

Ірина Селіна

Домашнє завдання. Виконано 3 рівня: Beet Seed, Beet Sprout, Mighty Beet

Beet Seed

1.Склади порівняльну таблицю функціонального, нефункціонального і пов'язаного зі змінами видів тестування.

Порівняння має містити такі блоки:
що перевіряється;
коли застосовується;
обмеження;
особливості

	Функціональне	Нефункціональне	Пов'язане зі змінами регресійне
що перевіряється	функціонал інтерфейсу, взаємодію з базами даних	характеристики ПЗ які можна виміряти різними величинами	виправлений баг, нові функції
коли застосовується	на всіх етапах розробки	коли функціонал вже стабільний, на пізніх етапах	після змін
обмеження	не перевіряє якість	потребує спеціальних інструментів і знань	складно проводити тести вручну (займе багато часу)
особливості	зосереджена на зовнішній поведінці	перевіряє якість	забезпечує стабільність системи

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

Ретестинг тестує виправлений баг, або нову функцію. Регресія тестує, що ці зміни не зламали систему в цілому, або конкретну частину. Вони будуть перевіряти різні функції. Ретестинг дає розуміння що баг був виправлений, або нова функція працює згідно очікуванням. Регресія ставить за мету повну або часткову перевірку функціоналу після внесення змін. Ретестинг буде перше, що тестувальник буде тестувати і для цього він буде брати тест-кейси які видали - failed, а при регресійній перевірці ми беремо тест-кейси по яким тестували passed.

Beet Sprout

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

Якщо так – в яких випадках?

Якщо ні – чому?

Обґрунтуй свою відповідь.

Так, можна провести тільки функціональне тестування. Якщо нам не важлива, або не критична якість продукту:

- внутрішні програми компанії (сайти з розкладом, системи пропуску)
- продукт який треба швидко випустити на ринок і подивитися реакцію користувача (для починаючого бізнесу)
- системи які рідко використовуються (архів старих даних, звіти)

Приклад: внутрішній сайт з розкладом заходів в інституті, може завантажуватися довго, або некоректно виглядати на різних пристроях - не критично. Тому що альтернативи немає, або йти в деканат до секретаря і шукати в документації розклад.

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

Smoke (димове) тестування проводиться після змін і перевіряє чи найважливіші функції ПЗ працюють як очікується. Це саме перше тестування яке ми виконуємо після змін. Чи завжди воно є доречним? - Так, якщо починати з ретельного тестування коли важливі функції не працюють то буде витрачено багато часу дарма, а іноді неможливість тестування коли програма навіть не відкрилася. Виявиться що важливий функціонал не працює і після його виправлення нам все одно прийдеться перевіряти все, бо відбулася ще одна зміна.

Mighty Beet

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.

ID: TC_001		Priority: High
Summary: перевірити чи завантажується картинку в форматі JPG і PNG на iOS пристрої Preconditions: користувач зареєстрований		
№	Steps	Expected result
1	зайти в застосунок	
2	натиснути особистий кабінет	
3	натиснути кнопку завантажити фото	
4	вибрати фото в форматі JPG і PNG	
5	натиснути кнопку додати фото	Користувач успішно додав картинку
ID: TC_002		Priority: Medium
Summary: перевірити чи не завантажується картинка інших тварин/людей/об'єктів Preconditions: користувач зареєстрований		
№	Steps	Expected result
1	зайти в застосунок	
2	натиснути особистий кабінет	
3	натиснути кнопку завантажити фото	
4	вибрати картинку на якій зображена людина	
5	натиснути кнопку додати картинку	користувач не зміг додати картинку, система блокує і видає помилку
ID: TC_003		Priority: Low
Summary: перевірити чи можна додавати вподобайки Preconditions: користувач зареєстрований		
№	Steps:	Expected result
1	зайти в застосунок	
2	натиснути на іконку головна сторінка	
3	на картці з фото котиків натиснути іконку сердечко	користувач успішно поставив вподобайку, іконка сердечко змінило колір
ID: TC_004		Priority: Medium
Summary: перевірити чи можна додати коментар до картинки Preconditions: користувач зареєстрований		

№	Steps	Expected result
1	зайти в застосунок	
2	натиснути іконку головної сторінки	
3	вибрати будь яку картинку на головній сторінці	
4	натиснути кнопку додати коментар	
5	в полі коментар написати <текст>	
6	натиснути Додати	користувач успішно додав коментар під картинкою
ID: TC_005		Priority: Low
Summary: перевірити чи можна додати коментар з пустим полем Preconditions: користувач зареєстрований		
№	Steps	Expected result
1	зайти в застосунок	
2	натиснути іконку головної сторінки	
3	вибрати будь яку картинку на головній сторінці	
4	натиснути кнопку додати коментар	
5	натиснути кнопку Додати	користувач не зміг додати пусте поле коментарів, кнопка додати була не активна

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

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

- Вимога: застосунок повинен витримувати 10000 користувачів одночасно присутніх на головній сторінці.
 - навантажувальне тестування (за допомогою коду імітувати присутність користувачів)
- Вимога: застосунок повинен коректно відображатися на операційних системах iOS та Android.
 - тест на сумісність (встановити додаток на різних пристроях та серіях)
- Вимога: застосунок повинен бути інтуїтивно зручним для користувача.
 - тест юзабиліти (створити фокус групу користувачів які не бачили додаток)