**Рівень 1.**

**Завдання 5.1.**

**Скласти порівняльну таблицю функціонального, нефункціонального і пов’язаного зі змінами видів тестування. Порівняння має містити такі блоки:**

* **що перевіряється**
* **коли застосовується**
* **обмеження**
* **особливості**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Функціональне тестування** | **Нефункціональне тестування** | **Тестування, пов’язане зі змінами** |
| *Що перевіряється* | Функції компонента або системи | Якісні характеристики компонента або системи | Факт виправлення знайденого раніше дефекту (багу) та повноцінне функціонування програми після виправлення |
| *Коли застосовується* | На всіх рівнях тестування | На інтеграційному, системному та приймальному рівнях | Після внесення змін, виправлення дефектів тощо |
| *Обмеження* | Розглядається зовнішня поведінка системи | Розглядається внутрішня логіка ПЗ. | Зосереджено на змінах, що відбулися (чи ні) після виявлення дефекту: виправлений баг чи ні; змінилась функціональність після виправлення багу чи ні тощо |
| *Особливості* | Проводяться першочергово.  Цікавить лише результат роботи системи, а не її проміжні стани: чи відповідає ПЗ визначеним вимогам. | Отримання детальних даних про роботу продукту: наскільки добре працює система загалом. | Підтвердження працездатності програми чи факта виправлення бага. |

**Завдання 5.2.**

**Поясни в чому різниця між регресією та ретестингом.**

**Regression testing** проводиться з метою перевірки працездатності існуючого функціоналу та перевірки на відсутність сторонніх помилок після оновлення ПЗ.

**Retesting** проводиться для підтвердження виправлення помилки та роботи даного функціоналу.

|  |  |
| --- | --- |
| **Regression testing** | **Retesting** |
| виконується тільки при додаванні нової функціональності ПЗ; | виконується в тому ж оточенні й з тими ж даними, але в оновленому ПЗ (на новому білді) |
| можна проводити паралельно з повторним тестуванням; | повторне тестування має вищий пріоритет та має бути виконано до регресійного |
| тест-кейси можуть бути автоматизовані; | тест-кейси не можуть бути автоматизовані |
| тест-кейси, які раніше були відмічені як “пройдені” повинні бути перевірені повторно. | перевіряються лише тест-кейси відмічені як “непройдені” |

**Рівень 2.**

**Завдання 5.3.**

**Як ти вважаєш, чи можливе для продукту проведення тільки функціонального тестування, без перевірки нефункціональних вимог?**

**Якщо так - в яких випадках?**

**Якщо ні - чому?**

**Обґрунтувати відповідь.**

Проведення виключно функціональної перевірки без перевірки нефункціональних вимог неможливе, оскільки функціональна перевірка надасть інформацію щодо того чи працює програма взагалі, але не надасть повноти картини щодо того, чи якісно працює програма, чи виконується в ній внутрішня логіка.

**Рівень 3.**

**Завдання 5.4.**

**Я – засновниця стартапу і планую випустити на ринок мобільний застосунок для обміну світлинами котиків для iOS та Android пристроїв.**

**Користувачі можуть завантажувати фотографії котиків. Але не можуть завантажувати фотографії інших тварин/людей/об’єктів. Користувачі можуть додавати друзів, ставити “вподобайки”, залишати коментарі.**

**Завдання 1: Напиши 5 функціональних тест-кейсів, які перевіряли б роботу застосунку.**

<https://docs.google.com/spreadsheets/d/1Ca7h8CZ53N9pMv36ac6nP3-TlBcpwJm989k6B17rtRo/edit?usp=sharing>

**Завдання 2. Напиши, які нефункціональні вимоги ти хотіла б застосувати для продукту твого стартапу.**

**Опиши перевірки, які б їх перевіряли (3-5 прикладів).**

|  |  |
| --- | --- |
| **Вимога** | **Перевірка** |
| Можливість одночасно завантажувати 10 фотографій | Завантажити 10 фотографій |
| Кількість символів у коментарі не може перевищувати значення 150 одиниць. | Написати коментар з кількістю символів 151 одиниця. |
| Збереження історії чату у разі некоректного відключення пристрою під час користування додатком | Вилучити акумулятор з пристрою під час користування додатком. Увімкнути пристрій та перевірити останнє збережене повідомлення у чаті. |