LLD Export

<u>Explanation:</u>

Low level design contains classes for User, Bike, Rental, Payment, Notification, and other components. Each class defines its attributes and methods. Relationships between classes are defined using UML notation. The design includes database schema representation with PK and FK annotations. Interfaces define contracts for PaymentGateway and NotificationService. The API package includes notes representing REST endpoints.

UML Diagram:

OrderService
- ● createOrder()
- ● getOrder()

API
- GET /orders/:id
- POST /users
- GET /bikes
- POST /rentals
- POST /payments
- GET /stations

UserManagement

User
- ▢ name : String
- ▢ email : String
- ▢ phone : String
- ▢ registrationDate : Date
- ▢ passwordHash : String
- ■ userId : UUID (PK)
- ● register()
- ● login()
- ● updateProfile()

1..1 owns > 0..*

AuthToken
- ▢ token : String
- ▢ expiryDate : Date
- ■ tokenId : UUID (PK)
- ■ userId : UUID (FK)
- ● generateToken()
- ● validateToken()
- ● revokeToken()

Users create rentals and reservations

RentalManagement

Rental
- ▢ startTime : DateTime
- ▢ endTime : DateTime
- ▢ rentalFee : double
- ■ rentalId : UUID (PK)
- ■ userId : UUID (FK)
- ■ bikeId : UUID (FK)
- ■ startStationId : UUID (FK)
- ■ endStationId : UUID (FK)
- ● startRental()
- ● endRental()
- ● calculateFee()

uses

Reservation
- ▢ startTime : DateTime
- ▢ endTime : DateTime
- ▢ reservationStatus : enum {ACTIVE, CANCELLED, COMPLETED}
- ■ reservationId : UUID (PK)
- ■ userId : UUID (FK)
- ■ bikeId : UUID (FK)
- ■ stationId : UUID (FK)
- ● createReservation()
- ● cancelReservation()
- ● confirmReservation()

Users use bikes at stations

Rentals use bikes

BikeManagement

BikeStation
- ▢ location : String
- ▢ capacity : int
- ▢ availableBikes : int
- ■ stationId : UUID (PK)
- ● addBike()
- ● removeBike()
- ● updateAvailability()

1 owns > 0..*

Bike
- ▢ bikeStatus : enum {AVAILABLE, RENTED, MAINTENANCE}
- ▢ model : String
- ▢ lastMaintenanceDate : Date
- ■ bikeId : UUID (PK)
- ■ stationId : UUID (FK)
- ● rentBike()
- ● returnBike()
- ● reportDamage()

Bikes trigger notifications

Notification

NotificationService
- ● sendNotification(userId : UUID, message : String) : void

SMSNotification
- ● sendNotification(userId : UUID, message : String) : void

EmailNotification
- ● sendNotification(userId : UUID, message : String) : void

Users make payments

Rentals generate payments

Users receive notifications

Payment

PaymentTransaction
- ▢ paymentDate : DateTime
- ▢ amount : double
- ▢ paymentMethod : enum {CREDIT_CARD, WALLET}
- ▢ transactionStatus : enum {SUCCESS, FAILED}
- ■ transactionId : UUID (PK)
- ■ userId : UUID (FK)
- ■ rentalId : UUID (FK)
- ● processPayment()
- ● refundPayment()
- ● getTransactionDetails()

uses

PaymentGateway
- ● processPayment(amount : double, paymentMethod : String) : boolean
- ● refundPayment(transactionId : UUID, amount : double) : boolean

CreditCardPayment
- ● processPayment(amount : double, paymentMethod : String) : boolean
- ● refundPayment(transactionId : UUID, amount : double) : boolean