PasoEats Design Explanation

Encapsulation:

- All fields are private. Variables are only accessible through getter and setter methods.

Inheritance:

- Customer, Driver, and Admin extend User to reuse its class attributes (`UUID id, String username, String name`).

Polymorphism:

- For our future CLI we will use different fields from the User class for each user type.

Abstraction:

- User class is abstract and has abstract methods for each user type.
- Orders exposes place/acceptNext/markStatus/get while hiding its Map and ArrayDeque

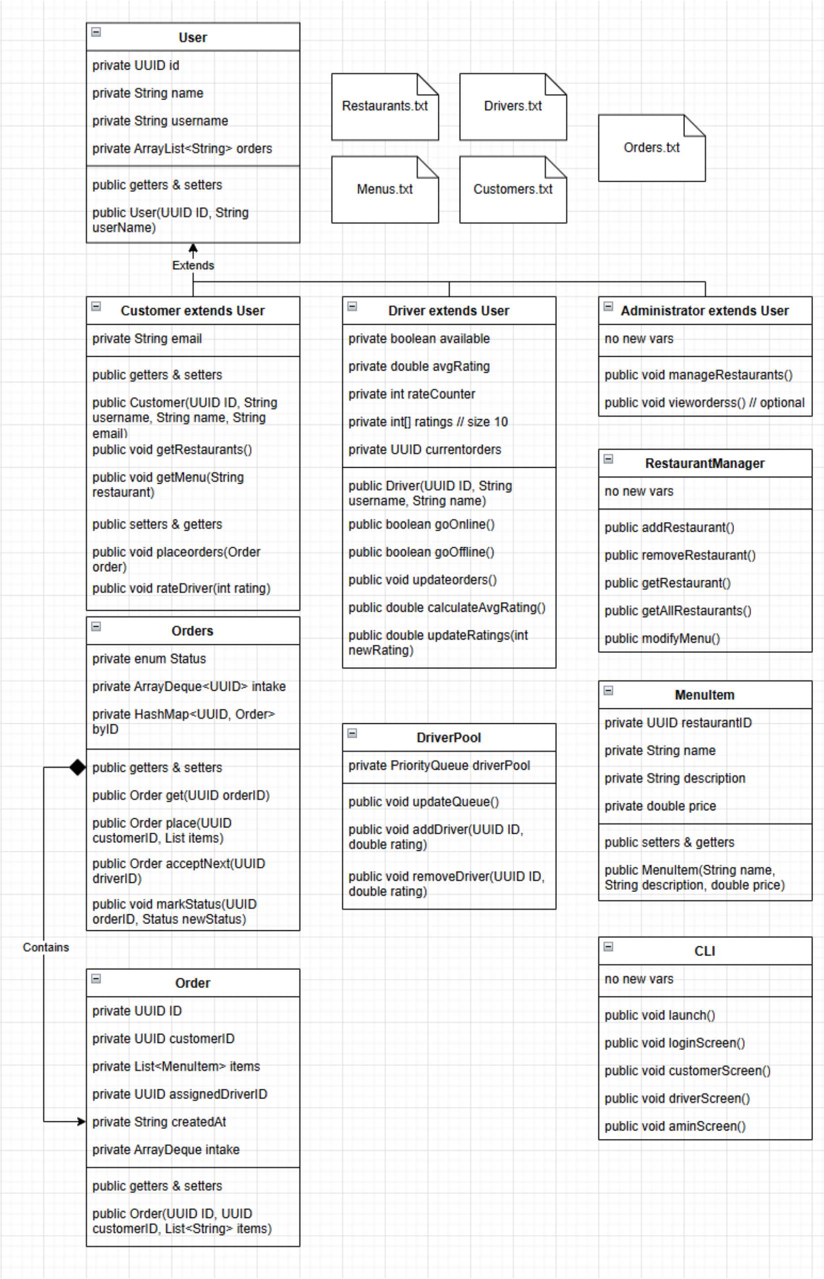
Data Structures:

- `byId: Map<UUID, Order>` - to store orders by their ID and have O(1) lookup time.
- `intake: ArrayDeque<UUID>` - to store orders in the order they were placed and have O(1) insertion and removal time. FIFO (First In, First Out).
- `List<OrderItem>` (inside Order) - to store the items in the order.
- `UUID` (User, Order, Driver, Customer, Admin) - to identify each object uniquely and control access to them in a secure way.

Overall Complexity:

Time Complexity: O(1) for all operations. (*Subject to change if the design evolves.*)
Space Complexity: O(n) for the map and queue.

Class Diagram



User

private UUID id
private String name
private String username
private ArrayList<String> orders

public getters & setters
public User(UUID ID, String userName)

Restaurants.txt

Drivers.txt

Orders.txt

Menus.txt

Customers.txt

Customer extends User

private String email

public getters & setters
public Customer(UUID ID, String username, String name, String email)
public void getRestaurants()
public void getMenu(String restaurant)
public setters & getters
public void placeorders(Order order)
public void rateDriver(int rating)

Driver extends User

private boolean available
private double avgRating
private int rateCounter
private int[] ratings // size 10
private UUID currentorders

public Driver(UUID ID, String username, String name)
public boolean goOnline()
public boolean goOffline()
public void updateorders()
public double calculateAvgRating()
public double updateRating(int newRating)

Administrator extends User

no new vars

public void manageRestaurants()
public void vieworderss() // optional

RestaurantManager

no new vars

public addRestaurant()
public removeRestaurant()
public getRestaurant()
public getAllRestaurants()
public modifyMenu()

Orders

private enum Status
private ArrayDeque<UUID> intake
private HashMap<UUID, Order> byID

public getters & setters
public Order get(UUID orderID)
public Order place(UUID customerID, List items)
public Order acceptNext(UUID driverID)
public void markStatus(UUID orderID, Status newStatus)

DriverPool

private PriorityQueue driverPool

public void updateQueue()
public void addDriver(UUID ID, double rating)
public void removeDriver(UUID ID, double rating)

MenuItem

private UUID restaurantID
private String name
private String description
private double price

public setters & getters
public MenuItem(String name, String description, double price)

CLI

no new vars

public void launch()
public void loginScreen()
public void customerScreen()
public void driverScreen()
public void aminScreen()

Order

private UUID ID
private UUID customerID
private List<MenuItem> items
private UUID assignedDriverID
private String createdAt
private ArrayDeque intake

public getters & setters
public Order(UUID ID, UUID customerID, List<String> items)