

```

public class MyMap extends LinkedHashMap<String, Object> {
    no usages
    private static final long serialVersionUID = 1L;

    /**
     * Return a {@link String String} that represents the JSON object.
     *
     * @author hieudm
     * https://hg.openjdk.java.net/jdk8/jdk8/jdk/file/tip/src/share/classes/java/util/Hashtable.java
     * @return a {@link String String}.
     */
    2 usages 1 thanhld
    public String toJSON() {
        int max = size() - 1;
        if (max == -1)
            return "{}";

        StringBuilder sb = new StringBuilder();
        Iterator<Map.Entry<String, Object>> it = entrySet().iterator();

        sb.append('{');
        for (int i = 0; i++ < max) {
            Map.Entry<String, Object> e = it.next();
            String key = e.getKey();
            Object value = e.getValue();
            sb.append('"' + key.toString() + '"');
            sb.append(':');
            sb.append(value instanceof MyMap ? ((MyMap) value).toJSON() : ('' + value.toString() + ''));
        }
        sb.append('}');

        if (value instanceof MyMap) {
            sb.append(((MyMap) value).toJSON());
        }
    }
}

```

Lớp

MyMap vi phạm SRP vì thực hiện quá nhiều nhiệm vụ.

```

private String getExpirationDate(String date) throws InvalidCardException {
    String[] strs = date.split("/");
    if (strs.length != 2) {
        throw new InvalidCardException();
    }

    String expirationDate = null;
    int month = -1;
    int year = -1;

    try {
        month = Integer.parseInt(strs[0]);
        year = Integer.parseInt(strs[1]);
        if (month < 1 || month > 12 || year < Calendar.getInstance().get(Calendar.YEAR) % 100 || year > 100) {
            throw new InvalidCardException();
        }
        expirationDate = strs[0] + strs[1];
    } catch (Exception ex) {
        throw new InvalidCardException();
    }

    return expirationDate;
}

```

lớp PaymentController.java vi phạm SRP vì có method getExpirationDate không liên quan đến các method còn lại.

```

public void validateDeliveryInfo(HashMap<String, String> info) throws InterruptedException, IOException, InvalidDeliveryInfoException {
    if (validatePhoneNumber(info.get("phone"))
        || validateName(info.get("name"))
        || validateAddress(info.get("address"))) return;
    else throw new InvalidDeliveryInfoException();
}

1 usage  thanhhd
public boolean validatePhoneNumber(String phoneNumber) {
    if (phoneNumber.length() != 10) return false;
    if (!phoneNumber.startsWith("0")) return false;
    try {
        Integer.parseInt(phoneNumber);
    } catch (NumberFormatException e) {
        return false;
    }
    return true;
}

1 usage  thanhhd
public boolean validateName(String name) {
    if (Objects.isNull(name)) return false;
    String patternString = "[a-zA-Z\\s]*";
    Pattern pattern = Pattern.compile(patternString);
    Matcher matcher = pattern.matcher(name);
    return matcher.matches();
}

```

lớp PlaceOrderController vi phạm SRP, do các method validate xử lý vấn đề không liên quan đến PlaceOrder, nên gộp các method ấy thành 1 class riêng.

```

public DeliveryInfo getDeliveryInfo() { return deliveryInfo; }

1 usage  thanhhd
public void setDeliveryInfo(DeliveryInfo deliveryInfo) {
    this.deliveryInfo = deliveryInfo;
    this.shippingFees = deliveryInfo.calculateShippingFee( order: this);
}

```

lớp Order.java vi phạm SRP vì method DeliveryInfo không được viết bên trong class DeliveryInfo

```

PaymentTransaction extractPaymentTransaction(String responseText) {
    MyMap response = convertJSONResponse(responseText);

    if (response == null)
        return null;
    MyMap transaction = (MyMap) response.get("transaction");
    CreditCard card = new CreditCard(
        (String) transaction.get("cardCode"),
        (String) transaction.get("owner"),
        (String) transaction.get("dateExpired"),
        Integer.parseInt((String) transaction.get("cvvCode")));

    PaymentTransaction trans = new PaymentTransaction(
        (String) response.get("errorCode"),
        card,
        (String) transaction.get("transactionId"),
        (String) transaction.get("transactionContent"),
        Integer.parseInt((String) transaction.get("amount")),
        (String) transaction.get("createdAt"));

    switch (trans.getErrorCode()) {
        case "00":
            break;
        case "01":
            throw new InvalidCardException();
        case "02":
            throw new NotEnoughBalanceException();
        case "03":
            throw new InternalServerErrorException();
        case "04":

```

lớp InterbankPayloadConverter.java vi phạm SRP vì các method getToday, extractPaymentTransaction không liên quan đến nhiệm vụ của lớp. 2 method này nên đc tách ra thành class riêng

```

1 usage 2 overrides thanhld
public List getAllMedia() throws SQLException {
    Statement stm = AIMSDB.getConnection().createStatement();
    ResultSet res = stm.executeQuery( sql: "select * from Media");
    ArrayList medium = new ArrayList<>();
    while (res.next()) {
        Media media = new Media(
            res.getInt( columnLabel: "id"),
            res.getString( columnLabel: "title"),
            res.getInt( columnLabel: "quantity"),
            res.getString( columnLabel: "category"),
            res.getString( columnLabel: "imageUrl"),
            res.getInt( columnLabel: "price"),
            res.getString( columnLabel: "type"));
        medium.add(media);
    }
    return medium;
}

```

```

1 usage 3 overrides thanhld
public Media getMediaById(int id) throws SQLException {
    String sql = "SELECT * FROM Media ";
    Statement stm = AIMSDB.getConnection().createStatement();
    ResultSet res = stm.executeQuery(sql);

    if (res.next()) {
        return new Media(
            res.getInt( columnLabel: "id"),
            res.getString( columnLabel: "title"),
            res.getInt( columnLabel: "quantity")

```

lớp MediaDao vi phạm SRP vì lớp này thực hiện quá nhiều nhiệm vụ.

```

public class ApplicationProgrammingInterface {

    no usages
    public static DateFormat DATE_FORMATTER = new SimpleDateFormat( pattern: "yyyy/MM/dd HH:mm:ss");
    4 usages
    private static Logger LOGGER = Utils.getLogger(Utils.class.getName());

    tanhlid
    public static String get(String url, String token) throws Exception {
        LOGGER.info( msg: "Request URL: " + url + "\n");
        HttpURLConnection conn = setupConnection(url);

        conn.setRequestMethod("GET");
        conn.setRequestProperty("Authorization", "Bearer " + token);
        BufferedReader in = new BufferedReader(new InputStreamReader(conn.getInputStream()));
        String inputLine;
        StringBuilder response = new StringBuilder(); // isng StringBuilder for the sake of memory and performance
        while ((inputLine = in.readLine()) != null)
            System.out.println(inputLine);
        response.append(inputLine + "\n");
        in.close();
        LOGGER.info( msg: "Response Info: " + response.substring(0, response.length() - 1).toString());
        return response.substring(0, response.length() - 1).toString();
    }
}

```

lớp ApplicationProgrammingInterface vi phạm SRP vì thực hiện quá nhiều nhiệm vụ

```

private String md5(String message) {
    String digest = null;
    try {
        MessageDigest md = MessageDigest.getInstance( algorithm: "MD5");
        byte[] hash = md.digest(message.getBytes(StandardCharsets.UTF_8));
        // converting byte array to Hexadecimal String
        StringBuilder sb = new StringBuilder( capacity: 2 * hash.length);
        for (byte b : hash) {
            sb.append(String.format("%02x", b & 0xff));
        }
        digest = sb.toString();
    } catch (NoSuchAlgorithmException ex) {
        Utils.getLogger(Utils.class.getName());
        digest = "";
    }
    return digest;
}
}

```

AuthenticationController.java vi phạm SRP, vì các method login, logout, getMainUser, isAnonymousSession liên quan đến authentication. Nhưng method md5 lại không liên quan. Do đó nên tách nó ra thành một class riêng thuộc gói Utils.

```

public class CDDAO extends MediaDAO {

1 usage  ▲ thanhd
    @Override
    public Media getMediaById(int id) throws SQLException {
        String sql = "SELECT * FROM " +
            "aims.CD " +
            "INNER JOIN aims.Media " +
            "ON Media.id = CD.id " +
            "where Media.id = " + id + ";";

        ResultSet res = AIMSDB.getConnection().createStatement().executeQuery(sql);
        if(res.next()) {

            // from media table
            String title = "";
            String type = res.getString( columnLabel: "type");
            int price = res.getInt( columnLabel: "price");
            String category = res.getString( columnLabel: "category");
            int quantity = res.getInt( columnLabel: "quantity");

            // from CD table
            String artist = res.getString( columnLabel: "artist");
            String recordLabel = res.getString( columnLabel: "recordLabel");
            String musicType = res.getString( columnLabel: "musicType");
            Date releasedDate = res.getDate( columnLabel: "releasedDate");

            return new CD(id, title, category, price, quantity, type,
                artist, recordLabel, musicType, releasedDate);
        } else {

```

CDDao vi phạm OCP vì khi cần thêm 1 sản phẩm thì cần thêm hàm getMediaById.

```

public class BookDAO extends MediaDAO {

    1 usage 1 thanhld
    @Override
    public Media getMediaById(int id) throws SQLException {
        String sql = "SELECT * FROM " +
            "aims.Book " +
            "INNER JOIN aims.Media " +
            "ON Media.id = Book.id " +
            "where Media.id = " + id + ",";
        Statement stm = AIMSDB.getConnection().createStatement();
        ResultSet res = stm.executeQuery(sql);
        if(res.next()) {

            // from Media table
            String title = "";
            String type = res.getString( columnLabel: "type");
            int price = res.getInt( columnLabel: "price");
            String category = res.getString( columnLabel: "category");
            int quantity = res.getInt( columnLabel: "quantity");

            // from Book table
            String author = res.getString( columnLabel: "author");
            String coverType = res.getString( columnLabel: "coverType");
            String publisher = res.getString( columnLabel: "publisher");
            Date publishDate = res.getDate( columnLabel: "publishDate");
            int numOfPages = res.getInt( columnLabel: "numOfPages");
            String language = res.getString( columnLabel: "language");
            String bookCategory = res.getString( columnLabel: "bookCategory");

            return new Book(id, title, category, price, quantity, type,

```

BookDao vi phạm OCP vì khi ta muốn thêm sản phẩm thì cần thêm hàm getMediaById