UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

PROGRAMMING TECHNIQUE II - SECJ1023

Section 1

Group 09

Problem Analysis and Design (Deliverable 2)

NAME	MATRIC NO.
MATHABA HASSAN MOHAMED HASSAN	A23CS4044
LEENA ATAELMANAN ELSIDDIG AWADALLA	A23CS4043
TAGWA BASHIR ABDULLA KUBUR	A23CS4057
AHMED KHALID AHMED MOHAMMEDAHMED	A23SC4037

Table of Content

1.0 Problem Analysis

Video discussion link :
https://drive.google.com/drive/folders/1b8_tTOQ1ey3uTDYXDmXsvTRNKtvjUcwf?usp=drive_link

1.1 Classes and objects

- **User** : Super class whose attributes are inherited in student and staff class.
- **Student**: Class inherits from User and includes additional attributes specific to students.
- **Staff**:Class also inherits from User and includes attributes specific to staff members.
- **Vendo**r:class represents vendors available in Merint , all vendors will be specified with their names, location, and cuisine type.
- **Menu:** Class represents the menus, specifying its details of items names and prices.
- **Order:** Class represents user orders from a specific vendor menu**.**
- **Payment** : Manages payment details for the order with payment methods option.
- **OrderTracking** : Class monitors the status of the order , location and estimated delivery time or order.
- **Delivery** : Class handles the delivery details which like time ,fees and status of orders in the real-time .

1.2 Attributes and methods

User:

Attributes:

- userID
- userName
- userEmail
- userNumber

Methods:

- User(string id, string name, string email, string phoneNum)
- getUserID()
- getUserName()
- getUserEmail()
- getUserNumber()

Student :
Additional attributes:

- MeritPoints

Methods:

- getMeritPoints() const
- Void setMeritPoints()

Staff class:
Additional attributes:

- staffpoistion

Methods:

- staff(){}
- String getstaffpoistion()
- Void setstaffpoistion()

Menu class:
Attributes:

- vector<string> menuItems;
- vector<double> prices;
- int count;

Methods:

- Menu(const vector<string>& items, const vector<double>& prices, int count): menuItems(items), prices(prices), count(count) {}
- Void displayMenu()

Vendor class:
Attributes:

- vendorID
- vendorName
- vendorLocation
- cusineType
- Menu menu

Methods:
Constructors:
Vendor(string id, string name, string location, string cuisineType){}

Methods:

- Void displayVendorList()
- getVendorID() const { return vendorID; }
- string getVendorName() const { return vendorName; }
- string getVendorLocation() const { return vendorLocation; }
- string getCuisineType() const { return cuisineType; }
- const Menu& getMenu() const { return menu; }
- void displayVendorMenu(const Vendor& vendor) { const Menu& menu = vendor.getMenu(); menu.displayMenu(); }//display vendor,s menu

Order Class

Attributes:

- orderID
- vector<string> orderItems
- vector<double> itemPrice
- vector<int> itemQuantity
- Payment orderPayment
- OrderTracking orderTracking
- Delivery* orderDelivery
- string orderLocation
- string orderType

Methods:

- Order(): Default constructor that initializes all attributes to default values.
- Order(const vector<string>& items, const vector<double>& prices, const vector<int>& quantities, Payment payment, OrderTracking tracking, const string& location, const string& type, Delivery* delivery = nullptr): Parameterized constructor that sets an order with items, prices, payment, tracking, location, type, and optionally delivery details, points to null if it is a pickup order.
- ~Order() : destructor
- void addItem(const string& itemName, int quantity, const Menu& menu)
- double calculateSubTotal()
- void displayOrderDetails() const

Payment Class

Attributes:

- paymentID
- totalAmount
- paymentMethod
- paymentStatus

Methods:

- Payment()
- Payment(int id, double amount, const string& method)
- void setPaymentStatus(const string& status)
- bool getPaymentStatus() const
- friend double calculateDiscount(Payment& payment, double discountPercentage)
- double calculateFinalAmount(double discountPercentage)

OrderTracking Class

Attributes:

- orderID (from order class)
- userLocation
- estimatedDeliveryTime

Constructors:

- OrderTracking() {
  estimatedDeliveryTime = 20;
  } a default constructor to set the default time of order to 20 minutes

Delivery Class

Attributes:

- deliveryID
- deliveryAddress
- deliveryStatus (delivered, pending)
- deliveryTime
- deliveryFee

Constructors:

- Delivery() {

    deliveryTime = 20;
    deliveryFee = 5;
    deliveryStatus = false; } a default constructor to set the default delivery

Methods:

- Int getEstimatedDeliveryTime(string userLocation)

  { // returns the time based on the location by predefined timings for specific predefined locations

  }

1.3 Class relationships

1.3.1 Inheritance

User is a superclass that inherits its attribute to **student** and **staff** class ,the last two classes share common attributes like userID ,userName ,userNumber,userEmail ,and behaviour like viewing the available vendors , menus and selecting menu items and order. That is why we included it in another class and implemented the inheritance relationship. **Student** class has an additional attribute which is the merit point that the student can use to get a discount . There exists a strong **is-a** relationship that indicates inheritance, user is a student or user is a staff.

1.3.2.Aggregation

The aggregation relationship between the classes **Order** and **Delivery.** This is a weak 'has a' optional relationship, the existence of the enclosing and enclosed objects are independent, because the order doesn't necessarily have a delivery since there is an option for pickup. So, if one is destroyed, the other remains available to use.

1.3.3 Composition

A)There is a strong ownership relation between **Order** class and **Payment** class, these two classes are highly dependent on each other- Order 'has a' payment.It represents a strong ownership since the payment is mandatory for each order.

B)There is a strong ownership relation between the **Order** class and **OrderTracking** class, where each order whether it has to be picked up or delivered has orderTracking features that include estimated delivery time.

1.3.4 Association

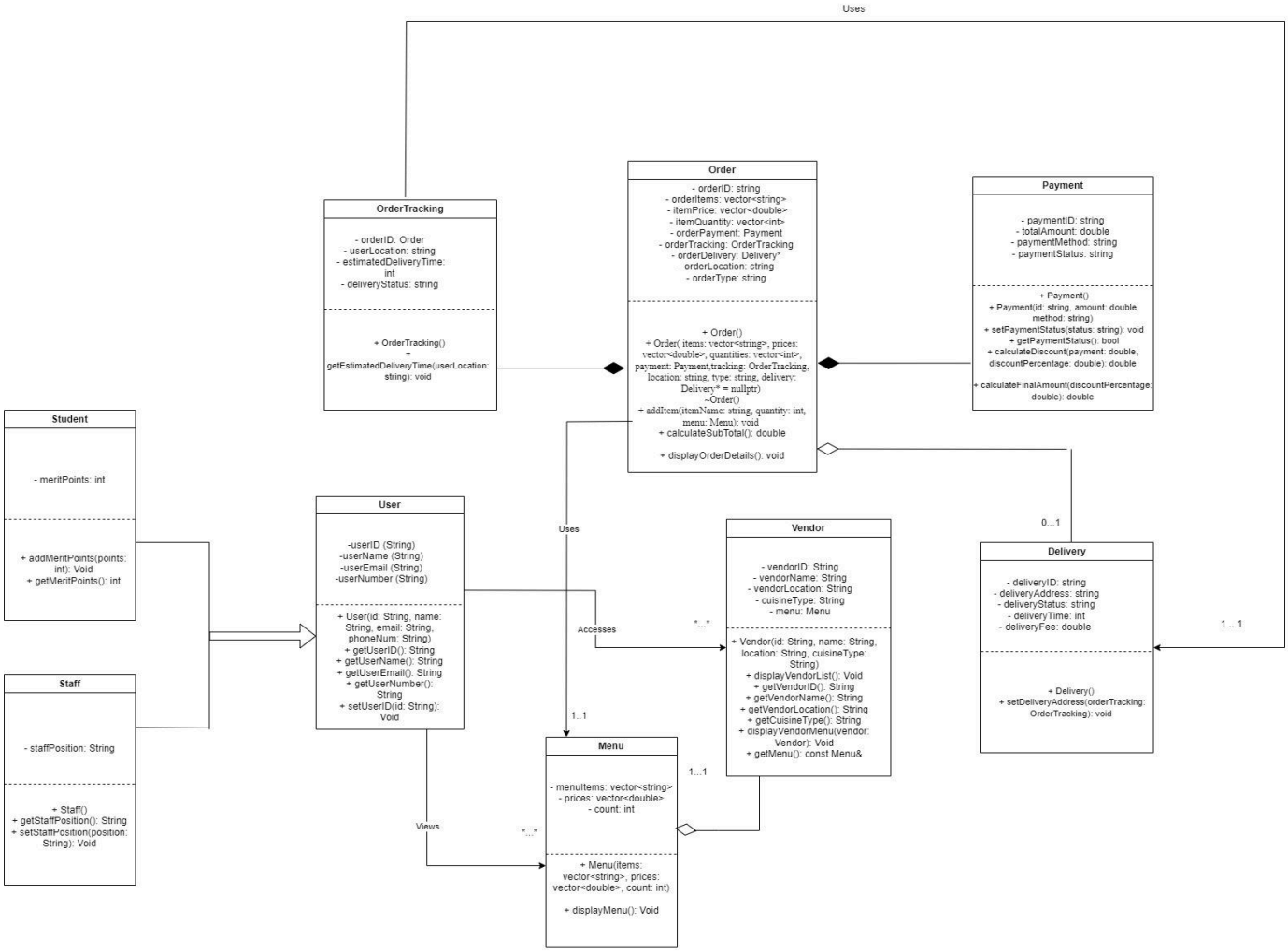
A) View relationships ,**User** class view list of the available **Vendor** with details, .User and vendor are independent from each other .The existence of the User does not depend on any particular vendor and vice versa.that why its association relationship.

B) View and select relationship between **User** class and **Menu** class ,menu class is not related with user class; there is no dependence relationship between both classes ,that is why they are just associated with each other .

C) Association relationship between **Order** class and **Menu** class because the order class "uses" the menu class in order to get the item prices from the Menu class that are chosen by the user when ordering, so the existence of one doesn't depend on the other.

D) Association relationship between OrderTracking and **Delivery, OrderTracking** depends on information from **Delivery** to estimate the delivery time and potentially update the order's location but it does not own **Delivery**, they exist separately and **OrderTracking** interacts with **Delivery** only for specific tasks.

2.0 Design



Reference for UML: https://drive.google.com/drive/folders/1b8_tTOQ1ey3uTDYXDmXsvTRNKtvjUcwf