Lab 3\_ part 1

**Testing requirement**

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

**Name: Tô Đình Văn – DE180061 – SE18D01.**

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**

* **Definition:** A formal, structured process of examining documents or code in detail to find defects.
* **Purpose:** To detect errors, inconsistencies, or omissions in software products or documents.
* **Characteristics:**
  + Roles are clearly defined (moderator, author, reviewers, recorder).
  + Follows a strict checklist or process.
  + Detailed recording and verification of defects.
* **Advantages:** Thorough error detection and early quality improvement.
* **Example:** Inspecting source code or requirement documents.

**2. Technical Review**

* **Definition:** A review focusing on the technical content of a product or document, carried out by technical experts.
* **Purpose:** To check technical feasibility, correctness, and compliance with standards.
* **Characteristics:**
  + Conducted by experienced technical team members.
  + Less formal than inspection but still focused on technical issues.
  + Provides suggestions for technical improvements.
* **Advantages:** Helps ensure technical accuracy and efficiency.
* **Example:** Reviewing software architecture or system design.

**3. Walkthrough**

* **Definition:** An informal presentation by the author of a document or code to the team for feedback.
* **Purpose:** To collect feedback, catch early defects, and ensure mutual understanding.
* **Characteristics:**
  + Usually informal, may or may not have a moderator.
  + Team members ask questions and give comments freely.
  + Often used as a learning or communication tool.
* **Advantages:** Improves team understanding and early defect detection.
* **Example:** Author presenting design documents to developers.

**4. Peer Review**

* **Definition:** A process where colleagues at the same level review each other's work.
* **Purpose:** To quickly find defects and improve quality.
* **Characteristics:**
  + Informal and flexible; can be done online or offline.
  + No complex procedures, mainly feedback and discussion.
* **Advantages:** Encourages collaboration and skill development.
* **Example:** Developers reviewing each other’s code before integration.

**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.
2. Students should quickly find their grades.
3. The app must always work on any computer.
4. User data will be kept safe.
5. The interface shall be nice and professional.
6. Teachers and admins can approve or reject grades and export reports.
7. The grade list shall be updated on time.
8. All users shall be satisfied with the system.
9. The system will never lose any student data.
10. The UI should be responsive and beautiful.
11. The system shall automatically assign student IDs logically.
12. The application shall run well on Windows, macOS, and Linux.
13. Students may access their records from anywhere at any time.
14. The code must be readable and easy to maintain.
15. Teachers can grade students and notify parents instantly.
16. The system should work fast with a large number of students.
17. It shall detect possible errors in transcripts intelligently.
18. The export feature shall support Excel, PDF, CSV, and more.
19. The app shall always show the best format for transcripts.
20. It should never crash under any condition.

**ANSWER:**

1. The system shall be easy to use for all teachers.  
   **Error:** Ambiguous, unverifiable  
   **Fix:** The system shall allow teachers to complete grading and attendance tasks in no more than 3 clicks.
2. Students should quickly find their grades.  
   **Error:** Non-specific, vague  
   **Fix:** Students shall be able to access their grades within 5 seconds from the dashboard page.
3. The app must always work on any computer.  
   **Error:** Unrealistic, non-specific  
   **Fix:** The app shall support the latest versions of Chrome, Firefox, and Edge on Windows 10+, macOS 11+, and Ubuntu 20.04+.
4. User data will be kept safe.  
   **Error:** Unverifiable, ambiguous  
   **Fix:** All user data shall be encrypted using AES-256 and stored in a secure cloud environment.
5. The interface shall be nice and professional.  
   **Error:** Ambiguous, unverifiable  
   **Fix:** The interface shall use the institution’s official color scheme and consistent font styles across all pages.
6. Teachers and admins can approve or reject grades and export reports.  
   **Error:** Compound requirement  
   **Fix:** Teachers can approve or reject submitted grades. Admins can export grade reports in selected formats.
7. The grade list shall be updated on time.  
   **Error:** Ambiguous, non-specific  
   **Fix:** The grade list shall be updated within 10 minutes after a teacher finalizes the grades.
8. All users shall be satisfied with the system.  
   **Error:** Unverifiable, ambiguous  
   **Fix:** The system shall achieve at least 85% user satisfaction in post-launch surveys.
9. The system will never lose any student data.  
   **Error:** Unrealistic, unverifiable  
   **Fix:** The system shall back up all student data every 12 hours and retain backups for at least 30 days.
10. The UI should be responsive and beautiful.  
    **Error:** Ambiguous, unverifiable  
    **Fix:** The UI shall adapt to screen sizes from 5” to 27” using responsive web design and pass mobile compatibility testing.
11. The system shall automatically assign student IDs logically.  
    **Error:** Ambiguous  
    **Fix:** The system shall assign student IDs using the format: [year][department code][sequential number].
12. The application shall run well on Windows, macOS, and Linux.  
    **Error:** Non-specific  
    **Fix:** The application shall be fully functional and tested on Windows 10+, macOS 11+, and Ubuntu 20.04+.
13. Students may access their records from anywhere at any time.  
    **Error:** Unrealistic, non-specific  
    **Fix:** Students shall be able to access their records via web portal 24/7, except during scheduled maintenance.
14. The code must be readable and easy to maintain.  
    **Error:** Design description instead of requirement, ambiguous  
    **Fix:** The system shall follow Google Java Style Guide and require inline documentation for all public functions.
15. Teachers can grade students and notify parents instantly.  
    **Error:** Compound requirement, unrealistic  
    **Fix:** Teachers can enter grades. The system shall notify parents via email within 5 minutes of grade submission.
16. The system should work fast with a large number of students.  
    **Error:** Ambiguous, non-specific  
    **Fix:** The system shall respond to dashboard queries in under 2 seconds for up to 10,000 student records.
17. It shall detect possible errors in transcripts intelligently.  
    **Error:** Ambiguous, unverifiable  
    **Fix:** The system shall flag transcripts with missing grades, duplicate courses, or credit totals below graduation requirements.
18. The export feature shall support Excel, PDF, CSV, and more.  
    **Error:** Ambiguous, non-specific  
    **Fix:** The export feature shall support exporting to Excel (.xlsx), PDF (.pdf), and CSV (.csv) formats.
19. The app shall always show the best format for transcripts.  
    **Error:** Ambiguous, unverifiable  
    **Fix:** The app shall display transcripts in a fixed format including course code, title, credit, and grade per subject.
20. It should never crash under any condition.  
    **Error:** Unrealistic, unverifiable  
    **Fix:** The system shall handle unexpected errors gracefully, log all failures, and auto-restart within 30 seconds when possible.