**Перший рівень**

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

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

| **Критерії порівняння** | **Functional testing** | **Non-functional testing** | **Change-related Testing** |
| --- | --- | --- | --- |
| Що перевіряється | Функції, які система повинна виконувати (What the system should do) | Такі характеристики систем і програмного забезпечення, як зручність використання, ефективність роботи або безпека (“how well” the system behaves). | Відповідність програмного забезпечення вимогам, що виникають у зв'язку зі змінами, що вносяться в систему. |
| Коли застосовується | Функціональні тести повинні виконуватися на всіх рівнях тестування | Нефункціональне тестування може і часто повинно проводитися на всіх рівнях тестування., і робити це потрібно якомога раніше. | Застосовується після внесення змін до програмного забезпечення |
| Обмеження | * Не допомагає виявляти візуальні дефекти: * Не забезпечує повну відповідність стандартам * Не допомагає виявляти проблеми з продуктивністю * Не забезпечує перевірку взаємодії з іншими системами | * Обмежена область застосування * Труднощі у встановленні критеріїв оцінки * Обмежена переносимість результатів * Висока складність виконання * Обмежена автоматизації | * Витрати (може бути дуже витратним процесом) * Час (Час, необхідний для проведення тестів, може бути обмеженим) * Недостатнє тестування (Існує ризик недостатнього тестування після внесення змін, що може призвести до проблем в роботі системи.) * Нестабільність ( може призвести до нестабільної роботи системи або до помилок в роботі.) |
| Особливості | * Орієнтація на користувача * Перевірка функцій * Функціональне тестування зазвичай виконується з використанням підходу тестування "чорного ящика" * Включає в себе тестування бізнес-логіки програми * часто використовується для регресійного тестування | * Тестові кейси створюються на основі нефункціональних вимог * Тестування "білого ящика" * Нефункціональне тестування включає в себе тестування продуктивності, юзабіліті-тестування, безпеки, сумісності | * Аналіз впливу змін: Перед проведенням тестування пов'язаного зі змінами, необхідно визначити, які аспекти програми змінилися, і як ці зміни можуть вплинути на інші частини системи * Використання автоматизованих інструментів |

2. Поясни, в чому різниця між регресією та ретестингом (5 речень).

Регресійне тестування (Regression testing) — це вид тестування, спрямований на перевірку всіх функцій програмного забезпечення після внесення змін, проведених у самому додатку або середовищі (виправлення дефекту, міграція операційної системи, змін у базі даних, налаштуваннях сервера тощо).

Повторне тестування (Re-test) — тестування, яке перевіряє, що дефект виправлений та змінений функціонал працює як треба.

На відміну від повторного тестування, регресійне тестування переважно здійснюється на основі автоматизованих тестів. Регресійне тестування має нижчий пріоритет, оскільки воно проводить розгортку програми для перевірки потенційних непередбачуваних дефектів також часто досліджує великі частини програми для виявлення помилок і тому може зайняти більше часу. Отже, поняття регресійне тестування більш ширше та включає в себе більшу кількість активностей з перевірки функціоналу, в якому відбулися зміни або який був дотичний до них з метою підтвердження, що наявна раніше функціональність працює, як і раніше.

**Другий рівень**

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

Якщо так – в яких випадках?

Якщо ні – чому?

Обґрунтуйте свою відповідь.

На мою думку, недоцільно виконувати лише функціональне тестування, оскільки для забезпечення належної якості розробленого продукту необхідно провести комплексне тестування з використанням як функціонального так і нефункціонального тестування. Адже саме нефункціональне тестування має за мету визначити як добре працює система. Такі як дефекти продуктивності, безпеки системи, надійність - якщо вони не будуть виявлені на ранніх етапах, то можуть призвести до значних грошових затрат, а також витрат часу. І звичайно призведе до некоректної роботи системи або ж до не задоволення потреб кінцевого користувача, в результаті можемо не отримати успішного випуску продукту на ринок.

2. Як ти розумієш необхідність проведення smoke (димового) тестування? Чи завжди воно є доречним?

Smoke тестування використовується для перевірки основних функцій програмного забезпечення і його готовності до більш детального тестування. Основна необхідність проведення smoke тестування полягає в тому, щоб виявити можливі серйозні проблеми в програмному забезпеченні на ранній стадії розробки. Smoke тестування також допомагає виявляти критичні проблеми в програмному забезпеченні, які можуть привести до неможливості подальшого тестування, виробництва та використання програмного забезпечення.

Проте, є випадки, коли Smoke тестування може бути непотрібним або недостатнім. Наприклад, якщо програмний продукт складається з великої кількості функцій, то Smoke тестування може не виявити проблем, що пов'язані з менш важливими функціями продукту. У цьому випадку потрібно виконати більш ретельне функціональне тестування. Отже, хоча Smoke тестування може бути корисним в багатьох випадках, варто враховувати його обмеження та використовувати його у поєднанні з іншими методами тестування.

**Третій рівень**

2. Ти – засновник/ця стартапу і плануєш 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: Напиши 5 функціональних тест-кейсів, які перевіряли б роботу застосунку.

| **Test Case ID** | | 0001 | **Test Case Description** | | Test the ability to send a friend request | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Created By** | | Anastasiia Luzina | **Reviewed By** | | Vasyl Suberliak | | **Version** | **1.0** |
|  | | | | | | | | |
| **Tester's Name** | | Anastasiia Luzina | **Date Tested** | | 21/03/2023 | | **Test Case (Pass/Fail/Not Executed)** | **Pass** |
|  | | | | | | | | |
| **S #** | **Preconditions:** | | |  | **S #** | **Test Data** | | |
| 1 | There must be 2 or more users registered in the system | | | 1 | Login = Anastasia | | |
| 2 | Log in to the app | | | 2 | Password = Password1 | | |
| 3 |  | | | 3 |  | | |
| 4 |  | | | 4 |  | | |
|  | | | | | | | | |
| **Test Scenario** | Verify that the user can send a friend request | | | | | | | |
|  | | | | | | | | |
| **Step #** | **Step Details** | | | | **Expected Results** | | | **Pass / Fail / Not executed / Suspended** |
|
| 1 | Go to the personal profile | | | | Personal profile displayed | | | Pass |
| 2 | Go to the "friends" tab | | | | The tab should be open and the list of recommended friends should be displayed | | | Pass |
| 3 | Click the "add" button | | | | The screen displays a message that I have sent a friend request | | | Pass |

| **Test Case ID** | | 0002 | **Test Case Description** | | Test the ability to add a friend | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Created By** | | Anastasiia Luzina | **Reviewed By** | | Vasyl Suberliak | | **Version** | **1.0** |
|  | | | | | | | | |
| **Tester's Name** | | Anastasiia Luzina | **Date Tested** | | 21/03/2023 | | **Test Case (Pass/Fail/Not Executed)** | **Pass** |
|  | | | | | | | | |
| **S #** | **Preconditions:** | | |  | **S #** | **Test Data** | | |
| 1 | There must be 2 or more users registered in the system | | | 1 | Login = Anna | | |
| 2 | The other user must send a friend request | | | 2 | Password = Password2 | | |
| 3 | Log in to the app | | | 3 |  | | |
| 4 |  | | | 4 |  | | |
|  | | | | | | | | |
| **Test Scenario** | Verify that the user can add a friend | | | | | | | |
|  | | | | | | | | |
| **Step #** | **Step Details** | | | | **Expected Results** | | | **Pass / Fail / Not executed / Suspended** |
|
| 1 | Go to the personal profile | | | | Personal profile displayed | | | Pass |
| 2 | Go to the "friends" tab | | | | The tab should be open  The friend request should be displayed | | | Pass |
| 3 | Click the "confirm " button | | | | A message appears on the screen that the friend request has been confirmed | | | Pass |
| 4 | Back to the personal profile | | | | Personal profile displayed | | | Pass |
| 5 | Go to the "friends" tab | | | | The Friends tab shows a new friend | | | Pass |

| **Test Case ID** | | 0003 | **Test Case Description** | | Test the ability to like publications | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Created By** | | Anastasiia Luzina | **Reviewed By** | | Vasyl Suberliak | | **Version** | **1.0** |
|  | | | | | | | | |
| **Tester's Name** | | Anastasiia Luzina | **Date Tested** | | 21/03/2023 | | **Test Case (Pass/Fail/Not Executed)** | **Pass** |
|  | | | | | | | | |
| **S #** | **Preconditions:** | | |  | **S #** | **Test Data** | | |
| 1 | There must be 2 or more users registered in the system | | | 1 | Login = Anastasiia | | |
| 2 | There must be at least 1 publication by another user | | | 2 | Password = Password1 | | |
| 3 |  | | | 3 |  | | |
| 4 |  | | | 4 |  | | |
|  | | | | | | | | |
| **Test Scenario** | Verify that the user can like publications | | | | | | | |
|  | | | | | | | | |
| **Step #** | **Step Details** | | | | **Expected Results** | | | **Pass / Fail / Not executed / Suspended** |
|
| 1 | Log in to the app | | | | The page with other users' posts is displayed | | | Pass |
| 2 | Press the like button next to the photo | | | | The like button has changed color to red  The number next to the button has increased by +1 | | | Pass |

| **Test Case ID** | | 0004 | **Test Case Description** | | Test the ability to write comments | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Created By** | | Anastasiia Luzina | **Reviewed By** | | Vasyl Suberliak | | **Version** | **1.0** |
|  | | | | | | | | |
| **Tester's Name** | | Anastasiia Luzina | **Date Tested** | | 21/03/2023 | | **Test Case (Pass/Fail/Not Executed)** | **Pass** |
|  | | | | | | | | |
| **S #** | **Preconditions:** | | |  | **S #** | **Test Data** | | |
| 1 | There must be 2 or more users registered in the system | | | 1 | Login = Anastasiia | | |
| 2 | There must be at least 1 publication by another user | | | 2 | Password = Password1 | | |
| 3 |  | | | 3 |  | | |
| 4 |  | | | 4 |  | | |
|  | | | | | | | | |
| **Test Scenario** | Verify that the user can write comments | | | | | | | |
|  | | | | | | | | |
| **Step #** | **Step Details** | | | | **Expected Results** | | | **Pass / Fail / Not executed / Suspended** |
|
| 1 | Log in to the app | | | | The page with other users' posts is displayed | | | Pass |
| 2 | Press the comment button next to the photo | | | | The text input field is opened | | | Pass |
| 3 | Write a comment in the open field | | | | The entered text is displayed on the screen | | | Pass |
| 4 | Click the "Save" button | | | | Comment saved and displayed below the post | | | Pass |

| **Test Case ID** | | 0005 | **Test Case Description** | | Test the ability to upload a photo of the cats | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Created By** | | Anastasiia Luzina | **Reviewed By** | | Vasyl Suberliak | | **Version** | **1.0** |
|  | | | | | | | | |
| **Tester's Name** | | Anastasiia Luzina | **Date Tested** | | 21/03/2023 | | **Test Case (Pass/Fail/Not Executed)** | **Pass** |
|  | | | | | | | | |
| **S #** | **Preconditions:** | | |  | **S #** | **Test Data** | | |
| 1 | Log in to the app | | | 1 | Login = Anastasiia | | |
| 2 |  | | | 2 | Password = Password1 | | |
| 3 |  | | | 3 |  | | |
| 4 |  | | | 4 |  | | |
|  | | | | | | | | |
| **Test Scenario** | Verify that the user can upload a photo of the cats | | | | | | | |
|  | | | | | | | | |
| **Step #** | **Step Details** | | | | **Expected Results** | | | **Pass / Fail / Not executed / Suspended** |
|
| 1 | Go to the personal profile | | | | Personal profile displayed | | | Pass |
| 2 | Click the "Publish" button | | | | The window with my publications is opened | | | Pass |
| 3 | Click the "Add new" button | | | | The gallery with photos is opened | | | Pass |
| 4 | Choose a photo with a cat in the gallery | | | | The photo with the check mark is displayed on the screen | | | Pass |
| 5 | Click the "Confirm" button | | | | Photo uploaded | | | Pass |
| 6 | Click the "Save" button | | | | The photo is published in my profile | | | Pass |

| **Test Case ID** | | 0006 | **Test Case Description** | | Test the option not to upload photos of other animals, people, objects | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Created By** | | Anastasiia Luzina | **Reviewed By** | | Vasyl Suberliak | | **Version** | **1.0** |
|  | | | | | | | | |
| **Tester's Name** | | Anastasiia Luzina | **Date Tested** | | 21/03/2023 | | **Test Case (Pass/Fail/Not Executed)** | **Pass** |
|  | | | | | | | | |
| **S #** | **Preconditions:** | | |  | **S #** | **Test Data** | | |
| 1 | Log in to the app | | | 1 | Login = Anastasiia | | |
| 2 |  | | | 2 | Password = Password1 | | |
| 3 |  | | | 3 |  | | |
| 4 |  | | | 4 |  | | |
|  | | | | | | | | |
| **Test Scenario** | Verify that the user can not upload photos of other animals, people, objects | | | | | | | |
|  | | | | | | | | |
| **Step #** | **Step Details** | | | | **Expected Results** | | | **Pass / Fail / Not executed / Suspended** |
|
| 1 | Go to the personal profile | | | | Personal profile displayed | | | Pass |
| 2 | Click the "Publish" button | | | | The window with my publications is opened | | | Pass |
| 3 | Click the "Add new" button | | | | The gallery with photos is opened | | | Pass |
| 4 | Choose a photo without a cat in the gallery | | | | The photo with the check mark is displayed on the screen | | | Pass |
| 5 | Click the "Confirm" button | | | | Photo uploaded | | | Pass |
| 6 | Click the "Save" button | | | | The system displays an error with the message "This content cannot be published" | | | Pass |

3. Напиши, які нефункціональні вимоги ти хотів/ла б застосувати для продукту твого стартапу.

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

1. **Висока продуктивність додатка**. Для даної вимоги необхідно застосувати Load Testing. Навантаженням буде 10 000 користувачів, які будуть одночасно працювати з додатком. Такий вид тестування дозволяє отримати час відгуку всіх найважливіших бізнес-транзакцій. А також Volume Testing - збільшимо кількість використовуваних даних, які зберігаються і використовуються в додатку.
2. **Можливість використання додатку на IOS та Android.** Для того, щоб перевірити дану вимогу застосуємо Compatibility testing, який дозволить перевірити чи буде працювати наш додаток відповідно до заданих вимог на мобільних операційних системах IOS та Android.
3. **Завантаження головної сторінки після входу в додаток повинна складати 2 секунди.** Перевірка даної вимоги відбувається шляхом створення тест-кейсу і далі виконання перевірки на те, скільки часу додаток використовує для відображення головної сторінки, також можна збільшити кількість користувачів і перевірити використовуючи Load Testing.