Part4

افكسوا دلوقتي لي (13-14)

12) **Are there pairs of Software Quality Factors that are not independent in your system? Give an example.**

In we system, **performance** and **security** may not be independent. For instance, when a user tries to pay using their visa card, adding more security checks (e.g., card validation and activity monitoring) can slow down the payment process, affecting performance.

---------------------------------------------------------------------

ده مش الحل ولكن لما نخلص نبقي نعمل الجزء 13 علشان مش هينفع علشان لازم نعرفه الفاكشن والحاجات الاستخدمناها هي ايه من حيث if /for كانوا فين حاجة غريبة اوي يعني حوار for if ده بس اشطا

13) **Calculate the LOC and CCM (Cyclomatic Complexity Metric) for the main functions in your system.**

**LOC (Lines of Code)**: هنا المفروض نحسب الأسطر التي تُجري عمليات التحقق وتُقدم الطلب.

**CCM (Cyclomatic Complexity Metric)**: حسب هذا المقياس عدد المسارات المستقلة عبر الكود. إذا كانت الدالة تحتوي على شروط (if/else) أو حلقات برمجية أو مفاتيح، فستحتاج إلى حساب مسارات التنفيذ الممكنة من خلال هذه الشروط.

--------------------------------------------------------------------

14)For the classes in your system, calculate all the following OO Complexity Metrics:

) WMC (الطرق المرجحة لكل فئة): احسب مجموع تعقيد الطرق في كل فئة. على سبيل المثال، في فئة UserProfile، قد يكون لديك طرق مثل addVisa() و checkBalance() وما إلى ذلك، وتحسب تعقيد كل منها.

ب) DIT (عمق شجرة الوراثة): يقيس هذا العمق. إذا ورثت فئة Admin من فئة User، فسيكون العمق 1. إذا كانت هناك فئة فرعية أخرى ترث من Admin، فسيكون العمق 2.

ج) NOC (عدد الأبناء): يحسب هذا العمق عدد الفئات الفرعية التي ترث من فئة. على سبيل المثال، إذا ورث كل من Admin و Customer من User، فسيكون NOC 2.

د) CBO (الاقتران بين الكائنات): يقيس هذا العمق عدد الفئات الأخرى التي يتفاعل معها فئة. على سبيل المثال، قد تتفاعل فئة TicketBooking مع TrainSchedule و UserProfile و PaymentGateway.

هـ) استجابة الفئة (RFC): يقيس هذا عدد الدوال التي يمكن استدعاؤها من فئة معينة. على سبيل المثال، قد يحتوي UserProfile على دوال مثل addVisa() وgetBalance() وupdateProfile().

و) LCOM (عدم ترابط الدوال): يقيس هذا الترابط بين الدوال في الفئة. إذا كانت فئة Admin تحتوي على دوال غير مرتبطة مثل updateSchedule() وremoveUser()، فسيكون LCOM أعلى.

**15) Considering White-Box Testing, generate a Unit-Testing Test Report for at least 6 main functions in your system.**

**Path Testing**: You need to test all possible execution paths in each function. For example, in the bookTicket() function, you should test different input cases such as:

One-way ticket.

Round-trip ticket.

Completed vs. incomplete trip.

Payment failure scenarios.

**Example for functions with path testing:**

**bookTicket()**: Test input cases like:

One-way ticket.

Round-trip ticket.

Active or inactive trip.

Payment failure.

**getBalance()**: Test cases like:

Sufficient balance.

Insufficient balance.

Server connection failure.

**16) Considering Black-Box Testing, generate a Functionality System-Testing Test Report for at least 6 main functions in your system.**

**Boundary Testing**: You will test input boundary values. For instance:

**Ticket Booking**:

Minimum days between ticket bookings (e.g., booking a ticket the next day).

Maximum days between bookings (e.g., booking a ticket a year in advance).

**Visa Balance Check**:

Test with a zero balance.

Test with a very high balance.

**Test report:**

**Function 1: bookTicket()**:

Input cases: One-way ticket, Round-trip ticket.

Boundary testing: Ensure proper handling of booking dates.

**Function 2: getBalance()**:

Input cases: Sufficient balance, Insufficient balance.

Boundary testing: Check for payment with low balance.