**Oject**

Nếu hệ thống chỉ sử dụng Singleton và các lớp DTO (Data Transfer Object), và không có các đối tượng khác trong hệ thống, thì việc vẽ Object Diagram có thể không cần thiết.

Object Diagram thường được sử dụng để biểu diễn các đối tượng cụ thể và mối quan hệ giữa chúng trong một thời điểm cụ thể. Tuy nhiên, nếu hệ thống chỉ sử dụng Singleton và các lớp DTO, và không có các đối tượng khác (không có đối tượng được tạo ra hay tương tác), thì Object Diagram không cung cấp nhiều thông tin hữu ích và có thể không cần thiết trong trường hợp này.

Trong trường hợp đơn giản như vậy, việc tập trung vào việc thiết kế và vẽ Class Diagram (biểu diễn cấu trúc và mối quan hệ giữa các lớp) có thể đủ để hiểu cấu trúc và luồng dữ liệu trong hệ thống của bạn. **Package**

Package diagram là một trong các loại biểu đồ UML để mô tả cấu trúc phân chia thành các gói (package) trong phần mềm.

Trong các dự án phần mềm lớn hơn và phức tạp hơn, việc sử dụng package diagram có thể hữu ích để quản lý cấu trúc của phần mềm và giúp cho việc phân tích, thiết kế, phát triển và bảo trì phần mềm trở nên dễ dàng và hiệu quả hơn.

Package diagram phù hợp với các phần mềm có cấu trúc lớn, phức tạp, có nhiều thành phần, mô-đun và chức năng khác nhau ví dụ như các hệ thống thông tin doanh nghiệp (ERP), các hệ thống quản lý cơ sở dữ liệu, các hệ thống phân tán và các hệ thống nhúng.

Tuy nhiên, trong một số trường hợp, nó không cần thiết và có thể không được sử dụng cho các dự án phần mềm đơn giản, bao gồm phần mềm đặt vé xem phim.

Lý do là vì khi xây dựng một phần mềm đơn giản, không có quá nhiều thành phần hoặc mô-đun cần được phân chia thành các gói riêng biệt. Thay vào đó, các thành phần chính của phần mềm đặt vé xem phim có thể được đặt trong một tệp hoặc một thư mục đơn giản.

Hơn nữa, trong các dự án phần mềm nhỏ và đơn giản, việc sử dụng các biểu đồ khác của UML như biểu đồ lớp và biểu đồ tuần tự có thể đủ để mô tả được cấu trúc và chức năng của phần mềm.

**Component**

Component diagram là một trong các loại biểu đồ UML để mô tả cấu trúc phần mềm dưới dạng các thành phần (component) và các liên kết giữa chúng.

Component diagram phù hợp với các phần mềm có kiến trúc phức tạp và được chia thành các thành phần rõ ràng. Nó cũng phù hợp với các phần mềm có độ phân tán cao và phải hoạt động trên nhiều hệ thống khác nhau. Các phần mềm web, hệ thống mạng, hệ thống đám mây và các hệ thống phân tán khác thường được mô hình hóa bằng component diagram.

Ngoài ra, component diagram còn phù hợp với các phần mềm có tính mô đun cao và sử dụng các thư viện, gói thùng hoặc các module khác để tách các chức năng riêng biệt. Với những phần mềm có cấu trúc rõ ràng, component diagram sẽ giúp cho các nhà phát triển dễ dàng hơn trong việc phát triển, kiểm thử, bảo trì và nâng cấp phần mềm.

Tuy nhiên, component diagram không phù hợp với các phần mềm đơn giản, có cấu trúc đơn giản và không phân tán. Các phần mềm nhỏ, đơn giản, chỉ chạy trên một máy tính hoặc một thiết bị không cần sử dụng component diagram.

Ứng dụng đặt vé xem phim thường chỉ chạy trên một máy chủ đơn giản và không cần phải tách thành các thành phần rõ ràng. Việc sử dụng component diagram trong trường hợp này có thể làm cho biểu đồ trở nên phức tạp và không cần thiết.

Ngoài ra, các phần mềm đơn giản và không có độ phân tán cao thường không cần phải sử dụng các biểu đồ phức tạp như component diagram. Thay vào đó, các biểu đồ khác trong UML như use case diagram, class diagram, sequence diagram hay activity diagram có thể đáp ứng được các yêu cầu mô hình hóa và phân tích cho ứng dụng này.

Vì vậy, component diagram không cần thiết cho phần mềm đặt vé xem phim vì đơn giản nó không phù hợp với một ứng dụng đơn giản và không có kiến trúc phức tạp.

**Deployment**

Deployment diagram là một trong các loại biểu đồ UML để mô tả cách thức triển khai (deploy) phần mềm trên các máy chủ, thiết bị hoặc hệ thống.

Deployment diagram được sử dụng để mô tả cách triển khai phần mềm trên các máy chủ và hệ thống, đặc biệt là các phần mềm phức tạp và phân tán. Tuy nhiên, trong trường hợp phần mềm đặt vé xem phim, đây là một ứng dụng đơn giản và không có tính phân tán cao. Vì vậy, deployment diagram không cần thiết cho phần mềm này.

Ứng dụng đặt vé xem phim thường chỉ chạy trên một máy chủ duy nhất và không cần phải triển khai trên nhiều hệ thống hoặc máy chủ khác nhau. Do đó, việc sử dụng deployment diagram sẽ không giúp ích gì cho việc phát triển và triển khai ứng dụng này.

Ngoài ra, các phần mềm đơn giản và không phân tán cao thường không cần phải sử dụng các biểu đồ phức tạp như deployment diagram. Thay vào đó, các biểu đồ khác trong UML như use case diagram, class diagram, sequence diagram hay activity diagram có thể đáp ứng được các yêu cầu mô hình hóa và phân tích cho ứng dụng này.