**Урок 5. Ніна Тиченок. Домашня робота. Тестова документація. Test plan, Check list, Test suite, Test case, Bug report**

1. Склади порівняльну таблицю трьох видів тестової документації:

| **Назва тестової документації** | **Основні характеристики** | **Переваги** | **Недоліки** |
| --- | --- | --- | --- |
| **Чек-ліст** | В тестуванні чек-лісти використовуються для систематичної перевірки функціональності, визначення відповідності вимогам та виявлення дефектів у програмному забезпеченні. Основні характеристики чек-листа:  Назва тест-ліста  Опис тест-ліста  Перелік Тест-Кейсів  Пріоритети Тест-Кейсів  Статус Тест-Ліста  Сценарії Тестування  Умови Виконання  Залежності Тест-Кейсів  Результати Виконання | По чек-лісту команда бачить і відслідковує поточний стан роботи і якість продукту. Працюючи за чек-лістом виключається можливість повторної перевірки, і підвищується якість тестування. Ймовірність залишити без уваги якийсь функціонал знижується. Співробітники-тестувальники можуть взаємозамінятись. Економія робочого часу. Написавши чекліст одного разу, його можна використовувати повторно, з огляду на актуальність інформації. | Потрібно регулярно оновлювати чек-листи, так як чек лісти можуть не охоплювати зміни у програмі або оновлення функціоналу ПЗ.  Чек-листи можуть бути недостатньо гнучкими для вирішення унікальних випадків. Чек лісти можуть не виявляти всі можливі дефекти, тому важливо постійно доповнювати чек-лісти іншими методами тестування.  Можливість пропуску дефектів, так як чек лісти можуть не виявляти всі можливі дефекти. |
| **Тест-кейс** | Тест-кейс у тестуванні - це документ, який описує конкретний тестовий сценарій або тестовий випадок для перевірки певної функціональності програмного забезпечення.  Характеристики:  Назва тест-кейсу  Опис тест-кейсу  Кроки виконання  Вхідні дані  Очікувані результати  Стан системи  Умови виконання  Тестовані області  Пріоритет тест-кейсу  Стан тест-кейсу | Структурований підхід  Явна специфікація, тобто кожен тест-кейс чітко визначає, що і як слід перевірити.  Точність та Повторюваність - Тест-кейси дозволяють виконувати тестові випадки з точністю та повторюваністю, що важливо для надійності результатів тестування.  Раннє Виявлення Дефектів, - використання тест-кейсів може сприяти ранньому виявленню дефектів, оскільки кожен аспект функціональності системи чітко перевіряється.  Валідація вимог, - Тест-кейси допомагають перевіряти, чи відповідає програмне забезпечення вимогам та специфікаціям.  Тест-кейси можуть легко інтегруватися з автоматизованими тестовими інструментами, що полегшує автоматизацію тестового процесу.  Тест-кейси надають можливість іншим членам команди перевірити та виконати тестування згідно із визначеними критеріями.  Стан виконання кожного тест-кейсу може служити індикатором прогресу тестового процесу.  Покращення Документації- Тест-кейси додаються до документації проекту і служать як важливий ресурс для розуміння та валідації функціональності | Статичність. Так як і тест-лісти, тест кейси потрібно постійно оновлювати, враховуючи зміни у вимогах чи функціональності.  Збільшення обсягу програмного забезпечення може вести до великої кількості тест-кейсів, що може стати важким у керуванні та підтримці.  Створення та підтримка великої кількості тест-кейсів може вимагати значного часу та зусиль.  Тест-кейси можуть не завжди виявляти неочікувані проблеми чи дефекти, які виникають під час роботи всієї системи.  Створення ефективних тест-кейсів вимагає досвіду та спеціалізованих знань в області тестування. |
| **Користувацький сценарій (Use Case**) | Користувацький сценарій (Use Case) в системному аналізі та проектуванні є специфікацією функціональності системи з точки зору користувача. Основні характеристики користувацького сценарію включають:  -Назва Користувацького Сценарію  -Учасники  - Опис Сценарію  -Попередження та Обмеження  -Приклади данних  -Альтернативні Потоки  -Дії, які слід виконати після успішого, або неуспішного завершення сценарію  -Вказівка на важливість, або пріоритетність сценарію в контексті функціональності системи | Користувацькі сценарії сконцентровані на взаємодії користувача з системою, що дозволяє тестувальникам легше розуміти та відтворювати реальні умови використання продукту.  Use Cases можуть служити як засіб ефективної комунікації між розробниками, тестувальниками та замовниками щодо функціональних вимог до системи.  Користувацькі сценарії часто визначають конкретні сценарії взаємодії, що полегшує розробку тест-кейсів та планування тестування.  Оскільки користувацькі сценарії відображають реальні умови взаємодії з системою, тестові випробування стають більш реалістичними та покликаними виявити проблеми, які можуть виникнути в реальному середовищі. | Користувацькі сценарії можуть бути обмежені у виявленні складних або взаємодіючих проблем, оскільки вони фокусуються на конкретних ситуаціях взаємодії.  Сценарії можуть бути обмежені у виявленні складних або взаємодіючих проблем, оскільки вони фокусуються на конкретних ситуаціях взаємодії.  Користувацькі сценарії можуть не охоплювати якісь непередбачувані події, які можуть виникнути в реальному середовищі. |

2. Склади чек-ліст для перевірки головної сторінки свого улюбленого інтернет-магазину (rozetka.ua, hotline.ua, silpo.ua тощо).

**Відповідь:**

мій чек-ліст за посиланням:

[чек-ліст. Урок 5. Ніна Тиченок. Домашня робота. Темплейт для Чек-листа](https://docs.google.com/spreadsheets/d/1FGUudQ__P2R0_tvltzWagUNSRsINko-nfJW7jAqVBmU/edit?usp=sharing)

3. Склади 5 позитивних і 5 негативних тест-кейсів для перевірки основної функціональності твого улюбленого інтернет-магазину (rozetka.ua, hotline.ua, silpo.ua тощо)

**Відповідь:**

мій тест-кейс за посиланням:

[Негативний і позитивний тест кейс. Урок 5. Ніна Тиченок. Темплейт для Чек-листа](https://docs.google.com/spreadsheets/d/1pokHRm7MJFeGSiYxPyg-Opu4YpU2MVxByKGfW9NKH2I/edit?usp=sharing)

4. Ти – засновник/ця стартапу, який планує випустити на ринок мобільний застосунок для обміну світлинами котиків для iOS та Android пристроїв. Стартап має команду з 3 розробників (back-end, iOS та Android). Вимоги та сценарії використання надаєш ти як головний стейкхолдер.

Завдання: Створи тест-план, який дозволить випустити продукт належної якості.

**Відповідь:**

### **Тест План: КотикОбмін\_App**

#### 1. Вступ

1.1 Мета:

* Перевірка функціональності та якості мобільного застосунку "КотикОбмін".

1.2 Об'єкт тестування:

* Мобільний застосунок "КотикОбмін" для iOS та Android.

1.3 Команда:

* Головний стейкхолдер (власник стартапу)
* Back-end розробник
* Розробник для iOS
* Розробник для Android
* Тестувальник

#### 2. Функціональність

2.1 Реєстрація та Вхід в Систему:

2.1.1 iOS:

* Реєстрація нового користувача.
* Вхід в систему існуючого користувача.

2.1.2 Android:

* Реєстрація нового користувача.
* Вхід в систему існуючого користувача.

2.2 Обмін Зображеннями:

2.2.1 Завантаження Зображення:

* Завантаження зображення котика на платформу.

2.2.2 Перегляд Зображень:

* Перегляд зображень котиків інших користувачів.

2.2.3 Оцінка та Коментування:

* Можливість оцінювати та залишати коментарі під зображеннями.

2.3 Сповіщення:

* Отримання сповіщень про нові коментарі, оцінки або події на платформі.

#### 3. ~~Негативні Сценарії~~

~~3.1 Неправильний Ввід Даних:~~

~~3.1.1 iOS:~~

* ~~Спроба зареєструватися з неправильним форматом email.~~

~~3.1.2 Android:~~

* ~~Спроба зареєструватися з неправильним форматом email.~~

3.2 Обмеження Пам'яті:

3.2.1 iOS:

* Завантаження зображення, яке перевищує максимально допустимий розмір.

3.2.2 Android:

* Завантаження зображення, яке перевищує максимально допустимий розмір.

#### 4. Продуктивність

4.1 Відповідь на Запити:

4.1.1 Back-End:

* Перевірка швидкості відповіді сервера на запити.

4.1.2 iOS та Android:

* Тестування швидкості завантаження та відображення зображень.

#### 5. Сумісність

5.1 iOS та Android:

* Тестування на різних версіях операційних систем.

#### 6. Безпека

6.1 Захист Даних:

* Перевірка наявності захисту особистих даних користувачів.

#### 7. Відмовостійкість

7.1 Стійкість до Навантаження:

* Тестування застосунку під великим навантаженням.

#### 8. Інтеграція

8.1 Підключення до Соціальних Мереж:

* Перевірка інтеграції з соціальними мережами для можливості обміну зображеннями.

#### 9. Тестування Апаратних Ресурсів

9.1 iOS та Android:

* Перевірка використання апаратних ресурсів під час роботи застосунку.

#### 10. Документація

10.1 Технічна Документація:

- Перевірка наявності та актуальності технічної документації.

### **Заключення:**

* Забезпечити 100% покриття коду тестами.
* Використовувати автоматизовані тести для регулярного виконання.
* Тестування повинно проводитися в реальних умовах використання (реальні мережі, пристрої різного характеру).