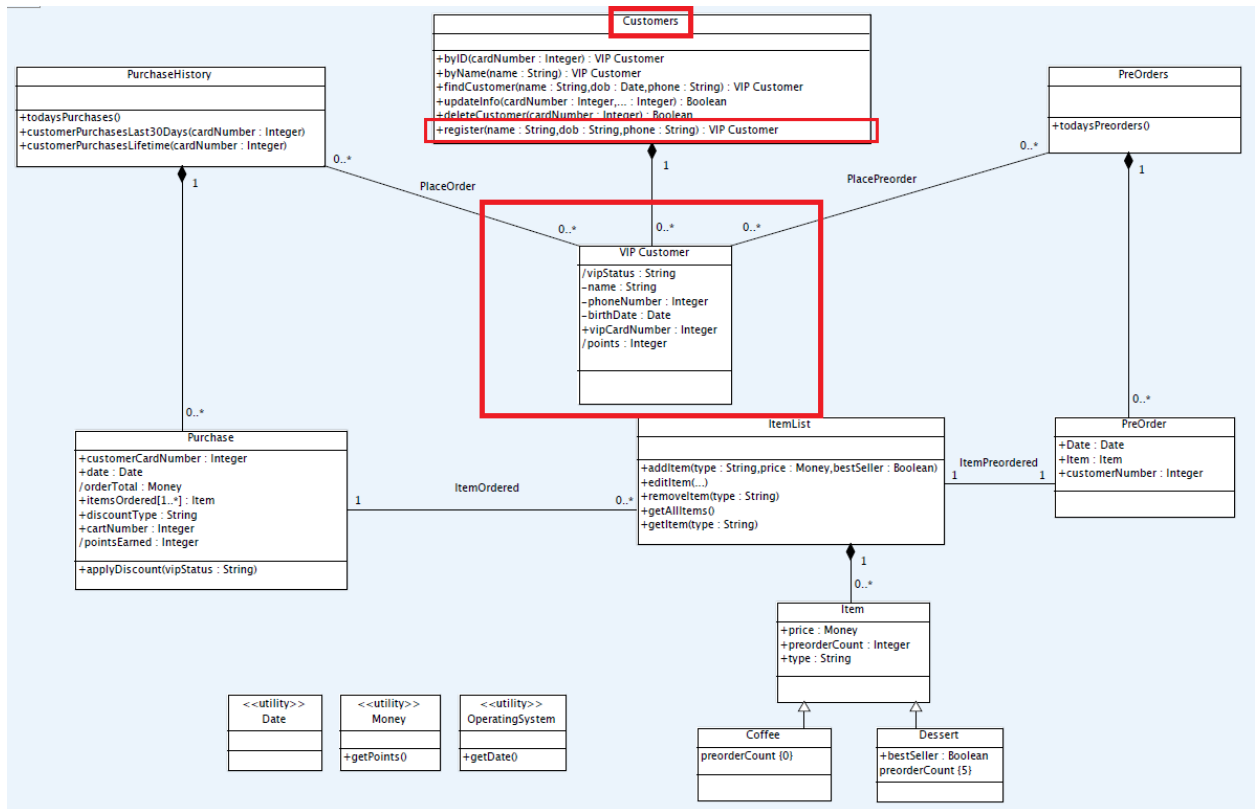
Main Flow	Step	Action
	1	The Coffee Cart Manager starts the VIP account creation process for a customer.
	2	The Coffee Cart Manager enters in the customer's information.
	3	The Coffee Cart Manager confirms the customer's information and submits it.
	4	All required customer information is verified as being present and valid.
	5	The customer information is sent to the VIP Database.
	6	If the customer's name, date of birth, and phone number are not altogether already in the system, the VIP Database creates a new VIP account and assigns a VIP card number to the customer.
	7	The VIP Database replies with a success message and the unique VIP card number.
	8	An information message confirming the successful creation of the account along with the unique VIP card number is displayed.
	9	The Coffee Cart Manager creates a VIP card with a unique VIP card number and gives it to the customer.
Extensions	Step	Branching Action
	4.1	Some required customer information is missing or is invalid.
	4.2	An error message is displayed to the Coffee Cart Manager.

	6.1	The VIP Database experiences an error trying to create a new VIP account, possibly because the name, phone number, and date of birth are all already in the system as a group.
	6.2	The VIP Database replies with an error message.
	6.3	The error message is displayed to the Coffee Cart Manager.
Other Flows	(1)	Cancellation: Anytime after starting the account creation process and before any customer information is sent to the VIP Database, the Coffee Cart Manager may cancel the process.
	(2)	Timeout: Anytime a request is made to the VIP Database, if a reply is not received in a reasonable period of time, a timeout error message is displayed to the Coffee Cart Manager.

3. Design model

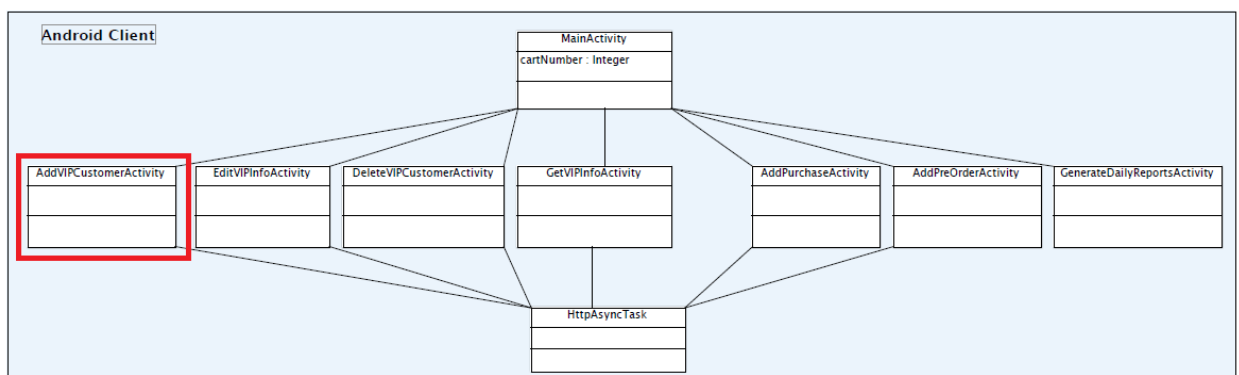
Server side

On the server side, VIP Customer class stores name of the Customer, Phone number, Date of birth and VIP Card number assigned to the user in the Coffee Carts Rewards Management System. In Customers class, *register* method is invoked to save this information in the object of VIP Customer class.



Client side

However, on the client side, *MainActivity* has a button “Add VIP”, which invokes *AddVIPCustomerActivity*. Coffee Cart Manager inputs the information through UI and information is passed on to server side class for storing in the database.



4. Code

The client side code is in *AddVipActivity.java* in CoffeeCartApp/src/ folder under gatech.cs6300.project2 package and corresponding layout is in *fragment_add_vip.xml* file in CoffeeCartApp/res/layout folder.

The server side code is in CoffeeCartApp/CoffeeCartServer. Look specifically in *main.py* and *database.py*.

5. Test cases

The test cases corresponding to Add VIP Customer are present in *TestPlan.md* file in CoffeeCartApp/Deliverable2 folder.

The server side test cases can be found below. Please note that source code for these test cases may be obtained by following link below.

[Add VIP Server](#)

- Happy path
- Missing name
- Missing DOB
- Missing phone number
- Missing all fields
- Invalid DOB
- DOB in future
- Duplicate

The client side test cases are listed below. Please note that source code for these test cases may be obtained by following link below.

[Add VIP Client](#)

- Happy path
- Cancel - Confirmed
- Cancel - Not Confirmed
- Missing name
- Missing DOB
- Missing phone number
- Missing all fields
- Invalid DOB
- Duplicate
- Server communication error

TRACEABILITY INFORMATION FOR RECORD VIP PURCHASE TO COFFEE CART REWARDS MANAGEMENT SYSTEM ANDROID APP:

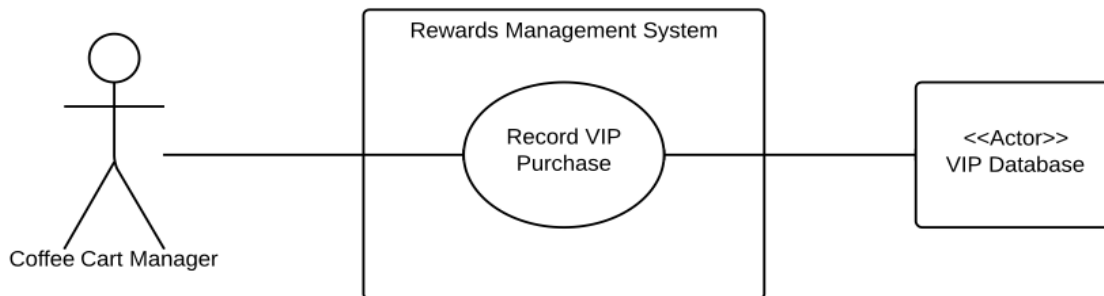
1. Business requirement

As per the Project Specifications Document, the coffee cart manager must be able to record purchases made by VIP customers. Related requirements are 5, 6, 9, 10, 11, and 12

2. Use case

The Use case corresponding to the above requirement is Record VIP Purchase documented in the Use Case Model document. The details of this use case are elaborated below:

Use Case: Record VIP Purchase



Use Case Name:	Record VIP Purchase
Related Requirements:	5, 6, 9, 10, 11, and 12
Goal:	Record items and points associated with a VIP Purchase.
Precondition:	The VIP card number of the customer must be known.
Successful End Condition:	The items and points associated with the purchase are recorded and the running total of all points earned by the customer is updated.
Failed End Condition:	No record of the purchase is made.
Primary Actors:	Coffee Cart Manager
Secondary Actors:	VIP Database

Trigger:	The Coffee Cart Manager looks up the customer using their VIP card number.
----------	--

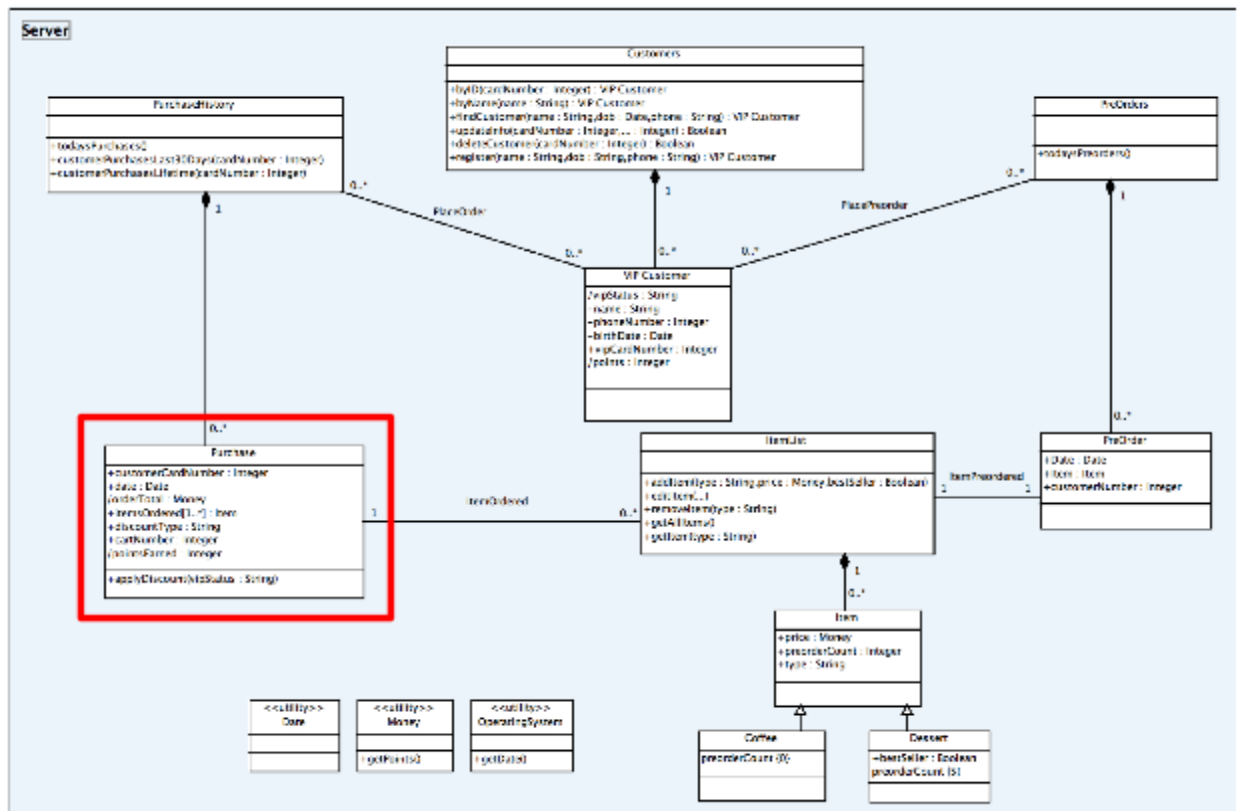
Main Flow	Step	Action
	1	The Coffee Cart Manager looks up the customer using their VIP card number.
	2	The VIP card number is found in the VIP Database.
	3	The customer's information, including VIP points earned and VIP status, is displayed to the Coffee Cart Manager.
	4	The Coffee Cart Manager asks to record a purchase to the customer's account.
	5	A new screen allowing a purchase to be recorded is displayed.
	6	The Coffee Cart Manager enters the items to be purchased and their quantities.
	7	The Coffee Cart Manager submits the purchase information.
	8	The VIP Database is used to help calculate the customer's VIP status and determine whether any discounts apply.
	9	The discounted price is displayed to the Coffee Cart Manager and they are asked if they would like to continue to record the purchase.
	10	The Coffee Cart Manager charges the customer the discounted price and indicates that the purchase should be recorded.
	11	The number of VIP points to be awarded is calculated and added to the purchase information.
	12	The purchase information is sent to the VIP Database, which records it and updates the total number of VIP points awarded to the customer.
	13	An information message indicating that the purchase was successfully recorded is displayed.
Extensions	Step	Branching Action
	2.1	The VIP card number is not found in the VIP Database.

	2.2	The Coffee Cart Manager is informed that the VIP card number was not found.
	10.1	The Coffee Cart Manager declines to continue recording the purchase.
	10.2	No purchase is made and the screen allowing purchase information to be entered is displayed with the values previously submitted.
	12.1	The VIP Database fails to record the purchase and replies with an error message.
	12.2	The error message is displayed to the Coffee Cart Manager.
Other Flows	(1)	Cancellation: Anytime after entering the screen to record purchase information and before submitting the purchase, the Coffee Cart Manager may back out of their changes by cancelling the edit process.
	(2)	Timeout: Anytime a request is made to the VIP Database, if a reply is not received in a reasonable period of time, a timeout error message is displayed to the Coffee Cart Manager.

3. Design model

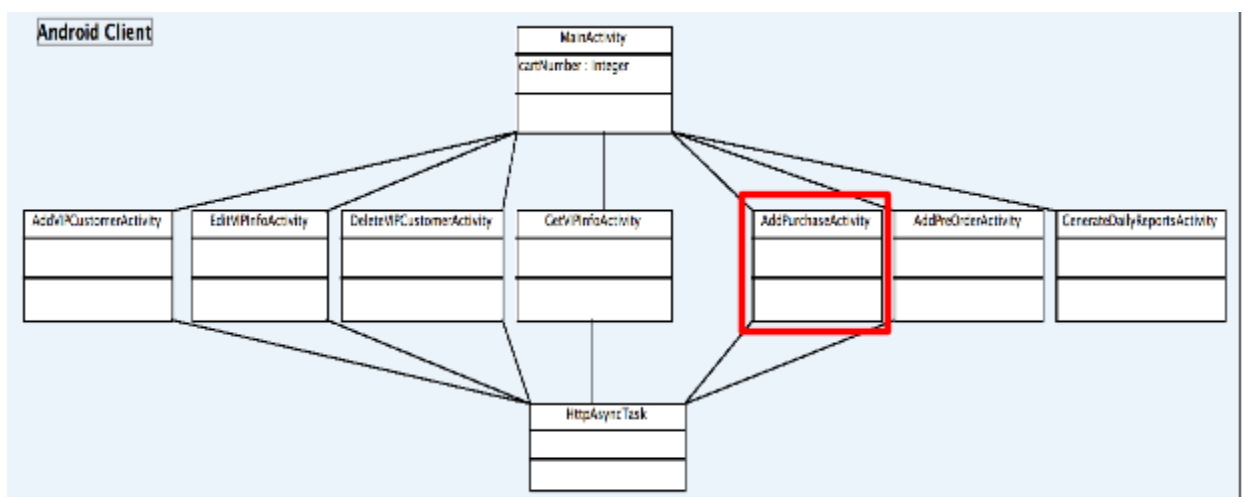
Server side

On the server side, the Purchase class stores the card number of the customer, date of purchase, total cost of purchase, points earned, the items ordered, discount information and the cart number. Of these items, only the card number, items ordered, and cart number are passed into the server; the rest of this data is determined by the server.



Client side

On the client side, a PurchaseActivity class is used where the Coffee Cart Manager inputs the information about the purchase through UI and information is passed on to server side class for storing in the database.



6. Code

The client side code is in *PurchaseActivity.java* in CoffeeCartApp/src/ folder under gatech.cs6300.project2 package and corresponding layout is in *activity_purchase.xml* file in CoffeeCartApp/res/layout folder.

The server side code is in CoffeeCartApp/CoffeeCartServer. Look specifically in *main.py* and *database.py*.

7. Test cases

The test cases corresponding to Record VIP Purch are present in *TestPlan.md* file in CoffeeCartApp/Deliverable2 folder.

The server side test cases can be found below. Please note that source code for these test cases may be obtained by following this link ([Record VIP Purchase Server](#))

[Record VIP Purchase Server](#)

- Happy path - not gold
- Happy path - has gold
- Happy path - becomes gold
- Happy path - with preorders
- Happy path - with preorders (deferred)
- Empty purchase
- Zero quantity with other purchases
- Zero quantity only
- Invalid quantity
- Invalid item
- Missing VIP ID
- VIP customer not found

The client side test cases are listed below. Please note that very detailed descriptions of these test cases may be obtained by following this link ([Record VIP Purchase Client](#))

[Record VIP Purchase Client](#)

- Happy path - not gold
- Happy path - has gold
- Happy path - becomes gold
- Happy path - with preorders
- Happy path - with preorders (deferred)
- Empty purchase
- Zero quantity with other purchases
- Zero quantity only
- Invalid quantity

- Invalid item
- Missing VIP ID
- VIP customer not found