

Завдання 1.			
1.1. Порівняльна таблиця видів тестової документації			
Назва тестової документації	Основні характеристики	Переваги	Недоліки
Чек-ліст	<ul style="list-style-type: none"> - Структурований формат: чек-листи, як правило, структуровані в простому форматі зі списком пунктів, які потрібно перевірити. - Послідовний порядок: Пункти в чек-листі зазвичай перераховані в послідовному порядку, щоб допомогти тестувальнику в процесі тестування. - Інструмент перевірки: чек-листи служать інструментом перевірки, щоб гарантувати, що всі необхідні тести проведені. - Стандартизоване тестування: чек-листи допомагають стандартизувати процес тестування, надаючи чіткий набір завдань або критеріїв для перевірки. 	<ul style="list-style-type: none"> - Покращує узгодженість: чек-листи допомагають забезпечити послідовне виконання тестів у повторюваній манері. - Зменшує кількість помилок: Використання чек-листів зменшує ймовірність пропуску певних тестових кейсів або кроків. - Ефективність: чек-листи допомагають тестувальникам ефективно виконувати тестові завдання, забезпечуючи структурований підхід. - Навчальний інструмент: чек-листи можна використовувати як навчальний інструмент для нових членів команди, щоб зрозуміти процес тестування. - Документація: чек-листи слугують цінною формою документації для відстеження прогресу та результатів тестування. 	<ul style="list-style-type: none"> - Обмежена гнучкість: Чек-листам може не вистачати гнучкості в обробці складних тестових сценаріїв, які вимагають адаптивних підходів до тестування. - Залежність від точності: якщо контрольний список не оновлюється або не є точним, це може призвести до неповного або неефективного тестування. - Суб'єктивність: Деякі тестові завдання можуть включати суб'єктивну оцінку, яку нелегко включити в контрольний список. - Обслуговування: Для забезпечення релевантності та точності чек-листів необхідна регулярна підтримка їхньої актуальності.

<p>Тест-кейс</p>	<ul style="list-style-type: none"> - Конкретні кроки: Тестові кейси складаються з конкретних кроків, які потрібно виконати для тестування певної функціональності або можливості. - Очікувана поведінка: Вони визначають очікуваний результат або поведінку при виконанні тесту. - Вхідні дані: Тестові кейси включають вхідні дані, необхідні для проведення тесту. - Передумови: Тестові кейси визначають будь-які передумови або умови, які необхідно виконати перед запуском тесту. - Очікуваний результат: Вказують очікуваний результат або результат після виконання кроків тесту. - Тестові дані: Тестові кейси містять дані, необхідні для запуску тесту. 	<ul style="list-style-type: none"> - Повторюваність: Тест-кейси забезпечують систематичний спосіб повторення тестів для отримання послідовних результатів. - Документація: Вони слугують документацією для діяльності з тестування, що полегшує новим членам команди розуміння процесу тестування. - Відстежуваність: Тест-кейси можуть бути пов'язані з вимогами, що допомагає у відстежуваності та гарантує, що всі вимоги будуть перевірені. - Ефективність: Тест-кейси допомагають виявити дефекти на ранніх стадіях, роблячи процес тестування більш ефективним. - Автоматизація: Тестові кейси можна автоматизувати, заощаджуючи час і зусилля на повторюваних завданнях тестування. 	<ul style="list-style-type: none"> - Багато часу: Написання детальних тестових кейсів може зайняти багато часу, особливо для складних систем. - Обслуговування: Тестові кейси потрібно регулярно оновлювати, щоб відображати зміни в системі, що може бути накладними витратами на обслуговування. - Накладні витрати: Управління великою кількістю тестових кейсів може призвести до накладних витрат на виконання та обслуговування. - Не вичерпність: Тестові кейси можуть не охоплювати всі можливі сценарії, що призводить до прогалин у тестовому покритті.
-------------------------	--	---	---

Користувачь кий сценарій (Use Case)	- Орієнтованість на користувача: Use case фокусуються на взаємодії між користувачами і програмною системою, деталізуючи конкретні сценарії того, як користувачі взаємодіють з системою для виконання завдання. - Розповідний формат: Use case зазвичай пишуться в розповідному форматі, описуючи кроки, які користувач робить для досягнення певної мети або виконання завдання. - Ідентифікує виконавців дій: Use case визначають виконавців дій, які беруть участь у взаємодії з програмною системою, таких як кінцеві користувачі, адміністратори або зовнішні системи. - Включає передумови та післядії: Use case визначають умови, які повинні бути виконані до і після виконання варіанту використання. - Можна візуалізувати: Use case часто представляють схематично, використовуючи такі методи, як діаграми варіантів використання, щоб забезпечити візуальне представлення функціональності системи.	- Ясність: Use case надають чіткий і детальний опис того, як користувачі взаємодіють з системою, допомагаючи забезпечити загальне розуміння функціональності системи. - Дизайн, орієнтований на користувача: Зосереджуючись на взаємодії користувачів, Use Case допомагають гарантувати, що програмна система відповідає потребам і очікуванням її користувачів. - Управління сферою застосування: Use case допомагають визначити сферу застосування системи, визначаючи різні взаємодії користувачів з нею, допомагаючи у плануванні проекту та визначенні пріоритетів. - Комунікація: Use case слугують інструментом комунікації між зацікавленими сторонами, дизайнерами, розробниками та тестувальниками, полегшуючи обговорення та забезпечуючи узгодження вимог. - Керівництво по тестуванню: Use case можуть безпосередньо використовуватися як основа для створення тестових кейсів, забезпечуючи всебічне покриття функціональності системи тестами.	- Складність: Створення детальных Use case вимагає значного часу і зусиль, особливо для великих і складних систем. - Суб'єктивність: Use case ґрунтуються на поглядах та розповідях користувачів, що може призвести до суб'єктивності та проблем з інтерпретацією. - Обслуговуваність: Підтримувати Use case в актуальному стані в міру розвитку системи може бути складно, що призводить до потенційних розбіжностей між задокументованими варіантами використання і фактичною поведінкою системи. - Потенційне перевантаження: У сценаріях, де є багато Use case, управління ними та визначення пріоритетів може бути непосильним для команди.
--	---	---	---

