**Lesson 6 Levels and types of testing**

Level 1

**1. Make a comparative table of functional, non-functional, and change-related types of testing.**

**The comparison should contain the following blocks:**

* **what is being checked;**
* **when applicable;**
* **restrictions;**
* **peculiarities.**

| Aspect | Functional Testing | Non-Functional Testing | Change Related Types of Testing |
| --- | --- | --- | --- |
| What is being checked | The behavior of individual functions, features, or components against specified requirements. | Performance, usability, security, and other non-functional aspects of the system. | Impact of changes (code updates, bug fixes, enhancements) on existing functionality. |
| When applicable | Throughout the development lifecycle, to validate the requirements. | At various stages, often after functional testing, to assess system qualities. | After code changes, updates, or modifications to ensure they didn't break existing functionality. |
| Restrictions | Focuses on specific functional aspects and user interactions. | Focuses on performance, scalability, reliability, and other non-functional attributes. | It focuses on the specific changes and their impact, so it may only cover some aspects of the application. |
| Peculiarities | Test cases are derived directly from documented requirements. | Test cases may be derived from performance standards, usability guidelines, or security protocols. | Test cases are often tailored to verify the specific changes introduced. |

**2. Explain the difference between regression and retesting (5 sentences).**

Regression testing and retesting serve different purposes and are conducted at different stages of the software development lifecycle. Regression testing verifies that the entire system functions correctly after any changes while retesting focuses on confirming that specific issues have been fixed. Both testing types are essential for delivering high-quality software, as regression testing helps prevent new defects, and retesting ensures that known issues are fixed.

Level 2

1. **Do you think it is possible to perform only functional testing for a product without checking non-functional requirements?**

* **If yes - in what cases?**
* **If not, why not?**

**Support your answer.**

Performing functional testing for a product without explicitly checking non-functional requirements is possible. Still, this approach has limitations and may not comprehensively assess the product's quality. The cases where only functional testing can be performed would be the early development stages or when a project has a limited scope with well-defined features.

At the same time, performing only functional testing can lead to an incomplete product assessment, such as poor performance and security vulnerabilities, impacting the user experience. Neglecting non-functional requirements may put a product at a disadvantage in the market. While performing only functional testing in certain situations is possible, it should be done carefully considering the project's context.

A well-balanced testing approach that includes both functional and non-functional testing ensures that the software meets its functional requirements and delivers a satisfactory overall user experience while minimizing the risk of late discovery of critical issues.

1. **How do you see the need for smoke testing? Is it always appropriate?**

Smoke testing is a testing technique used to quickly assess the overall stability of a software build or application. It helps ensure the basic functionality and stability of a software build or release. While it's typically appropriate and beneficial, its use should be evaluated in the context of the project's goals, constraints, and development processes to determine when and how it should be applied.

Level 3

**1. You are the founder of a startup planning to launch a mobile application for sharing cat photos on iOS and Android devices.**

**Users can upload photos of cats, but they cannot upload photos of other animals/people/objects. Users can add friends and leave likes and comments.**

**Write five functional test cases that would test the application.**

| Test Case 1 | User Registration |
| --- | --- |
| Tested Data | To verify that users can successfully register for the application. |
| Test Steps | 1. Launch the app. 2. Click on the "Sign Up" or "Register" button. 3. Enter valid registration details (username, email, password). 4. Click on the "Register" button. |
| Expected Result | The user should be registered successfully and redirected to the home feed. |

| Test Case 2 | Cat Photo Upload |
| --- | --- |
| Tested Data | To confirm that users can upload cat photos and that non-cat photos are not accepted. |
| Test Steps | 1. Log in to the app. 2. Navigate to the "Upload Photo" feature. 3. Select a cat photo for upload. 4. Click on the "Upload" button. |
| Expected Result | The cat photo should be uploaded successfully. Attempting to upload a non-cat photo should result in an error or rejection. |

| Test Case 3 | Add Friend |
| --- | --- |
| Tested Data | To ensure users can add friends to their network. |
| Test Steps | 1. Log in to the app. 2. Access the user's profile or settings. 3. Select the "Add Friends" or "Search for Friends" option. 4. Enter the username or email of a friend to add. 5. Click on the "Add Friend" button. |
| Expected Result | The selected friend should receive a friend request, and upon acceptance, they should be added to the user's friend list. |

| Test Case 4 | Like and Comment |
| --- | --- |
| Tested Data | To validate that users can like and comment on cat photos. |
| Test Steps | 1. Log in to the app. 2. Browse the cat photo feed. 3. Select a cat photo. 4. Click on the "Like" button. 5. Add a comment in the comment section. 6. Click on the "Post" or "Submit" button. |
| Expected Result | The like should be registered, and the comment should be displayed below the photo. The user's name and comment should be visible to others. |

| Test Case 5 | User Privacy Settings |
| --- | --- |
| Tested Data | To confirm that users can adjust their privacy settings. |
| Test Steps | 1. Log in to the app. 2. Access the user's profile or settings. 3. Navigate to the "Privacy Settings" section. 4. Adjust privacy settings for photo sharing ( public, private, friends-only). 5. Save the changes. |
| Expected Result | Users' chosen privacy settings should be applied to their uploaded cat photos. For example, if they set photos to "private," only approved friends should be able to view them. |

**2. Write what non-functional requirements you want to apply to your startup's product.**

**Describe the tests that would check them (3-5 examples).**

1. **Performance requirements:**

The app should load cat photos and user feeds within 2 seconds.

The app must provide a responsive user experience, even under heavy load.

1. **Security requirements:**

Implement proper user authentication and authorization mechanisms to ensure users can only access their data.

User data must be securely stored and transmitted. Unauthorized access must be prevented.

1. **Scalability requirements:**

User data must be encrypted and stored securely, including login credentials and uploaded cat photos.

Implement proper user authentication and authorization mechanisms to ensure users can only access their data.

1. **Usability Requirements:**

The app should adhere to best practices for mobile app user interfaces, with intuitive navigation and user-friendly design.