

Sequence Diagram Documentation: Ride Flow

1. Request Ride

1. The `User` initiates the ride request by calling the `requestRide` method on the `RideController`.
2. The `RideController` forwards this request to the `RideService` via the `handleRideRequest` method.
3. The `RideService` interacts with the `DriverService` to fetch a list of available drivers.
4. The `DriverService` returns the list of available drivers to the `RideService`.
5. The `RideService` creates a new ride and saves it to the database via the `RideRepository`.
6. The `RideService` notifies the `RideController` that the ride has been assigned to a driver, which completes this step.

2. Accept Ride

1. The `Driver` accepts the assigned ride by calling the `acceptRide` method on the `RideController`.
2. The `RideController` delegates this action to the `RideService` via the `handleRideAcceptance` method.
3. The `RideService` updates the ride's status to "ACCEPTED".
4. The `RideService` confirms the ride acceptance to the `RideController`, completing this step.

3. Start Ride

1. The `Driver` starts the ride by calling the `startRide` method on the `RideController`.
2. The `RideController` invokes the `handleRideStart` method in the `RideService`.
3. The `RideService` validates the ride OTP by calling the `validateOTP` method on the `OTPService`.
4. The `OTPService` confirms that the OTP is valid and returns the result to the `RideService`.
5. The `RideService` updates the ride status to "STARTED".
6. The `RideService` notifies the `RideController` that the ride has started, completing this step.

4. Complete Ride

1. The `Driver` completes the ride by calling the `completeRide` method on the `RideController`.
2. The `RideController` forwards the request to the `RideService` via the `handleRideCompletion` method.
3. The `RideService` updates the ride status to "COMPLETED".
4. The `RideService` informs the `RideController` that the ride is completed.

5. Calculate Fare

1. The `Driver` requests the `RideService` to calculate the fare for the completed ride by calling the `calculateFare` method with the ride details.
2. The `RideService` calculates the fare and returns the amount to the `Driver`.
3. The `Driver` provides the calculated fare to the `User`.

6. Verify Payment

1. The `User` verifies the payment by calling the `verifyPayment` method on the `RideController` and providing the fare amount.
2. The `RideController` delegates this request to the `RideService` via the `handlePaymentVerification` method.
3. The `RideService` updates the payment status to "VERIFIED".
4. The `RideService` informs the `RideController` that the payment has been successfully verified, completing the process.