Uber System

Description

- **The domain is The system is a ride-sharing platform designed to connect users with drivers for transportation services. It enables users to request rides, select vehicle types, and make payments, while drivers can manage their availability and provide transportation. The system handles user and driver profiles, ride requests, Feedback,Location and payment processing, ensuring smooth and secure interactions between users and drivers.**

1-User Model

```
@NotEmpty(message = "User ID cannot be NULL.")
@Size(min = 3, message = "ID must be more than 2 characters")
private Long userId;

@NotEmpty(message = "Name cannot be empty.")
 @Size(max = 50, message = "Name must be less than 50 characters.")
 private String name;

@NotEmpty(message = "Email cannot be empty.")
 @Email(message = "Email should be valid.")
private String email;

@NotEmpty(message = "Phone number cannot be empty.")
@Pattern(regexp = "^05\\d{8}$", message = "Phone number must start with '05' and contain
exactly 10 digits")
private String phoneNumber;

@NotEmpty(message = "Password cannot be empty.")
private String passwordHash;
```

This model includes essential personal details such as name, email, phone number, and payment methods. It also tracks the user's ride history and profile picture, facilitating user management and interaction.

2-Driver Model

```
@NotEmpty(message = "Driver ID cannot be NULL.")
@Size(min = 3, message = "ID must be more than 2 characters”)
private String driverId;

@NotEmpty(message = "Name cannot be empty.”)
 @Size(max = 50, message = "Name must be less than 50 characters.”)
 private String name;

@NotEmpty(message = "Email cannot be empty.")
@Email(message = "Email should be valid.”)
 private String email;

@NotEmpty(message = "Phone number cannot be empty.”)
 @Pattern(regexp = "^05\\d{8}$", message = "Phone number must start with '05' and contain
exactly 10 digits”)
 private String phoneNumber;

@NotEmpty(message = "License number cannot be empty.”)
 private String licenseNumber;

@NotNull(message = "Vehicle cannot be null.”)
@OneToOne(mappedBy = "driver")
private Vehicle vehicle;


@DecimalMin(value = "0.0", inclusive = true, message = "Rating must be positive.")
@DecimalMax(value = "5.0", inclusive = true, message = "Rating must be less than or equal to 5.")
private Double rating;

@NotEmpty(message = "Background check status cannot be empty.”)
private String backgroundCheckStatus;
```

This model captures the driver's personal information, license details, vehicle information, and current availability status. It also tracks the driver’s rating and background check status to ensure safety and reliability.

3- Payment Model

```
@NotEmpty(message = "Payment ID cannot be NULL.")
@Size(min = 3, message = "ID must be more than 2 characters”)
private Long paymentMethodId;

 @NotNull(message = "User cannot be null.")
private User user;

@NotEmpty(message = "Card number cannot be empty.")
@Pattern(regexp = "^[0-9]{16}$", message = "Card number must be 16 digits.")
private String cardNumber;

 @NotEmpty(message = "Expiry date cannot be empty.”)
@Pattern(regexp = "^(0[1-9]|1[0-2])/[0-9]{2}$", message = "Expiry date must be in MM/YY
format.”)
 private String expiryDate;

@NotEmpty(message = "Card holder name cannot be empty.")
@Size(max = 50, message = "Card holder name must be less than 50 characters.”)
 private String cardHolderName;

@NotEmpty(message = "Billing address cannot be empty.")
private String billingAddress;
```

This model includes information such as the card number, expiration date, cardholder’s name, and billing address. It ensures secure and accurate processing of transactions for ride payments.

4-Location Model

```
@NotNull(message = "Latitude cannot be null.”)
@DecimalMin(value = "-90.0", message = "Latitude must be greater than or equal to -90.0")
@DecimalMax(value = "90.0", message = "Latitude must be less than or equal to 90.0")
privateDouble latitude;

@NotNull(message = "Longitude cannot be null.”)
 @DecimalMin(value = "-180.0", message = "Longitude must be greater than or equal to -180.0")
@DecimalMax(value = "180.0", message = "Longitude must be less than or equal to 180.0")
private Double longitude;


@NotEmpty(message = "Address cannot be empty.")
private String address;
```

This model captures the geographical coordinates and address details for both pickup and dropoff points in the ride-sharing system. It ensures that the latitude and longitude values are valid and within the acceptable range, providing precise location data for rides.

5- Feedback Model

```
@NotEmpty(message = "Feedback ID cannot be NULL.")
@Size(min = 3, message = "ID must be more than 2 characters”)
 private Long feedbackId;

@NotNull(message = "Ride cannot be null.”)
private Ride ride;

@NotNull(message = "User cannot be null.")
private User user;

@NotNull(message = "Driver cannot be null.")
private Driver driver;

@NotEmpty(message = "Comments cannot be empty.")
@Size(max = 255, message = "Comments must be less than 255 characters.")
private String comments;

@DecimalMin(value = "1.0", inclusive = true, message = "Rating must be at least 1.")
@DecimalMax(value = "5.0", inclusive = true, message = "Rating must be at most 5.”)
private Double rating;
```

This model collects and stores user feedback for rides. It includes the ride, user, driver, comments, rating, and a timestamp for when the feedback was given.