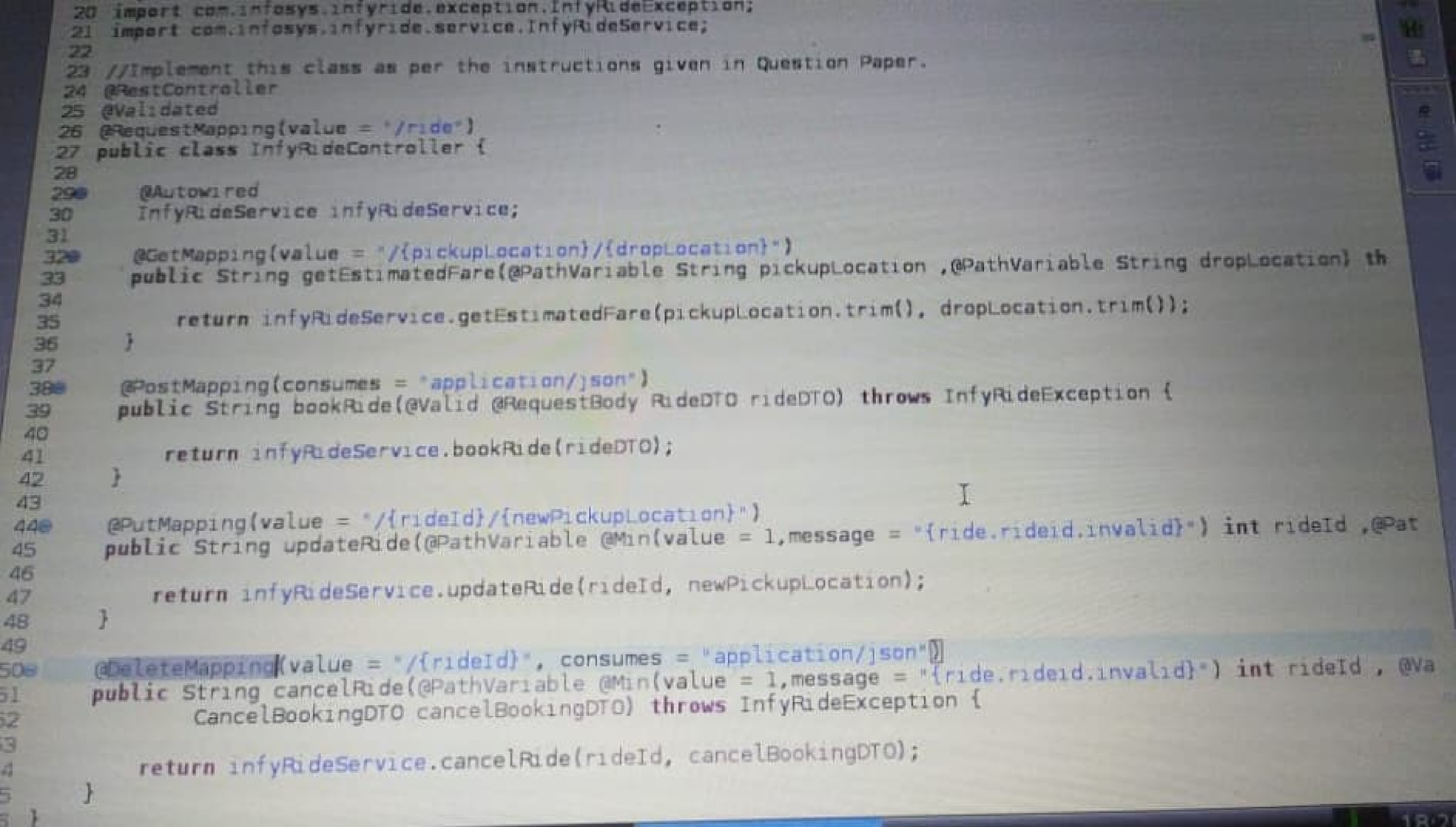
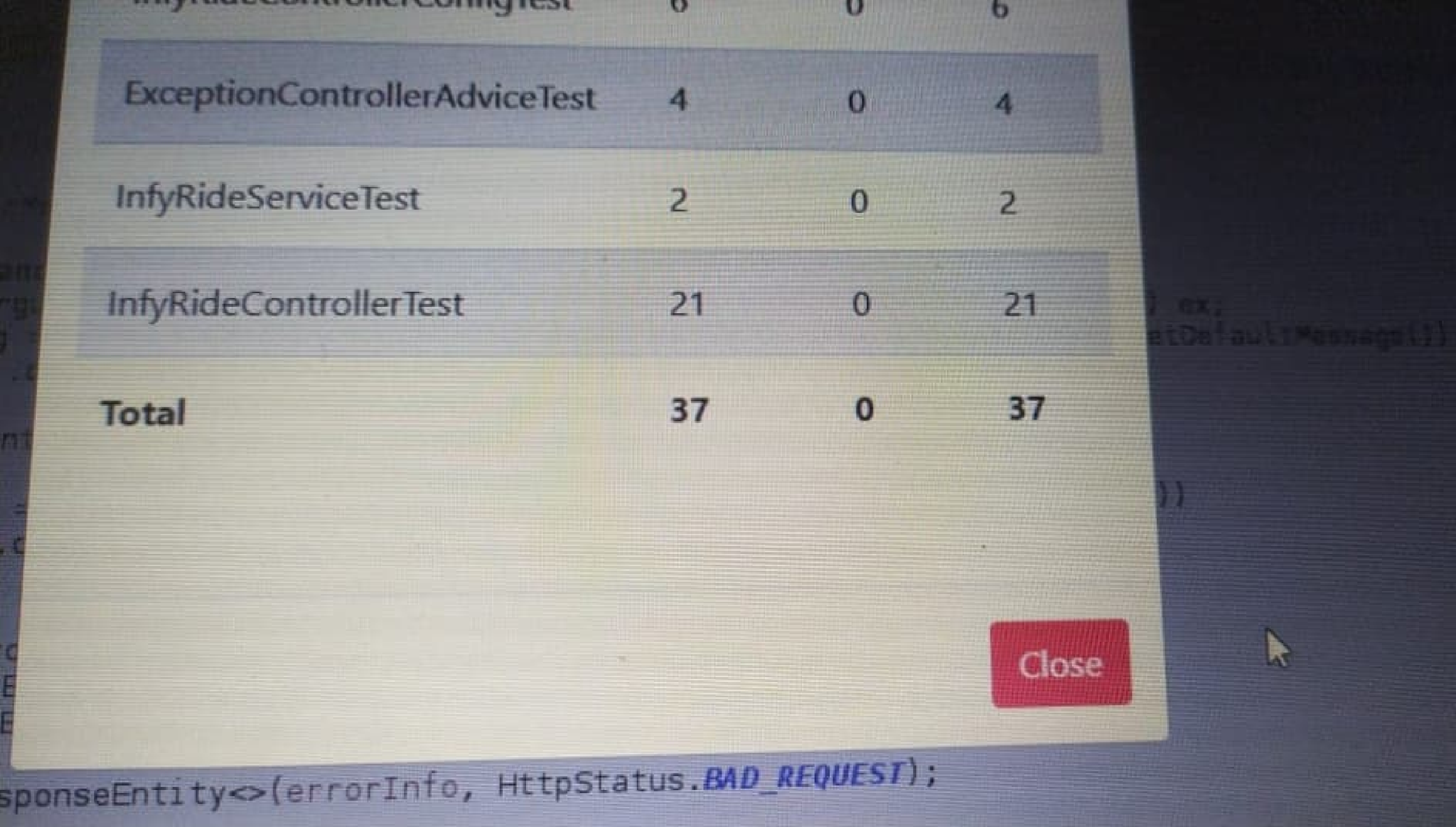


```java
import com.infosys.infyride.exception.InfyRideException;
import com.infosys.infyride.service.InfyRideService;

//Implement this class as per the instructions given in Question Paper.
@RestController
@Validated
@RequestMapping(value = "/ride")
public class InfyRideController {

    @Autowired
    InfyRideService infyRideService;

    @GetMapping(value = "/{pickupLocation}/{dropLocation}")
    public String getEstimatedFare(@PathVariable String pickupLocation ,@PathVariable String dropLocation) th

        return infyRideService.getEstimatedFare(pickupLocation.trim(), dropLocation.trim());
    }

    @PostMapping(consumes = "application/json")
    public String bookRide(@Valid @RequestBody RideDTO rideDTO) throws InfyRideException {

        return infyRideService.bookRide(rideDTO);
    }

    @PutMapping(value = "/{rideId}/{newPickupLocation}")
    public String updateRide(@PathVariable @Min(value = 1,message = "{ride.rideid.invalid}") int rideId ,@Pat

        return infyRideService.updateRide(rideId, newPickupLocation);
    }

    @DeleteMapping(value = "/{rideId}", consumes = "application/json")
    public String cancelRide(@PathVariable @Min(value = 1,message = "{ride.rideid.invalid}") int rideId , @Va
            CancelBookingDTO cancelBookingDTO) throws InfyRideException {

        return infyRideService.cancelRide(rideId, cancelBookingDTO);
    }
}
```

```java
        if (fareEntity == null)
            throw new InfyRideException(InfyRideConstants.INFYRIDE_PICKUPTODROPLOCATION_NOT_FOUND.toString()

        return environment.getProperty(InfyRideConstants.INFYRIDE_GETESTIMATED_FARE_SUCCESS.toString())
                + fareEntity.getFare();
    }

    //Implement this method as per the instructions given in Question Paper.
    @Override
    public String bookRide(RideDTO rideDTO) throws InfyRideException {
        //String estimatedFare = getEstimatedFare(rideDTO.getPickupLocation(),rideDTO.getDropLocation());
        FareEntity fareEntity = fareRepository.
                getByPickupLocationIgnoreCaseAndDropLocationIgnoreCase
                (rideDTO.getPickupLocation(),rideDTO.getDropLocation());
        if (fareEntity == null)
            throw new InfyRideException(InfyRideConstants.INFYRIDE_PICKUPTODROPLOCATION_NOT_FOUND.toString()
        RideEntity rideEntity = RideDTO.prepareRideEntity(rideDTO);
        rideEntity.setStatus("BOOKED");
        rideEntity.setTotalFare(fareEntity.getFare());
        rideRepository.save(rideEntity);
        return environment.getProperty(InfyRideConstants.INFYRIDE_BOOKING_SUCCESS.toString())
                + fareEntity.getFare();
    }

    @Override
    @Transactional
    public String updateRide(int rideId, String newPickupLocation) throws InfyRideException {

        Optional<RideEntity> rideEntityOpt = rideRepository.findById(rideId);

        if (!rideEntityOpt.isPresent())
            throw new InfyRideException(InfyRideConstants.INFYRIDE_RIDEID_NOT_FOUND.toString());
```



InfyRideControllerConfigTest	6	0	6
ExceptionControllerAdviceTest	4	0	4
InfyRideServiceTest	2	0	2
InfyRideControllerTest	21	0	21
Total	**37**	**0**	**37**

```
sponseEntity<>(errorInfo, HttpStatus.BAD_REQUEST);
```