

# Issue Analysis Report – Event Processing System Refactoring

Student Name: Mu'men R Hijazi

Course: Advanced Topics in Software Development

## 1. Introduction

الموجودة في الكود الأصلي لنظام معالجة الأحداث قبل (Design Issues) يهدف هذا التقرير إلى تحليل المشاكل التصميمية (Refactoring). عملية إعادة هيكلة. هذه المشاكل تتعلق بالتصميم المعماري وجودة الكود، وليس أخطاء تشغيل أو ترجمة.

## 2. Overview of the Original Design

يقوم بجميع المهام (EventProcessor) الكود الأصلي كان يعتمد بشكل أساسي على كلاس واحد مركزي التخزين في قاعدة البيانات، وإرسال الإشعارات، مثل التحقق من الحدث، تنفيذ المنطق، بناء الـ payload. هذا الأسلوب أدى إلى تصميم متشابك وصعب التوسيعة.

## 3. Identified Design Issues

### 3.1 Single Responsibility Principle (SRP) Violation

الشرح:

كان مسؤولاً عن أكثر من وظيفة في نفس الوقت EventProcessor كلاس، مما جعله كبير الحجم وصعب الصيانة، وأي تعديل صغير فيه قد يؤثر على أجزاء أخرى من النظام.

Resolution:

للمعالجة تم فصل المسؤوليات إلى عدة كائنات مستقلة مثل Pipeline، للتعامل مع أنواع الأحداث، و Subscribers للإشعارات، و Handlers.

### 3.2 Open/Closed Principle (OCP) Violation

الشرح:

للتعامل مع أنواع الأحداث المختلفة if / switch / الكود الأصلي استخدم شروط وبالتالي إضافة نوع حدث جديد كانت تتطلب تعديل الكود الموجود.

Resolution:

حيث يتم فصل منطق كل نوع حدث في كلاس مستقل Strategy Pattern تم تطبيق وبالتالي يمكن إضافة أنواع جديدة بدون تعديل الكود الأساسي.

### 3.3 Tight Coupling

:الشرح

كان هناك اعتماد مباشر على كائنات محددة مثل قاعدة البيانات والـ Logger، وهذا يقلل من مرونة النظام ويصعب عملية الاختبار.

Resolution:

بين المكونات (Interfaces) لتقليل الترابط (Coupling) تم استخدام.

### 3.4 Rigid Payload Construction

:الشرح

كان مدمجاً داخل منطق المعالجة Payload بناء الـ، مما جعل إضافة خصائص جديدة مثل التشفير أو الضغط أمراً معقداً.

Resolution:

بشكل مرن وдинاميكي Decorator Pattern لبناء الـ Payload تم استخدام.

### 3.5 Inefficient Resource Management

:الشرح

الوصول إلى قاعدة البيانات لم يكن منظماً ولم يستخدم أسلوب إعادة استخدام الموارد.

Resolution:

لتحسين إدارة الموارد Repository و Proxy تم تطبيق Connection Pool.

## 4. Conclusion

عملية إعادة الهيكلة حسّنت تصميم النظام بشكل كبير من حيث التنظيم قابلية التوسيع، وسهولة الصيانة، وذلك من خلال تطبيق مبادئ هندسة البرمجيات الصحيحة.