

# Currency Exchange

Created by

Oleg Arusoo

Ulvi Balashov

Yuliia Sidelnik

## **Problem Statement:**

Design a basic currency exchange system that allows a cashier to:

- Convert money between currencies
- Issue receipts for transactions
- Handle partial exchanges and denomination requests
- Track currency reserves and notify management if reserves are low
- Generate daily summary reports
- Calculate and apply cashier bonuses based on profit

## User Stories:

1. As a client, I want to exchange USD to EUR.
2. As a cashier, I want to check if enough EUR is available before confirming the transaction.
3. As a client, I want a printed receipt showing the exchange details.
4. As a cashier, I want to perform partial exchanges (e.g., 50 EUR and rest in local currency).

## User Stories:

- 5. As management, I want to receive a daily report of currency balances and profit.
- 6. As a cashier, I want to be notified when a currency reserve is low.
- 7. As management, I want to award a 5% bonus to the cashier based on daily profit.

## Functional Requirements:

- The system must allow currency conversion between supported currencies.
- The system must verify reserve availability before exchange.
- The system must generate a receipt after each transaction.
- The system must support partial and denomination-based exchanges.

## Functional Requirements:

- The system must generate a daily report with balances and profit.
- The system must alert when reserves fall below a critical minimum.
- The system must calculate and apply cashier bonuses

## **Non-Functional Requirements:**

- The system should respond to exchange requests within 1 second.
- The system should store transaction history securely.
- The system should be modular and easy to extend.
- The system should run on standard C++ compilers.

## Overview of CRC cards

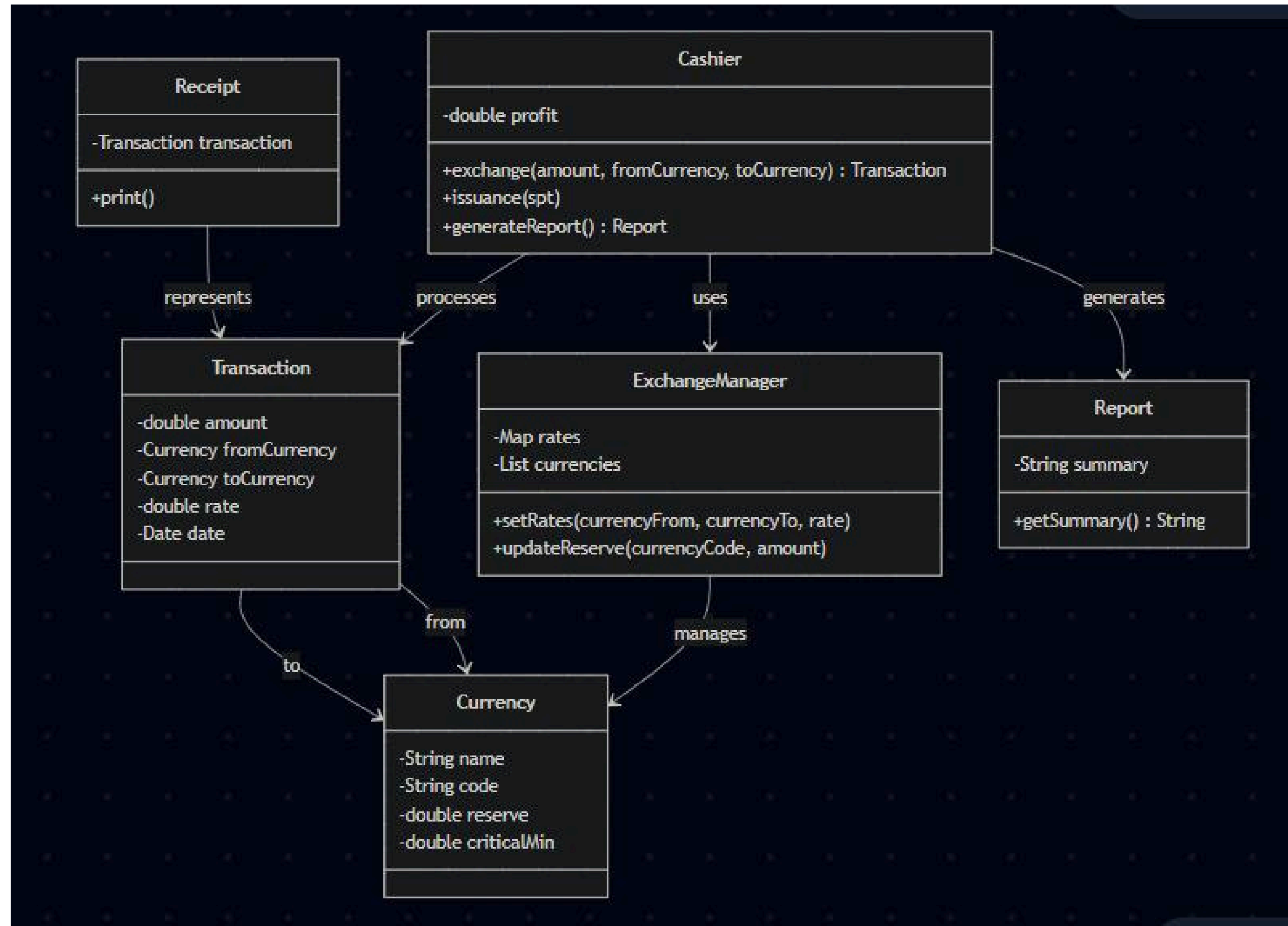
Class:	Responsibilities:	Collaborators:
Cashier	<ul style="list-style-type: none"><li>• perform exchange</li><li>• print receipt</li><li>• generate report</li></ul>	<ul style="list-style-type: none"><li>• Transaction</li><li>• Receipt</li><li>• ExchangeManager</li></ul>
Transaction	<ul style="list-style-type: none"><li>• store exchange details (amount, currencies, rate, date)</li></ul>	<ul style="list-style-type: none"><li>• Cashier</li><li>• Receipt</li></ul>



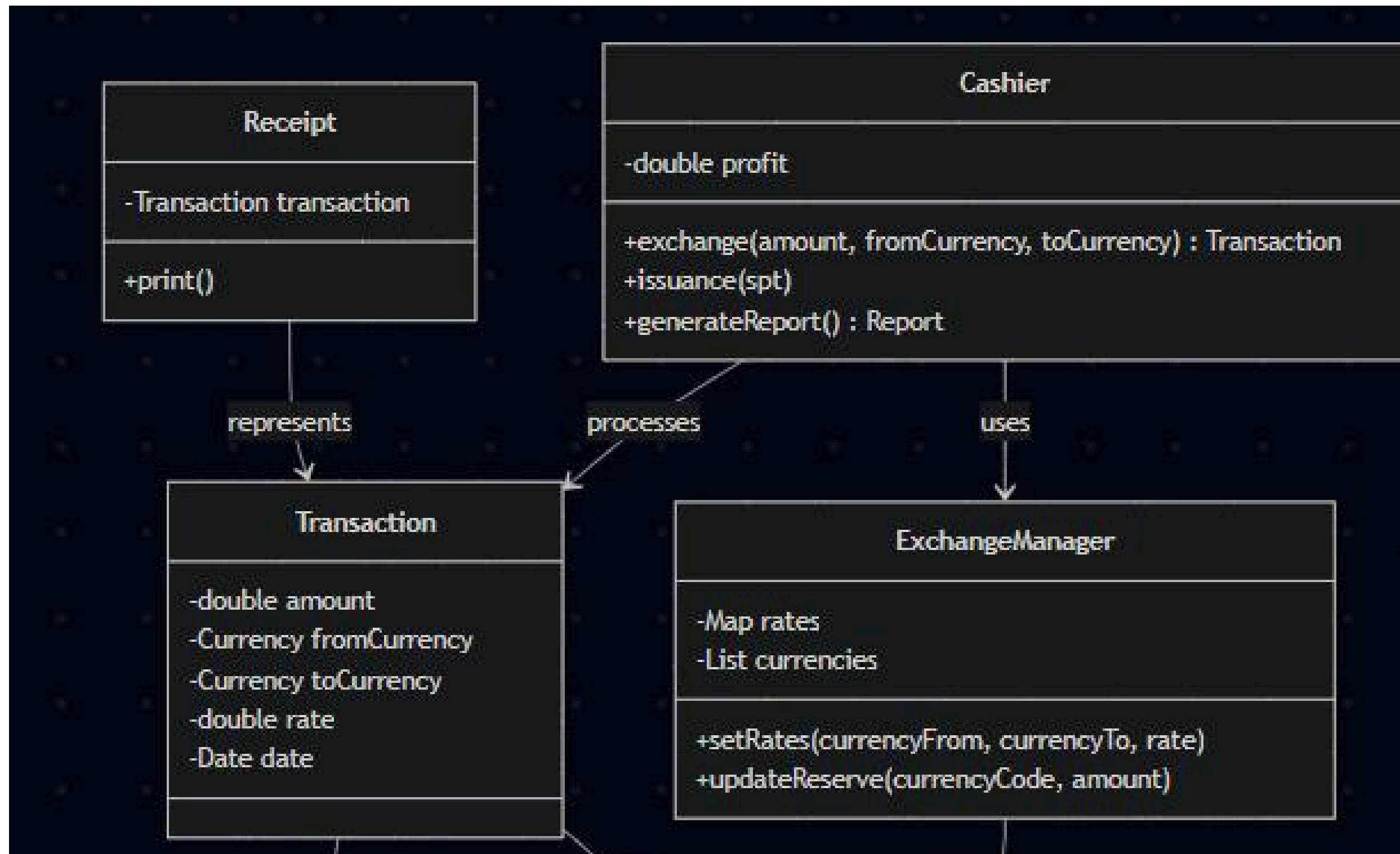
## Overview of CRC cards

Class:	Responsibilities:	Collaborators:
Receipt	<ul style="list-style-type: none"><li>• format and print transaction receipt</li></ul>	<ul style="list-style-type: none"><li>• Transaction</li></ul>
ExchangeManager	<ul style="list-style-type: none"><li>• set rates</li><li>• manage reserves</li><li>• calculate bonus</li></ul>	<ul style="list-style-type: none"><li>• Currency</li><li>• Cashier</li></ul>

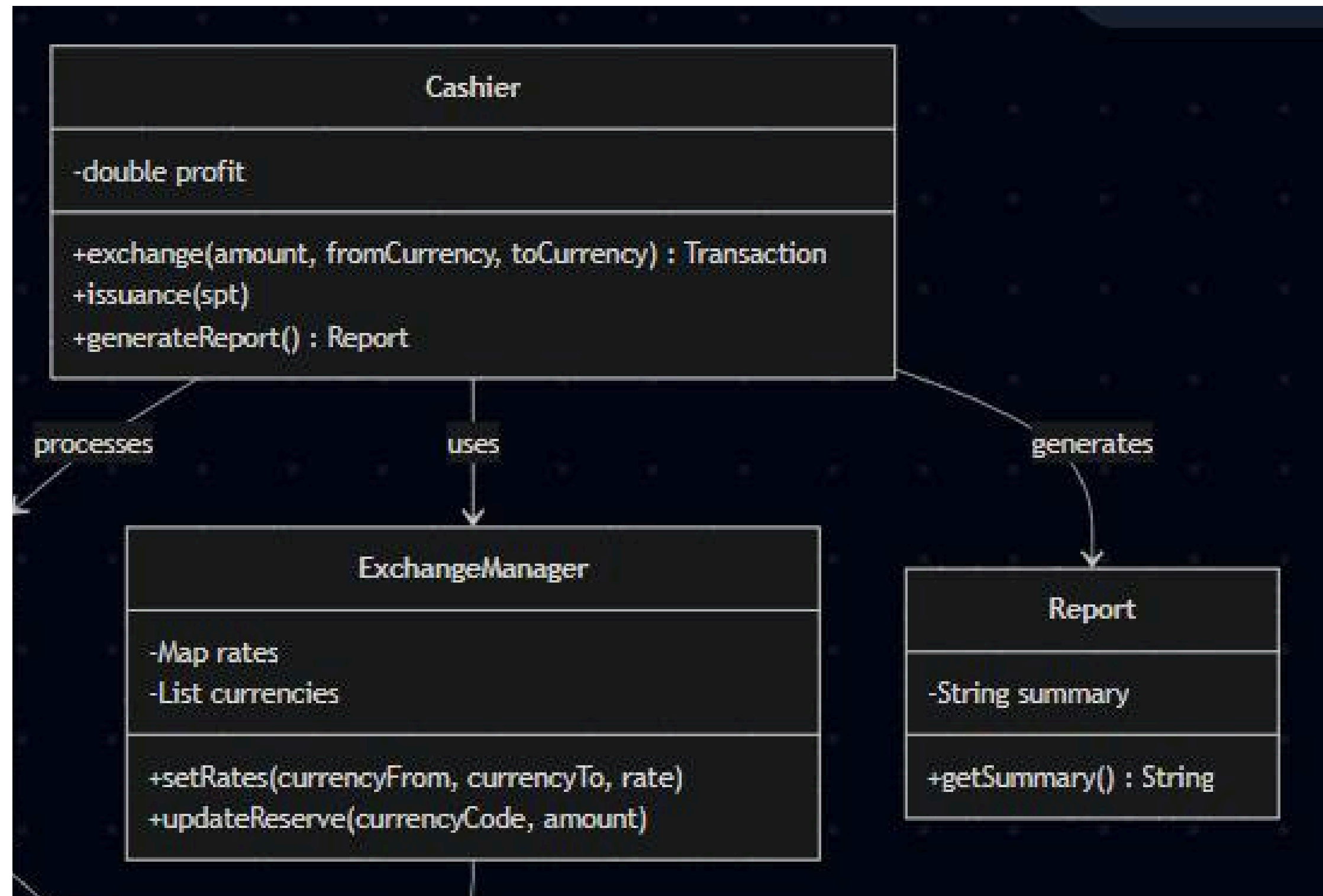
# Conceptual Simple Class Diagram



# Conceptual Simple Class Diagram



# Conceptual Simple Class Diagram



# Conceptual Simple Class Diagram



# Traceability table

User Story	CRC Responsibility	Simple Class Diagram Element	C++ Element
Client exchanges currency	Cashier: perform exchange	Cashier — Transaction	<code>Cashier::exchange(amount, fromCurrency, toCurrency)</code>
Cashier issues receipt	Cashier: issue receipt	Cashier — Receipt	<code>Cashier::issuance(spt)</code>
Cashier generates report	Cashier: generate report	Cashier — Report	<code>Cashier::generateReport()</code>
Update exchange rates	ExchangeManager: set rates	ExchangeManager — Currency	<code>ExchangeManager::setRates(currencyFrom, currencyTo, rate)</code>
Update currency reserves	ExchangeManager: update reserves	ExchangeManager — Currency	<code>ExchangeManager::updateReserve(currencyCode, amount)</code>
Print receipt	Receipt: print receipt	Receipt — Transaction	<code>Receipt::print()</code>
View transaction details	Transaction: store data	Transaction — Currency	(attributes: <code>amount</code> , <code>currencyFrom</code> , <code>currencyTo</code> , <code>date</code> )
View exchange summary	Report: get summary	Report	<code>Report::getSummary()</code>

# Team roles
