| :росток:Beet Seed — відпрацюй навички на базовому рівні.  1. Склади порівняльну таблицю трьох видів тестової документації:   | Назва тестової документації | № | Основні характеристики | Переваги | Недоліки | | --- | --- | --- | --- | --- | | Чек-ліст | 1 |  |  |  | | 2 |  |  |  | | 3 |  |  |  | | Тест-кейс | .. |  |  |  | | Користувацький сценарій (Use Case) | .. |  |  |  |   2. Склади чек-ліст для перевірки головної сторінки свого улюбленого інтернет-магазину (rozetka.ua, hotline.ua, silpo.ua тощо). Темплейт для чек-ліста можна взяти в кроці три теорії.  *Відповіді до завдань запиши у файл Google Docs. Додай посилання на нього в LMS.* |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| :травы:Beet Sprout — детальніше заглибся в практику.  1. Виконай завдання попереднього рівня.  2. Склади 5 позитивних і 5 негативних тест-кейсів для перевірки основної функціональності твого улюбленого інтернет-магазину (rozetka.ua, hotline.ua, silpo.ua тощо)  *Відповіді запиши в той самий файл Google Docs.* |

| Name of testing document | Main features | Advantages | Disadvantages |
| --- | --- | --- | --- |
| Checklist | A list of checks which are needed to be performed during the system testing process. Check list contains the names of features that need testing, without explanations how to do it. It is assumed that the testers already know how to check these. | 1. Check list helps you to be structured and consistent in your testing, it helps the tester to be well-organized in their testing process. 2. By systematically verifying each item on the checklist, testers can reduce the chance of missing an issue in test scenarios or requirements, leading to fewer errors and defects in the software. | 1. Each new version of the build that will go in release has new features so they need to be added in the list. It means checklists can sometimes be really long so the testing process might be long lasting and exhausting. 2. Checklists can be rigid and may not easily accommodate variations or exceptions in testing scenarios. So when testers encounter unexpected situations it can confuse them and even freeze the testing process for some time. 3. Checklists really depend on their fulfillness, if some of the scenarios are missing or a checklist is kinda old compared to the software’s new version it will lead to insufficient testing process. |
| Test case | A document which contains a set of conditions used to determine whether a system, software application, or product functions correctly according to its specifications. Test cases are designed to cover various scenarios and a tester should go through all the steps a test case has to meet the expected results of it. | 1. Test cases have exact steps, which is really helpful in defining bugs; if a bug shows itself after a specific step you will know which area needs fixing for sure. 2. After being created once, test cases can be reused ( for example for regression testing), ensuring that previously identified issues won’t happen after changes that are made to the software. 3. The documentation of test cases makes it much easier for testers to understand the test objectives and the whole process, to meet the expected results and why these exact steps have been written. It especially helps for newbies in this sphere. | 1. As it was mentioned in advantages, steps in a test case should be very accurate and specific. If any information is missing (for example in the Preconditions section there is no information about operating system) or one of the steps is not detailed enough it will mean only incorrect testing of this case. 2. Even if there is only one change in the new software build (for example, a section name was changed), you need to rewrite this change in many other test cases. All test cases need to be periodically checked for relevance and constantly refined to ensure that they perform their role well. |
| Use Case | A documentation management technique, which includes possible cases of user (in documents is mentioned as an Actor) and system interaction. There can be even more than two participants and the user can be either a person or another system. | 1. Use cases provide a clear understanding of how users interact with the system to achieve their goals, so all of the stakeholders and others involved in a project have a common understanding of user requirements. 2. Because of the first point, we can say that use cases facilitate communication between all kinds of departments. Designers, developers, end-users and testers - use cases ensure that everyone has a shared understanding of system behavior. 3. Use cases can be used to validate system requirements by ensuring that they accurately reflect user needs and expectations. Moreover, they provide a basis for verifying system behavior through testing. | 1. Developing use cases for complex systems can be time-consuming and challenging, especially for large-scale projects with multiple stakeholders or/and diverse user needs. 2. Use cases primarily focus on functional requirements and may not adequately address non-functional requirements such as performance, security, and scalability. |

2. [Viktoriia Khodakivska - Lesson №5. Registration checklist](https://docs.google.com/spreadsheets/d/1pqEPNEQ_kyqbCz1hSYCFzOuEEsFePRo-hSEwejquAgw/edit?usp=sharing)

Beet Sprout

[Viktoriia Khodakivska - Lesson №5. 5 positive and 5 negative test cases](https://docs.google.com/spreadsheets/d/1J48Fwgb-8mFJK5N8Rp9IHSBq7MfWOYUa7QTDSowlxnw/edit?usp=sharing)