۱. روت رانندگان (Drivers):
   - GET /drivers/: دریافت لیست تمام رانندگان.
   - GET /drivers/{id}: دریافت جزئیات یک راننده با استفاده از شناسه.
   - POST /drivers/: ایجاد یک راننده جدید.
   - PUT /drivers/{id}/: به‌روزرسانی جزئیات یک راننده با استفاده از شناسه.
   - DELETE /drivers/{id}/: حذف یک راننده با استفاده از شناسه.
۲. روت مشتریان (Customers):
   - GET /customers/: دریافت لیست تمام مشتریان.
   - GET /customers/{id}/: دریافت جزئیات یک مشتری با استفاده از شناسه.
   - POST /customers/: ایجاد یک مشتری جدید.
   - PUT /customers/{id}/: به‌روزرسانی جزئیات یک مشتری با استفاده از شناسه.
   - DELETE /customers/{id}/: حذف یک مشتری با استفاده از شناسه.
۳. روت سفرها (Trips):
   - GET /trips/: دریافت لیست تمام سفرها.
   - GET /trips/{id}/: دریافت جزئیات یک سفر با استفاده از شناسه.
   - POST /trips/: ایجاد یک سفر جدید.
   - PUT /trips/{id}/: به‌روزرسانی جزئیات یک سفر با استفاده از شناسه.
   - DELETE /trips/{id}/: حذف یک سفر با استفاده از شناسه.

۴. روت سفرهای راننده (Driver Trips):
   - GET /drivers/{driver_id}/trips/: دریافت لیست سفرهای یک راننده با استفاده از شناسه راننده.
   - GET /drivers/{driver_id}/trips/{trip_id}/: دریافت جزئیات یک سفر راننده با استفاده از شناسه راننده و شناسه سفر.
   - POST /drivers/{driver_id}/trips/: ایجاد یک سفر جدید برای یک راننده با استفاده از شناسه راننده.
   - POST /drivers/{driver_id}/trips/{trip_id}/: پایان سفر با شناسه ی سفر
۵. روت سفرهای مشتری (Customer Trips):
   - GET /customers/{customer_id}/trips/: دریافت لیست سفرهای یک مشتری با استفاده از شناسه مشتری.
   - GET /customers/{customer_id}/trips/{trip_id}: دریافت جزئیات یک سفر مشتری با استفاده از شناسه مشتری و شناسه سفر.
   - POST /customers/{customer_id}/trips: ایجاد یک سفر جدید برای یک مشتری با استفاده از شناسه مشتری.
۶. روت موقعیت جغرافیایی راننده‌ها (Driver Locations):
   - GET /drivers/locations: دریافت لیست موقعیت جغرافیایی تمام رانندگان.
   - GET /drivers/{driver_id}/location: دریافت موقعیت جغرافیایی یک راننده با استفاده از شناسه راننده.
   - POST /drivers/{driver_id}/location: ثبت موقعیت جغرافیایی یک راننده با استفاده از شناسه راننده.
۷. روت اعتبارسنجی (Authentication):
   - POST /auth/login: ورود کاربر به سیستم.
   - POST /auth/logout: خروج کاربر از سیستم.
   - POST /auth/register: ثبت نام کاربر جدید.

۸. روت جستجوی راننده‌ها (Driver Search):
- GET /drivers/search: جستجوی راننده‌ها بر اساس موقعیت جغرافیایی و سایر پارامترهای مورد نظر.

۹. روت نظرات (Reviews):
- GET /reviews: دریافت لیست تمام نظرات.
- GET /reviews/{id}: دریافت جزئیات یک نظر با استفاده از شناسه.
- POST /reviews: ایجاد یک نظر جدید.
- PUT /reviews/{id}: به‌روزرسانی جزئیات یک نظر با استفاده از شناسه.
- DELETE /reviews/{id}: حذف یک نظر با استفاده از شناسه.

۱۰. روت نقشه (Map):
- GET /map/drivers: دریافت لیست رانندگان فعال بر روی نقشه.
- GET /map/drivers/{id}: دریافت موقعیت جغرافیایی یک راننده با استفاده از شناسه.

۱۱. روت پرداخت (Payments):
- POST /payments/{trip-id}: ایجاد یک پرداخت جدید برای یک سفر با شناسه سفر.
- GET /payments/{id}: دریافت جزئیات یک پرداخت با استفاده از شناسه.

```python
from django.db import models
from django.contrib.auth.models import User
from django.utils.translation import gettext_lazy as _



class PStatus(models.IntegerChoices):
    BUSY = 1, _('Busy')
    IDLE = 0, _('Idle')

class TStatus(models.IntegerChoices):
    ONPROCESS = 0, _('On Process')
    FINISHED = 1, _('Finished')
    CANCEL = 2, _('Cancel')
    UNKNOWN = 3, _('Unknown')

class PaymentStatus(models.IntegerChoices):
    ONPROCESS = 0, _('On Process')
    SUCCESS = 1, _('Success')
    CANCEL = 2, _('Cancel')
    ERROR = 3, _('Error')
    UNKNOWN = 4, _('Unknown')
```

```python
class Driver(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    car_model = models.CharField(max_length=100)
    license_plate = models.CharField(max_length=20)
    status = models.IntegerField(
        choices=PStatus.choices,
        default=PStatus.IDLE,
        blank=True,
        null=True
    )
    current_X = models.CharField(max_length=20)
    current_Y = models.CharField(max_length=20)

    def __str__(self):
        return self.user.username

class Customer(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    phone_number = models.CharField(max_length=20)
    address = models.CharField(max_length=255)
    status = models.IntegerField(
        choices=PStatus.choices,
        default=PStatus.IDLE,
        blank=True,
        null=True
    )
    current_X = models.CharField(max_length=20)
    current_Y = models.CharField(max_length=20)

    def __str__(self):
        return self.user.username
```

```python
class Trip(models.Model):
    date = models.DateField(auto_now_add=True)
    time = models.TimeField(auto_now_add=True)
    customer = models.ForeignKey(Customer, on_delete=models.CASCADE)
    driver = models.ForeignKey(Driver, on_delete=models.SET_NULL, null=True,
blank=True)
    start_location = models.CharField(max_length=255)
    end_location = models.CharField(max_length=255)
    pickup_time = models.DateTimeField()
    status = models.IntegerField(
        choices=TStatus.choices,
        default=TStatus.UNKNOWN,
        blank=True,
        null=True
    )

    def __str__(self):
        return f"Trip #{self.id}"

class Payment(models.Model):
    trip = models.OneToOneField(Trip, on_delete=models.CASCADE)
    amount = models.DecimalField(max_digits=8, decimal_places=2)
    status = models.IntegerField(
        choices=PaymentStatus.choices,
        default=PaymentStatus.UNKNOWN,
        blank=True,
        null=True
    )

    def __str__(self):
        return f"Payment for Trip #{self.trip.id}"

class Reviews(models.Model):
    date = models.DateField(auto_now_add=True)
    time = models.TimeField(auto_now_add=True)
    trip = models.OneToOneField(Trip, on_delete=models.CASCADE)
    customer = models.ForeignKey(Customer, on_delete=models.CASCADE)
    driver = models.ForeignKey(Driver, on_delete=models.CASCADE)
    rating = models.PositiveIntegerField()
    comment = models.TextField()

    def __str__(self):
        return f"Rating for Trip #{self.trip.id}"
```