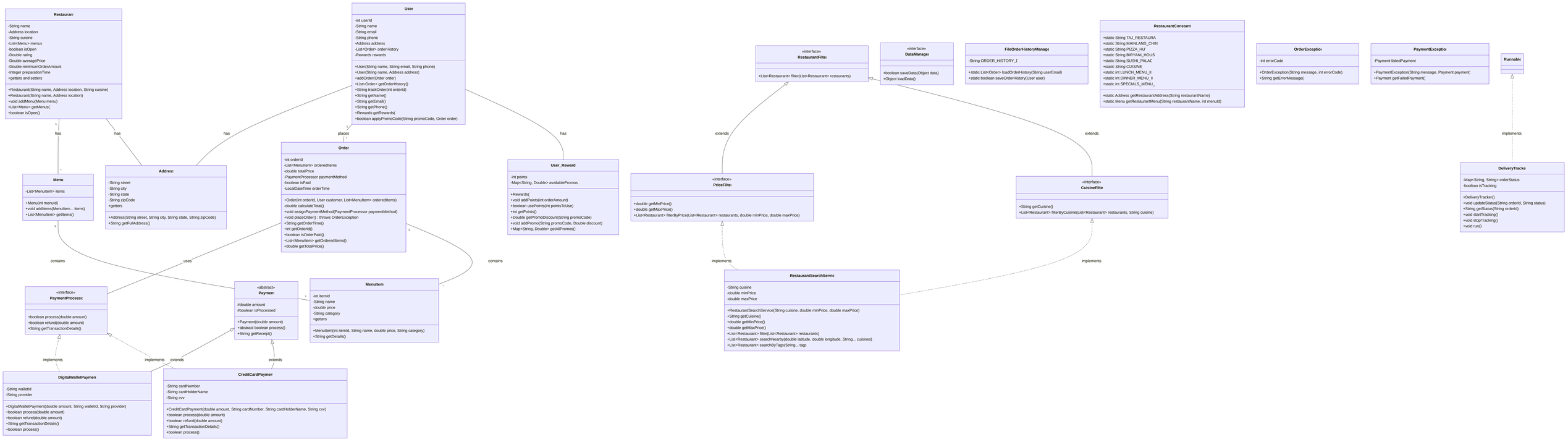
Restaurant
-String name
-Address location
-String cuisine
-List<Menu> menus
-boolean isOpen
-Double rating
-Double averagePrice
-Double minimumOrderAmount
-Integer preparationTime
+getters and setters
+Restaurant(String name, Address location, String cuisine)
+Restaurant(String name, Address location)
+void addMenu(Menu menu)
+List<Menu> getMenus()
+boolean isOpen()
User
-int userId
-String name
-String email
-String phone
-Address address
-List<Order> orderHistory
-Rewards rewards
+User(String name, String email, String phone)
+User(String name, Address address)
+addOrder(Order order)
+List<Order> getOrderHistory()
+String trackOrder(int orderId)
+String getName()
+String getEmail()
+String getPhone()
+Rewards getRewards()
+boolean applyPromoCode(String promoCode, Order order)
has
has
has
places
has
Menu
-List<MenuItem> items
+Menu(int menuId)
+void addItems(MenuItem... items)
+List<MenuItem> getItems()
Address
-String street
-String city
-String state
-String zipCode
+getters
+Address(String street, String city, String state, String zipCode)
+String getFullAddress()
Order
-int orderId
-List<MenuItem> orderedItems
-double totalPrice
-PaymentProcessor paymentMethod
-boolean isPaid
-LocalDateTime orderTime
+Order(int orderId, User customer, List<MenuItem> orderedItems)
-double calculateTotal()
+void assignPaymentMethod(PaymentProcessor paymentMethod)
+void placeOrder() : throws OrderException
+String getOrderTime()
+int getOrderId()
+boolean isOrderPaid()
+List<MenuItem> getOrderedItems()
+double getTotalPrice()
User_Reward
-int points
-Map<String, Double> availablePromos
+Rewards()
+void addPoints(int orderAmount)
+boolean usePoints(int pointsToUse)
+int getPoints()
+Double getPromoDiscount(String promoCode)
+void addPromo(String promoCode, Double discount)
+Map<String, Double> getAllPromos()
contains
uses
contains
«interface»
PaymentProcessor
+boolean process(double amount)
+boolean refund(double amount)
+String getTransactionDetails()
«abstract»
Payment
#double amount
#boolean isProcessed
+Payment(double amount)
+abstract boolean process()
+String getReceipt()
MenuItem
-int itemId
-String name
-double price
-String category
+getters
+MenuItem(int itemId, String name, double price, String category)
+String getDetails()
implements
implements
extends
extends
DigitalWalletPayment
-String walletId
-String provider
+DigitalWalletPayment(double amount, String walletId, String provider)
+boolean process(double amount)
+boolean refund(double amount)
+String getTransactionDetails()
+boolean process()
CreditCardPayment
-String cardNumber
-String cardHolderName
-String cvv
+CreditCardPayment(double amount, String cardNumber, String cardHolderName, String cvv)
+boolean process(double amount)
+boolean refund(double amount)
+String getTransactionDetails()
+boolean process()
«interface»
RestaurantFilter
+List<Restaurant> filter(List<Restaurant> restaurants)
«interface»
DataManager
+boolean saveData(Object data)
+Object loadData()
FileOrderHistoryManager
-String ORDER_HISTORY_F
+static List<Order> loadOrderHistory(String userEmail)
+static boolean saveOrderHistory(User user)
RestaurantConstant
+static String TAJ_RESTAURA
+static String MAINLAND_CHIN
+static String PIZZA_HU
+static String BIRYANI_HOUS
+static String SUSHI_PALAC
+static String CUISINE
+static int LUNCH_MENU_II
+static int DINNER_MENU_II
+static int SPECIALS_MENU_
+static Address getRestaurantAddress(String restaurantName)
+static Menu getRestaurantMenu(String restaurantName, int menuId)
OrderException
-int errorCode
+OrderException(String message, int errorCode)
+String getErrorMessage()
PaymentException
-Payment failedPayment
+PaymentException(String message, Payment payment)
+Payment getFailedPayment()
Runnable
extends
extends
implements
implements
DeliveryTracke
-Map<String, String> orderStatus
-boolean isTracking
+DeliveryTracker()
+void updateStatus(String orderId, String status)
+String getStatus(String orderId)
+void startTracking()
+void stopTracking()
+void run()
«interface»
PriceFilte
+double getMinPrice()
+double getMaxPrice()
+List<Restaurant> filterByPrice(List<Restaurant> restaurants, double minPrice, double maxPrice)
«interface»
CuisineFilte
+String getCuisine()
+List<Restaurant> filterByCuisine(List<Restaurant> restaurants, String cuisine)
RestaurantSearchServic
-String cuisine
-double minPrice
-double maxPrice
+RestaurantSearchService(String cuisine, double minPrice, double maxPrice)
+String getCuisine()
+double getMinPrice()
+double getMaxPrice()
+List<Restaurant> filter(List<Restaurant> restaurants)
+List<Restaurant> searchNearby(double latitude, double longitude, String... cuisines)
+List<Restaurant> searchByTags(String... tags)