# Kaushal Kishore Rai

Atlantis University

MIT 563: Programming and Applications Development

Fall “A” 2024

Professor: Alex Lima

09/16/2024

**Customer Service module and integration with HR module –**

**A diagram of customer service

Description automatically generated**

# -\*- coding: utf-8 -\*-

"""CustomerService

This module manages employees in the customer-service department and tracks customer service metrics such as service quality, number of clients served, average service time, and customer satisfaction.

"""

class CustomerServiceEmployee:

    def \_\_init\_\_(self, emp\_id, name, role):

        self.emp\_id = emp\_id

        self.name = name

        self.role = role

        self.clients\_served = 0

        self.total\_service\_time = 0

        self.customer\_satisfaction = []

        self.service\_area = ""

    def serve\_client(self, service\_time, satisfaction\_level):

        self.clients\_served += 1

        self.total\_service\_time += service\_time

        self.customer\_satisfaction.append(satisfaction\_level)

    def set\_service\_area(self, service\_area):

        self.service\_area = service\_area

    def calculate\_average\_service\_time(self):

        return self.total\_service\_time / self.clients\_served if self.clients\_served > 0 else 0

    def calculate\_average\_satisfaction(self):

        return sum(self.customer\_satisfaction) / len(self.customer\_satisfaction) if self.customer\_satisfaction else 0

    def display\_customer\_service\_info(self):

        print(f"Customer Service Employee ID: {self.emp\_id}")

        print(f"Name: {self.name}")

        print(f"Role: {self.role}")

        print(f"Clients Served: {self.clients\_served}")

        print(f"Average Service Time: {self.calculate\_average\_service\_time()} minutes")

        print(f"Average Customer Satisfaction Level: {self.calculate\_average\_satisfaction()}")

        print(f"Service Area: {self.service\_area}")

        print("\n")

class CustomerServiceManager:

    def \_\_init\_\_(self):

        self.cs\_employees = {}

    def add\_cs\_employee(self, emp\_id, name, role):

        self.cs\_employees[emp\_id] = CustomerServiceEmployee(emp\_id, name, role)

    def log\_service(self, emp\_id, service\_time, satisfaction\_level):

        if emp\_id in self.cs\_employees:

            self.cs\_employees[emp\_id].serve\_client(service\_time, satisfaction\_level)

    def set\_service\_area(self, emp\_id, service\_area):

        if emp\_id in self.cs\_employees:

            self.cs\_employees[emp\_id].set\_service\_area(service\_area)

    def display\_cs\_employee\_info(self, emp\_id):

        if emp\_id in self.cs\_employees:

            self.cs\_employees[emp\_id].display\_customer\_service\_info()

# Example usage

if \_\_name\_\_ == "\_\_main\_\_":

    cs\_manager = CustomerServiceManager()

    # Add customer service employees

    cs\_manager.add\_cs\_employee(1, "Chris Brown", "Customer Service Rep")

    cs\_manager.add\_cs\_employee(2, "Emma Davis", "Customer Support Specialist")

    # Log service information

    cs\_manager.log\_service(1, 15, 4.5)  # 15 minutes call with 4.5 satisfaction

    cs\_manager.log\_service(1, 10, 4.0)  # 10 minutes call with 4.0 satisfaction

    cs\_manager.log\_service(2, 20, 5.0)  # 20 minutes call with 5.0 satisfaction

    # Set service area

    cs\_manager.set\_service\_area(1, "Product Support")

    cs\_manager.set\_service\_area(2, "Technical Assistance")

    # Display customer service info for employees

    cs\_manager.display\_cs\_employee\_info(1)

    cs\_manager.display\_cs\_employee\_info(2)

**Document: Integration of Customer Service Module and Human Resources in**

**Overview**

This document describes the integration of the Human Resources (HR) module with the Customer Service (CS) module. The integrated system tracks employee management and specific customer service metrics, demonstrating how the two systems collaborate to provide a complete overview of an organization's HR and customer service operations. The program ensures that HR employees belonging to the customer service department can be tracked for their customer-facing performance metrics, while other employees are managed through general HR operations.

**System Architecture**

1. **HR Module**:
   * Handles employee details like ID, name, role, hours worked, overtime, unavailable hours, training hours, promotion status, and vacation status.
   * Provides functionality to log these details for all employees across the organization.
2. **Customer Service Module**:
   * Extends HR functionality specifically for customer service employees.
   * Tracks clients served, average service time, customer satisfaction, and the specific service area.
3. **Integration**:
   * The integration ensures that employees added to the HR module who belong to the customer service department are also tracked in the CS module.
   * HR manages employee attributes such as hours worked, promotions, etc., while CS tracks customer service performance metrics.

**Integration Details**

* The CustomerServiceManager class interacts with employees managed by the HRManager class.
* When an employee is hired or added to the HR system, they can be registered in the CustomerServiceManager if they belong to the customer service department.
* Variables like employee ID, name, and role are common between both modules. HR focuses on general employee management (hours worked, promotions), and the CS module focuses on client interaction metrics (clients served, service quality, etc.).

**Code Implementation**

python

# Human Resource Module (HR)

class Employee:

def \_\_init\_\_(self, emp\_id, name, role):

self.emp\_id = emp\_id

self.name = name

self.role = role

self.hours\_worked = 0

self.overtime\_hours = 0

self.unavailable\_hours = 0

self.training\_hours = 0

self.promoted = False

self.on\_vacation = False

def work(self, hours):

self.hours\_worked += hours

def log\_overtime(self, hours):

self.overtime\_hours += hours

def mark\_unavailable(self, hours):

self.unavailable\_hours += hours

def log\_training(self, hours):

self.training\_hours += hours

def promote(self):

self.promoted = True

def start\_vacation(self):

self.on\_vacation = True

def end\_vacation(self):

self.on\_vacation = False

class HRManager:

def \_\_init\_\_(self):

self.employees = {}

def add\_employee(self, emp\_id, name, role):

self.employees[emp\_id] = Employee(emp\_id, name, role)

def log\_hours(self, emp\_id, hours):

if emp\_id in self.employees:

self.employees[emp\_id].work(hours)

def promote\_employee(self, emp\_id):

if emp\_id in self.employees:

self.employees[emp\_id].promote()

# Customer Service Module (CS)

class CustomerServiceEmployee(Employee):

def \_\_init\_\_(self, emp\_id, name, role):

super().\_\_init\_\_(emp\_id, name, role)

self.clients\_served = 0

self.total\_service\_time = 0

self.customer\_satisfaction = []

self.service\_area = ""

def serve\_client(self, service\_time, satisfaction\_level):

self.clients\_served += 1

self.total\_service\_time += service\_time

self.customer\_satisfaction.append(satisfaction\_level)

def set\_service\_area(self, service\_area):

self.service\_area = service\_area

def calculate\_average\_service\_time(self):

return self.total\_service\_time / self.clients\_served if self.clients\_served > 0 else 0

def calculate\_average\_satisfaction(self):

return sum(self.customer\_satisfaction) / len(self.customer\_satisfaction) if self.customer\_satisfaction else 0

def display\_customer\_service\_info(self):

print(f"Employee ID: {self.emp\_id}")

print(f"Name: {self.name}")

print(f"Role: {self.role}")

print(f"Clients Served: {self.clients\_served}")

print(f"Average Service Time: {self.calculate\_average\_service\_time()} minutes")

print(f"Average Customer Satisfaction Level: {self.calculate\_average\_satisfaction()}")

print(f"Service Area: {self.service\_area}")

print("\n")

class CustomerServiceManager:

def \_\_init\_\_(self, hr\_manager):

self.cs\_employees = {}

self.hr\_manager = hr\_manager

def add\_cs\_employee(self, emp\_id):

if emp\_id in self.hr\_manager.employees:

emp = self.hr\_manager.employees[emp\_id]

self.cs\_employees[emp\_id] = CustomerServiceEmployee(emp.emp\_id, emp.name, emp.role)

def log\_service(self, emp\_id, service\_time, satisfaction\_level):

if emp\_id in self.cs\_employees:

self.cs\_employees[emp\_id].serve\_client(service\_time, satisfaction\_level)

def set\_service\_area(self, emp\_id, service\_area):

if emp\_id in self.cs\_employees:

self.cs\_employees[emp\_id].set\_service\_area(service\_area)

def display\_cs\_employee\_info(self, emp\_id):

if emp\_id in self.cs\_employees:

self.cs\_employees[emp\_id].display\_customer\_service\_info()

# Integration Example

if \_\_name\_\_ == "\_\_main\_\_":

hr\_manager = HRManager()

# Adding employees

hr\_manager.add\_employee(1, "Chris Brown", "Customer Service Rep")

hr\_manager.add\_employee(2, "Emma Davis", "Technical Support Specialist")

hr\_manager.add\_employee(3, "John Doe", "Developer")

# Log hours worked in HR

hr\_manager.log\_hours(1, 40)

hr\_manager.log\_hours(2, 35)

hr\_manager.promote\_employee(1)

# Customer service manager setup

cs\_manager = CustomerServiceManager(hr\_manager)

# Add CS-specific employees

cs\_manager.add\_cs\_employee(1)

cs\_manager.add\_cs\_employee(2)

# Log customer service data

cs\_manager.log\_service(1, 20, 4.5) # 20 minutes call, 4.5 satisfaction

cs\_manager.log\_service(1, 15, 4.0)

cs\_manager.log\_service(2, 30, 5.0)

# Set service areas

cs\_manager.set\_service\_area(1, "Product Support")

cs\_manager.set\_service\_area(2, "Technical Assistance")

# Display CS employees' info

cs\_manager.display\_cs\_employee\_info(1)

cs\_manager.display\_cs\_employee\_info(2)

**How Integration Works:**

1. **Shared Employee Data**: Employee details such as emp\_id, name, and role are shared between HR and CS systems.
2. **Customer Service Specifics**: Employees in customer service are extended from the HR employee class and have additional metrics like clients\_served, service\_time, and satisfaction.
3. **Automatic Addition to Customer Service**: The CustomerServiceManager class adds employees to customer service only if they are managed by the HR system.

**Results Produced by the Program:**

plaintext

Employee ID: 1

Name: Chris Brown

Role: Customer Service Rep

Clients Served: 2

Average Service Time: 17.5 minutes

Average Customer Satisfaction Level: 4.25

Service Area: Product Support

Employee ID: 2

Name: Emma Davis

Role: Technical Support Specialist

Clients Served: 1

Average Service Time: 30.0 minutes

Average Customer Satisfaction Level: 5.0

Service Area: Technical Assistance

**Conclusion:**

This integrated program provides a complete solution to manage both general HR and customer service metrics in one system, producing concrete results about employee performance and customer service quality. This allows businesses to maintain efficient employee tracking and customer service optimization.