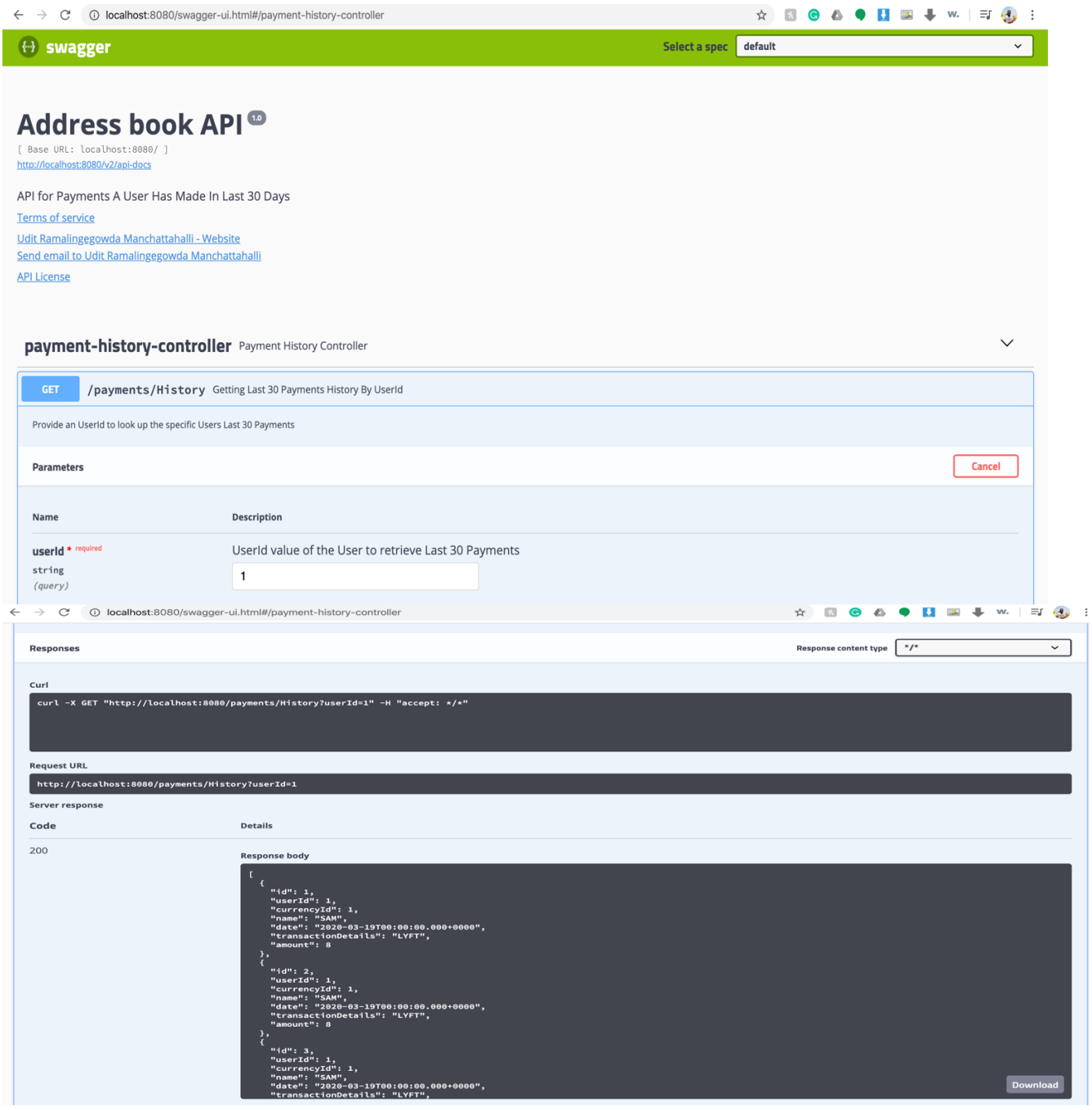
Creating a REST API Design Using Swagger and Implementation in Spring Boot Application

Task:

"*Design a REST API that returns a list of payments a user has made in the last 30 days. Looking for RESTful API design with good interface design.* No need to go into algorithms, data structures, or code, just the API specification as if you're documenting the API for the client to use."



404 Request and Response (Not Valid UserId)

Request:

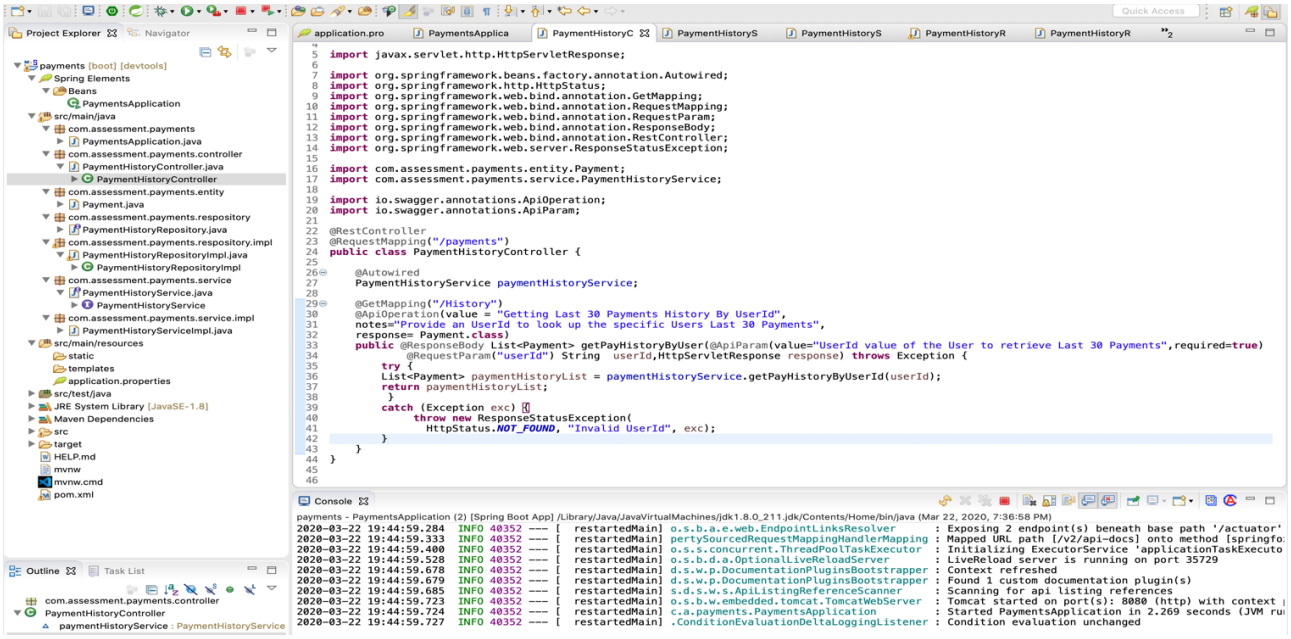
GET /payment/History?userId=2
Host: localhost:8080
Content-Type: application/json

Response:

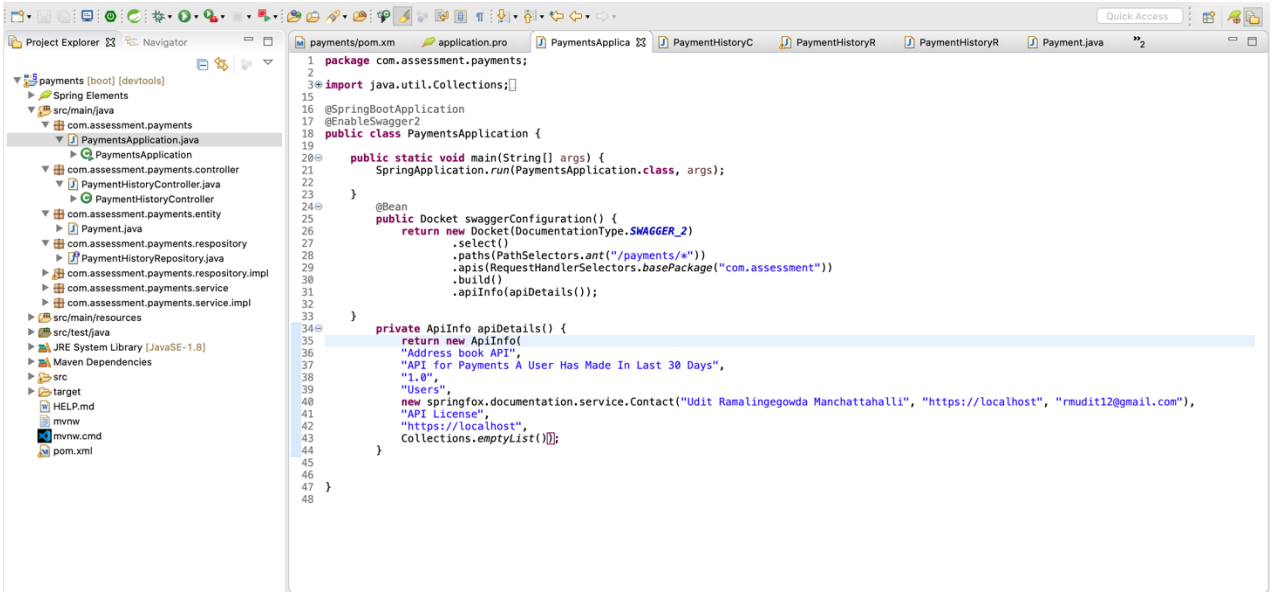
{
"timestamp": "2020-03-21T20:47:31.561+0000",
"status": 404,
"error": "Not Found",
"message": “Invalid UserId",
"path": "/payments/Transcation?userId=2"
}

Spring Boot Application

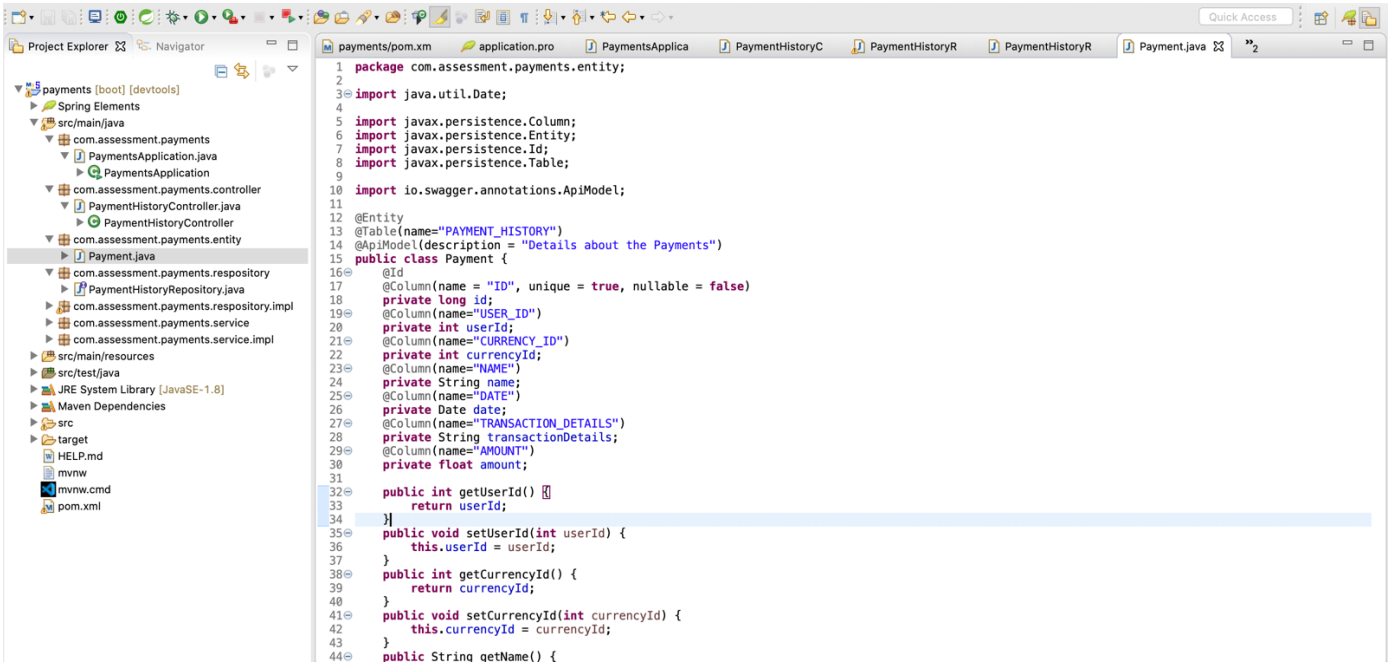
Controller



SpringBoot Starter

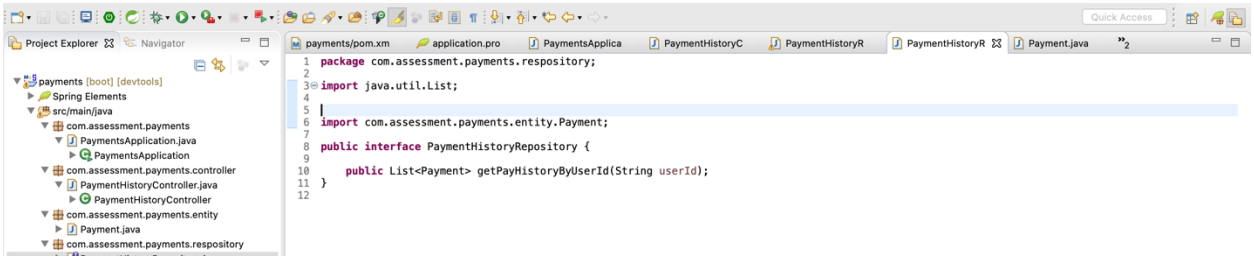


Entity

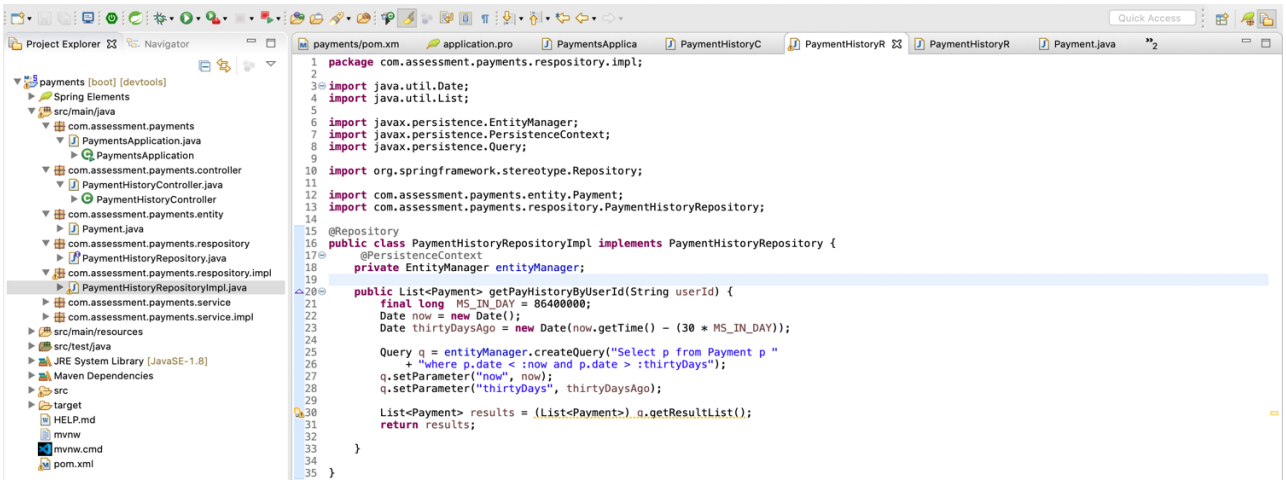


```
package com.assessment.payments.entity;

import java.util.Date;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

import io.swagger.annotations.ApiModel;

@Entity
@Table(name="PAYMENT_HISTORY")
@ApiModel(description = "Details about the Payments")
public class Payment {
    @Id
    @Column(name = "ID", unique = true, nullable = false)
    private long id;
    @Column(name="USER_ID")
    private int userId;
    @Column(name="CURRENCY_ID")
    private int currencyId;
    @Column(name="NAME")
    private String name;
    @Column(name="DATE")
    private Date date;
    @Column(name="TRANSACTION_DETAILS")
    private String transactionDetails;
    @Column(name="AMOUNT")
    private float amount;

    public int getUserId() {
        return userId;
    }
    public void setUserId(int userId) {
        this.userId = userId;
    }
    public int getCurrencyId() {
        return currencyId;
    }
    public void setCurrencyId(int currencyId) {
        this.currencyId = currencyId;
    }
    public String getName() {
```

Repository

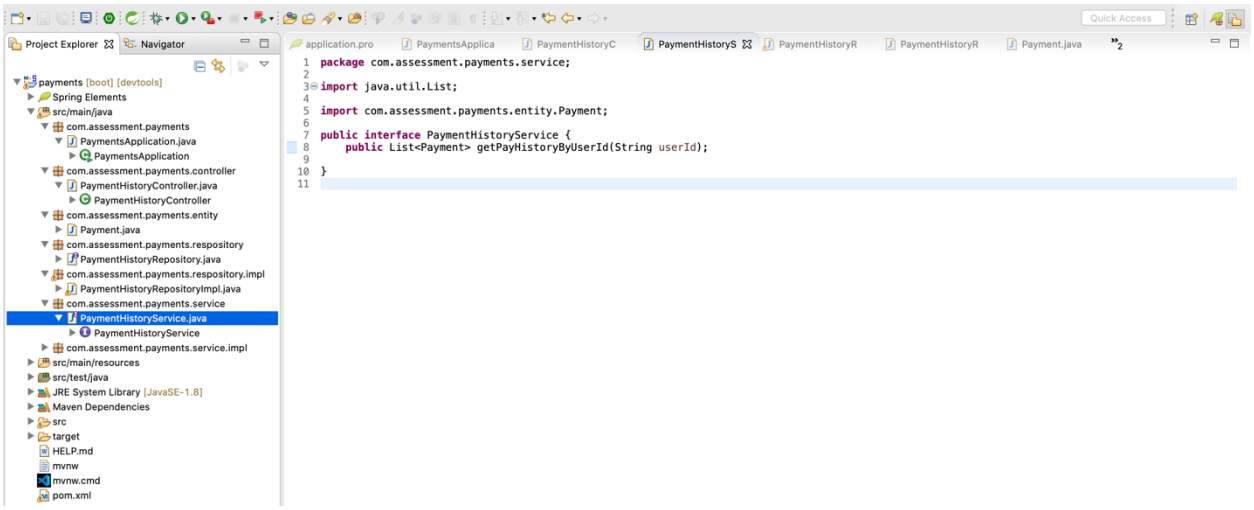


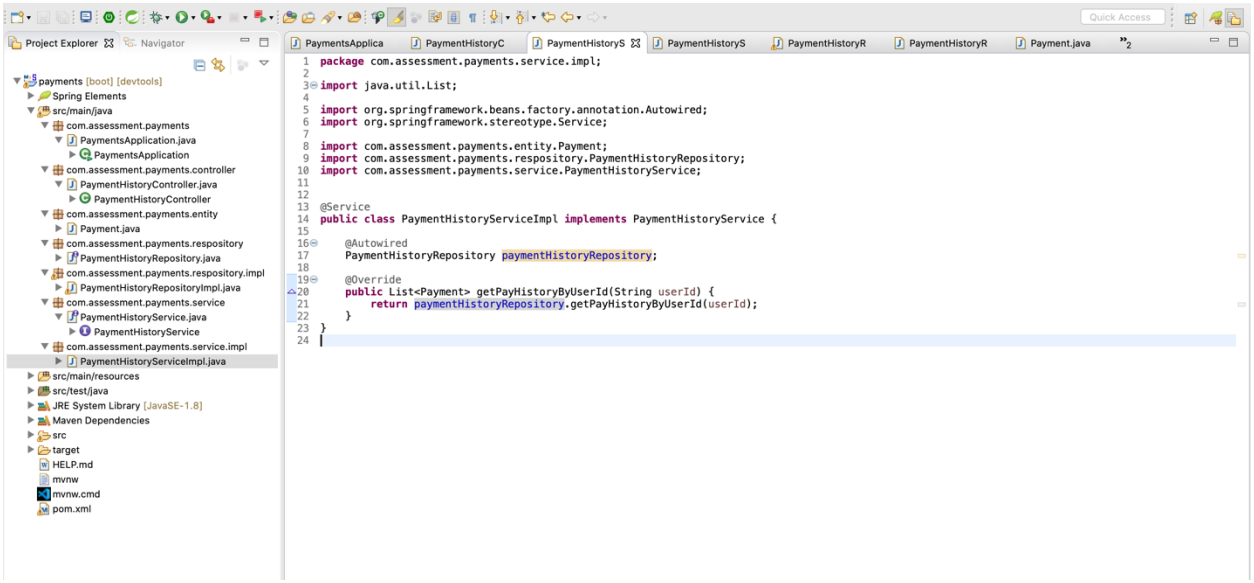
```
package com.assessment.payments.respository;

import java.util.List;

import com.assessment.payments.entity.Payment;

public interface PaymentHistoryRepository {

    public List<Payment> getPayHistoryByUserId(String userId);
}
```



```
package com.assessment.payments.respository.impl;

import java.util.Date;
import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

import org.springframework.stereotype.Repository;

import com.assessment.payments.entity.Payment;
import com.assessment.payments.respository.PaymentHistoryRepository;

@Repository
public class PaymentHistoryRepositoryImpl implements PaymentHistoryRepository {
    @PersistenceContext
    private EntityManager entityManager;

    public List<Payment> getPayHistoryByUserId(String userId) {
        final long  MS_IN_DAY = 86400000;
        Date now = new Date();
        Date thirtyDaysAgo = new Date(now.getTime() - (30 * MS_IN_DAY));

        Query q = entityManager.createQuery("Select p from Payment p "
            + "where p.date < :now and p.date > :thirtyDays");
        q.setParameter("now", now);
        q.setParameter("thirtyDays", thirtyDaysAgo);

        List<Payment> results = (List<Payment>) q.getResultList();
        return results;

    }
}
```

Service





THANK YOU