Lab 3\_ part 1

**Testing requirement**

By [traltb@fe.edu.vn](mailto:traltb@fe.edu.vn)

Bên dưới là danh sách các lỗi phổ biến trong hệ thống phần mềm Đặt đồ ăn trực tuyến, các lỗi hay gặp và cách sửa các lỗi này.

1. The system shall be user-friendly.
2. The food delivery must be fast and reliable.
3. Customers should be able to order their favorite meals easily.
4. The application shall work on all devices without any issues.
5. The system will use a secure method to store customer information.
6. The app should have a modern and attractive interface.
7. Users can pay through different methods like credit cards, PayPal, and others.
8. The restaurant list shall be always up to date.
9. The system shall automatically detect and recommend meals based on user preferences and mood.
10. All users will be happy with the ordering experience.
11. The app shall never crash under any circumstances.
12. The code shall be written in a clean and maintainable way.
13. The system will handle a large number of users effectively.
14. The ordering process shall be intuitive and straightforward.
15. The system should process payments quickly.
16. The system shall allow customers to change or cancel orders when needed.
17. The app will support multiple languages fluently.
18. The system shall provide the best restaurants near the user.
19. The user shall be able to track their order in a good way.
20. All deliveries will be on time, regardless of traffic or weather conditions.

**📌 Các lỗi phổ biến có trong những yêu cầu này**

| **STT** | **Dạng lỗi** | **Ví dụ** |
| --- | --- | --- |
| 1 | Mơ hồ (ambiguous) | "user-friendly", "modern", "good way", "fast", "reliable" |
| 2 | Không kiểm chứng được (unverifiable) | "happy with the experience", "never crash", "best restaurants" |
| 3 | Kết hợp nhiều ý (compound requirement) | #7, #9, #16 |
| 4 | Không rõ ràng hoặc không định lượng (non-specific) | "work on all devices", "large number of users", "quickly" |
| 5 | Không thực tế (unrealistic) | "never crash", "always up to date", "on time regardless of traffic" |
| 6 | Mô tả thiết kế thay vì yêu cầu (design vs. need) | "code shall be written in a clean and maintainable way" |

## ****20 Rewritten “Good” Requirements for a Food Ordering System****

1. The system shall allow users to complete an order in no more than 5 steps from cart to checkout.
2. The system shall display confirmation of successful order placement within 2 seconds after checkout.
3. Users shall be able to filter food items by category, price range, dietary preference, and delivery time.
4. The mobile application shall support Android 10+ and iOS 13+ operating systems.
5. Customer passwords shall be stored using SHA-256 hashing and never saved in plain text.
6. The system shall support three payment methods: Visa, PayPal, and cash on delivery.
7. The list of restaurants shall be updated once every 24 hours via an automated API sync.
8. The application shall allow users to set dietary preferences, and recommend at least 3 meals matching the preference.
9. After order placement, the system shall provide real-time order tracking with status updates at each delivery stage.
10. The system shall provide support for English and Vietnamese, based on the user’s device language settings.
11. Users shall be able to cancel an order within 3 minutes after placement, if the restaurant has not yet accepted it.
12. The system shall process 95% of payments within 3 seconds after user confirmation.
13. Order history shall be stored for 6 months and accessible from the user profile page.
14. The system shall respond to user input (e.g., menu clicks, filters) within 1 second in 90% of cases under normal load.
15. The system shall allow users to rate each order on a 5-star scale and submit optional written feedback.
16. The mobile app shall occupy no more than 100 MB of device storage after installation.
17. The application shall log all failed payment transactions with timestamp and failure reason.
18. Restaurants shall be sorted by delivery time by default, with optional sort by rating or price.
19. Users shall receive a notification if delivery is delayed more than 10 minutes from estimated time.
20. The system shall allow users to save up to 5 delivery addresses per account.

* **Lưu ý**:
  + Các yêu cầu này được viết theo cấu trúc rõ ràng: **[Subject] shall [Action] [Condition/Constraint]**.
  + Mỗi yêu cầu **chỉ diễn tả một hành vi cụ thể**, tránh nhập nhiều ý.

**Exercise 1: Explain types of review.**

### **1. Inspection**

**Explanation**: The most formal and detailed type of review. It involves a team (author, reviewers, moderator, recorder) to find detailed defects and ensure the product meets technical requirements.

**Participants**:

**Author**: Creator of the work product (e.g., code, design).

**Reviewers**: Team members with relevant expertise.

**Moderator**: Leads the review, ensuring the process runs smoothly.

**Recorder**: Documents the issues found during the review.

**Characteristics**:

Strict process with detailed checklists.

Thorough examination for defects, adherence to standards, and potential improvements.

Time-consuming but highly effective in identifying serious issues.

**Expected Outcome**:

Detailed feedback on defects.

A report listing identified issues and recommendations for fixes.

Improved product quality through early defect identification.

### **2. Technical Review**

**Explanation**: A structured review where technical experts examine aspects like design, code, and technical details of the product to ensure it meets technical requirements.

**Participants**:

**Author**: Creator of the product or feature being reviewed.

**Technical Experts**: Developers, architects, and other specialists.

**Moderator**: Guides the review and keeps the focus on technical aspects

**Characteristics**:

Focuses on technical aspects such as architecture, design, and functionality.

Less formal than **Inspection**, but still structured.

Allows experts to provide technical insights and suggestions.

**Expected Outcome:**

Technical feedback and suggestions for improvements.

Identification of technical flaws, risks, or limitations.

A product that meets the necessary technical specifications.

### **3. Walkthrough**

**Explanation**: A less formal review where the author presents the product to the team and explains design, coding, or structural decisions.

**Participants**:

**Author**: Creator of the product or part being reviewed.

**Team Members**: Developers, testers, and other stakeholders who need to understand the product.

**Characteristics**:

Informal and focused on knowledge sharing rather than defect detection.

Author explains design and coding decisions.

Primarily aimed at understanding and improving the product.

**Expected Outcome**:

Shared understanding of the product's design.

Opportunity for team members to ask questions or suggest improvements.

Clarification of design or implementation choices.

Greater alignment within the team regarding the product.

### **4. Peer Review**

**Explanation**: An informal review where colleagues or team members review each other's work. It’s often used to catch issues early in development without a formal meeting.

**Participants:**

**Author**: Person whose work is being reviewed.

**Peers**: Colleagues or team members with knowledge of the subject matter.

**Characteristics**:

Informal and typically not done in a formal meeting.

Collaborative approach to identify issues early.

Fast and cost-effective, but may not be as thorough as formal reviews.

**Expected Outcome**:

Feedback on defects and improvement suggestions.

Early identification of issues without formal meetings, saving time and resources.

Improved collaboration and communication among team members.

**Exercise 2: Liệt kê ra các lỗi ở các requirements sau:**

## ****Hệ thống quản lý sinh viên (Student Management System)****

**1.The system shall be easy to use for all teachers.**

**Issue**: The phrase "easy to use" is ambiguous and subjective, as it can mean different things to different users.

**Fix**: The system should have an intuitive user interface that allows teachers to complete tasks with no more than three clicks.

**2.Students should quickly find their grades.**

**Issue**: The word "quickly" is non-specific and does not define a measurable time frame.

**Fix**: Students shall be able to find their grades within five seconds after logging into the system.

**3.The app must always work on any computer.**

**Issue**: The phrase "always work" is unverifiable because it cannot be proven under all possible conditions.

**Fix**: The app shall work on Windows, macOS, and Linux, and function on the latest three major browser versions (Chrome, Firefox, Safari).

**4.User data will be kept safe.**

**Issue**: "Kept safe" is too vague and doesn't provide details on how the data will be protected.

**Fix**: User data will be encrypted using AES-256 encryption and stored in a secure database with access control.

**5.The interface shall be nice and professional.**

**Issue**: The terms "nice" and "professional" are subjective and cannot be measured objectively.

**Fix**: The UI shall follow standard design principles for consistency, accessibility, and ease of use.

**6.Teachers and admins can approve or reject grades and export reports.**

**Issue**: This is a compound requirement that combines multiple actions into one statement.

**Fix**: Teachers and admins shall be able to approve or reject grades. Additionally, they shall be able to export reports in PDF or Excel format.

**7.The grade list shall be updated on time.**

**Issue**: "On time" is vague and doesn't define a specific time frame for updating.

**Fix**: The grade list shall be updated within 24 hours after grades are entered.

**8.All users shall be satisfied with the system.**

**Issue**: It is unverifiable to claim that all users will be satisfied, as satisfaction varies.

**Fix**: At least 80% of users shall rate the system as "satisfactory" or "excellent" in a user satisfaction survey conducted every six months.

1. **The system will never lose any student data.**

**Issue**: It is impossible to guarantee that no data will ever be lost.

**Fix**: The system shall back up all student data every 24 hours to ensure no data loss in the event of failure.

**10.The UI should be responsive and beautiful.**

**Issue**: "Responsive" and "beautiful" are vague terms that are difficult to measure objectively.

**Fix**: The UI shall be fully responsive, adapting to desktop, tablet, and mobile devices, and adhere to modern design standards.

**11.The system shall automatically assign student IDs logically.**

**Issue**: "Logically" is an unclear term that doesn't explain how IDs should be assigned.

**Fix**: The system shall assign student IDs sequentially, starting with the year of enrollment and a unique identifier (e.g., 20230001).

**12.The application shall run well on Windows, macOS, and Linux.**

**Issue**: "Run well" is non-specific and doesn't define what constitutes "well".

**Fix**: The application shall be compatible with the latest versions of Windows 10, macOS 11, and Linux (Ubuntu 20.04), ensuring no performance issues.

**13.Students may access their records from anywhere at any time.**

**Issue**: "Anywhere" and "any time" are unverifiable terms, as they are too broad to test.

**Fix**: Students shall be able to access their records securely via the system’s web portal or mobile app at any time with a stable internet connection.

**14.The code must be readable and easy to maintain.**

**Issue**: This is more of a design specification than a functional requirement.

**Fix**: The code shall follow industry best practices for readability and be well-documented with inline comments explaining complex logic.

**15.Teachers can grade students and notify parents instantly.**

**Issue**: This is a compound requirement, as it combines two separate actions (grading and notifying parents) into one.

**Fix**: Teachers shall be able to grade students. The system shall also notify parents within five minutes after grading.

**16.The system should work fast with a large number of students.**

**Issue**: "Fast" is non-specific and needs to be measurable.

**Fix**: The system shall handle at least 10,000 students concurrently, with response times of no more than two seconds per action.

**17.It shall detect possible errors in transcripts intelligently.**

**Issue**: "Intelligently" is an ambiguous term that isn't clearly defined.

**Fix**: The system shall detect errors in transcripts, such as duplicates or missing grades, using a predefined error detection algorithm.

**18.The export feature shall support Excel, PDF, CSV, and more.**

**Issue**: "And more" is non-specific and undefined.

**Fix**: The export feature shall support Excel (.xlsx), PDF, and CSV formats, with additional formats available upon request.

**19.The app shall always show the best format for transcripts.**

**Issue**: "The best format" is subjective and cannot be tested in all situations.

**Fix**: The app shall automatically select the most suitable format for displaying transcripts based on the user’s device and screen size.

**20.It should never crash under any condition.**

**Issue**: It is unverifiable to claim that the system will never crash under all conditions.

**Fix**: The system shall have a crash rate of no more than 0.1% during high usage or extreme conditions.