## **Рівні та види тестування**

Перший рівень

1.

|  | функціональне | нефункціональне | пов’язане зі змінами |
| --- | --- | --- | --- |
| що перевіряється | тестування всіх функцій системи.   * тестування безпеки * функціональне тестування * тестування взаємодії | направлено на перевірку тих аспектів ПЗ, які можуть бути описані в документації, але не відносяться до функцій програмних продуктів | ПЗ повинне бути перетестоване для підтвердження того факту,що проблема була дійсно вирішена |
| коли застосовується | для підтвердження, що кожна функція програми працює відповідно до документації | Тестування стабільності;  Юзабіліті тестування;  Тестування ефективності;  Тестування ремонтопридатності;  Перевірка портативності;  Приймальне тестування;  Тестування вимог;  Тестування витривалості системи;  Тестування навантаження;  Тестування продуктивності;  Тестування сумісності;  Тестування безпеки;  Об'ємне тестування;  Стрес тестування;  Тестування швидкості відновлення;  Тестування локалізації, інтернаціоналізація | після проведення необхідних змін,таких як виправлення бага/дефекта |
| обмеження | існує ймовірність пропустити кілька помилок логіки програмного забезпечення під час перевірки функціоналу програми | не включає тестування кінцевого користувача | Трудомісткість регресійного тестування, неможливість автоматизації тест-кейсів у ретестах, вимагається додатковий час для проходження вже раніше пройдених тест-кейсів |
| особливості | базується на функціях та особливостях,а також взаємодії з іншими системами і може бути представлене на всіх рівнях тестування: компонентному або модульному, інтеграційному, системному та приймальному | описує тести, необхідні для визнання характеристик ПЗ, які можуть бути виміряні різними величинами | містить види:  Димове тестування-запуск продукту і перевірка основних функцій на працездатність.  Регресійне тестування-перевірка після апдейту старої частини функціоналу.  Ретест-проводиться для підтвердження виправлення помилки та роботи даного функціоналу.  Санітарне -поглиблене тестування однієї функції. |

2.

| **Регресійне** | **Ретест** |
| --- | --- |
| Регресійне тестування виконується тільки при додаванні нової функціональності ПЗ або істотній зміні функціоналу системи. | Ретест виконується в тому ж оточенні й з тими ж даними, але на новому білді. |
| Регрес 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. |
| В рамках регресійного тестування тест-кейси, які були відмічені раніше як «Passed», повинні бути перевірені повторно. | В рамках повторного тестування (ретест) перевіряються тест-кейси тільки зі статусом «Failed». |

Другий рівень

1. +
2. Вважаю,що для продукту важливо проведення як функціонального тестування, так і перевірка нефункціональних вимог. Оскільки тестування функціональності програмного забезпечення дає відповідь на питання „що” робить продукт? - перевіряє, чи справжній результат працює відповідно до очікуваного результату, перевірка операцій і дій Програми.Здійснюється вручну і тестує відповідно до вимог замовника.

А нефункціональне тестування проводиться на основі очікувань замовника та вимог до продуктивності та перевіряє поведінку програми, перевіряє час відгуку та швидкість програмного забезпечення за певних умов.Відгуки клієнтів є більш цінними для нефункціонального тестування, оскільки допомагають вдосконалити та дозволяють тестувальнику знати очікування клієнта. Виконується в більшості за допомогою автоматизованих тестів.

3. Димне тестування - у тестуванні програмного забезпечення означає мінімальний набір тестів на явні помилки. Цей тест зазвичай виконується самим програмістом. Програму, що не пройшла такий тест, немає сенсу передавати на глибше тестування. Виконується з метою перевірити, що критично важливі функціональні частини продукту працюють як належить. Виконується перед регресійним тестуванням.

Третій рівень

1. +

| **Test Scenario** | Check the kitten photo upload | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Go to the application | | Application should open | | Pass | | |
| 2 | Enter Userid & Password | | Credential can be entered | | Pass | | |
| 3 | Click Submit | | Customer is logged in | | Pass | | |
| 4 | Try upload the cats photo | | Cats photo is uploaded to the app | | Pass | | |
| 5 | Click Save | | Uploaded photos of cats must be saved in the application | | Pass | | |

| **Test Scenario** | Check uploading photos of other animals/people/objects | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Go to the application | | Application should open | | Pass | | |
| 2 | Enter Userid & Password | | Credential can be entered | | Pass | | |
| 3 | Click Submit | | Customer is logged in | | Pass | | |
| 4 | Try upload photo a hamster | | The application should classify the photo of a hamster as other animal and not allow downloading | | Pass | | |
| 5 | Click Save | | The save button should not be active as the photo of the cat has not been uploaded | | Pass | | |

| **Test Scenario** | Check adding friends | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Go to the application | | Application should open | | Pass | | |
| 2 | Enter Userid & Password | | Credential can be entered | | Pass | | |
| 3 | Click Submit | | Customer is logged in | | Pass | | |
| 4 | Search friends in contacts | | When clicked on the search button uploading list of contacts friends | | Pass | | |
| 5 | Click Add friend | | Selected from the list friends are added | | Pass | | |

| **Test Scenario** | Check the likes | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Go to the application | | Application should open | | Pass | | |
| 2 | Enter Userid & Password | | Credential can be entered | | Pass | | |
| 3 | Click Submit | | Customer is logged in | | Pass | | |
| 4 | Choose a cat photo | | Cat photo is uploaded on the screen | | Pass | | |
| 5 | Like the chosen cat photo | | The cat photo shows a like mark | | Pass | | |

| **Test Scenario** | Check the comments | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Pass / Fail / Not executed / Suspended** | | |
|
| 1 | Go to the application | | Application should open | | Pass | | |
| 2 | Enter Userid & Password | | Credential can be entered | | Pass | | |
| 3 | Click Submit | | Customer is logged in | | Pass | | |
| 4 | Choose a cat photo | | Cat photo is uploaded on the screen | | Pass | | |
| 5 | Leave a comment under the chosen cat photo | | The comment shows on comment box | | Pass | | |

1. - Додаток повинен бути захищений від несанкціонованого доступу;

- Додаток повинен бути здатний обслуговувати необхідну кількість користувачів без будь-якого зниження продуктивності;

- Додаток повинен бути доступний для користувача у будь-який момент часу;

- Додаток повинен бути простим, зручним та зрозумілим у використанні;

- Додаток повинен бути сумісний з іншими системами;

- Додаток має відповідати всім чинним законам і нормам.

1. Додаток надає доступ тільки зареєстрованим користувачам з верифікованим профілем (тестування безпеки).
2. Час завантаження додатку не може перевищувати 5 секунд, до 1000 користувачів мають до нього доступ одночасно(тестування продуктивності).
3. Додаток має бути встановленим на всіх версіях Android починаючи з 4.4 та iOS 7.1.2 (тестування сумісності).