

## Лабораторна робота № 2 Системне тестування. Розробка Check List і Test Case (4 години)

### Теоретична частина

Системне тестування якісно відрізняється від інтеграційного і модульного рівнів. Системне тестування розглядає систему в цілому і застосовується на рівні користувача інтерфейсів, на відміну від останніх фаз інтеграційного тестування, яке оперує на рівні інтерфейсів модулів, хоча набір модулів може бути аналогічним. Різні й цілі цих рівнів тестування. На рівні системи часто складно і малоефективно аналізувати проходження тестових траєкторій усередині програми, а також відстежувати правильність роботи конкретних функцій. Основним завданням системного тестування є виявлення дефектів, пов'язаних з роботою системи в цілому, таких як неправильне використання ресурсів системи, непередбачені комбінації даних користувача рівня, несумісність з оточенням, непередбачені сценарії використання, відсутня або неправильна функціональність, незручність у використанні тощо.

Оскільки системне тестування проводиться на рівні користувача інтерфейсів, то побудова спеціальної тестової системи стає технічно обов'язковим. Однак обсяги даних на цьому рівні такі, що зазвичай більш ефективним підходом є повна або часткова автоматизація тестування, що може зажадати створення тестової системи набагато складнішою, ніж система тестування на рівні модулів або їх комбінацій.

Необхідно підкреслити, що існує два принципово різних підходи до системного тестування.

У першому варіанті для побудови тестів використовуються вимоги до системи, наприклад, для кожного вимоги будується тест, який перевіряє виконання даної вимоги в системі. Цей підхід особливо широко застосовується при розробці військових і наукових систем, коли замовник цілком усвідомлює, яка функціональність йому потрібна, і становить повний набір формальних вимог. Тестувальник в даному випадку тільки перевіряє, чи відповідає розроблена система цього набору. Такий підхід передбачає довгу і дорогу фазу збору вимог, виконуваних до початку власне проекту. У цьому випадку для визначення вимог зазвичай розробляється прототип майбутньої системи.

У другому підході основою для побудови тестів служить уявлення про способи використання продукту і про завдання, які він вирішує. На основі більш-менш формальної моделі користувача створюються випадки використання системи, за якими потім будуються власне тестові випадки. Випадок використання (use case) описує, як суб'єкт використовує систему, щоб виконати те чи інше завдання. Суб'єкти або актори (actors) можуть виконувати різні ролі при роботі з системою. Випадки використання можуть описуватися з різним ступенем абстракції. Випадки використання не обов'язково охоплюють кожну вимогу. Можна конкретизувати випадки використання і розширювати

їх в набори більш специфічних випадків використання (покроковий опис випадку використання). В контексті конкретного випадку використання можна визначити один або більше число сценаріїв (test suite). Сценарій представляє конкретний екземпляр випадку використання - шлях в покроковому описі випадку використання. Кожен шлях (сценарій) в разі використання повинен бути протестований.

Вхідні дані для кожного сценарію треба вибирати таким чином:

- Ідентифікувати всі значення (вхідні дані), що може ставити суб'єкти для випадку використання.
- Визначити класи еквівалентності для кожного типу вхідних дан-них.
- Побудувати таблицю зі списком значень з різних класів еквівалентні-валентності.
- Побудувати тестові випадки на базі таблиці з урахуванням зовнішніх обмежень.
- Далі при побудові тестових випадків застосовувалися обидва підходи і при виконанні завдань необхідно діяти наступним обра-зом:
- На основі вимог визначити випадки використання (use case)
- На основі кожного випадку використання (use case) побудувати сценарії (test suite).
- Для кожного сценарію розробити тестові випадки (набір тестів) (test case).

Тест кейси (Test Cases). Для того щоб виявити ту чи іншу помилку (bug) необхідно провести певні дії, порівняти отримані результати з очікуваними і зробити висновок про те, чи є помилка (bug) чи ні.

Тест кейс - це мінімальний елемент тестування. Хороший тест кейс складається з трьох частин:

- Привести тестований продукт в потрібний стан
- Верифікувати то, що підлягає перевірці
- Привести продукт в початковий стан.

## Розроблення Check List і Test Case з використанням TestLog

Необхідні ресурси для тестування:

- Індивідуальне завдання
- Звіт лабораторній роботі.
- Програма TestLog.

## Опис тестованої системи

ТесPublisher - видавнича фірма, яка випускає книги з різної тематики. Фірма має ряд філій, розташованих в різних містах. Кожна філія функціонує як незалежна видавець-ство. Філії зберігають інформацію про авторів свого регіону, видавців, тематиці в базі даних. Базы даних філій синхронізовані з головним базою даної і працюють під управлінням SQL Server.

Завдання проекту полягає в тестуванні програмного модуля, котрий дозволяє співробітникам кожної філії оновлювати записи про авторів і виданнях у міру необхідності. Модифікації повинні отобра-тулитися на сервері баз даних. За запитом співробітників необхідно генерувати відповідні звіти.

### Сформовано наступний список вимог до проекту:

- Дані, які відображаються і модифікуються необхідно отримувати з бази даних, яка зберігається на сервері типу SQL.
- Зі звичайними правами (user) має можливість вибирати учасника зі списку. Кожен автор в списку повинен мати унікальний ідентифікатор.
- Переглядати і змінювати дані, що відносяться до автора має право тільки привілейований користувач (power user).
- Звіт створюється у вигляді Windows додатки.
- Звіт відображає імена авторів і назви книг в табличному вигляді.
- Звіт відправляється в головний офіс по електронній пошті.

### Робоче оточення системи і моделі її використання

Архітектура системи, яка реалізує всі етапи обробки даних: збір даних, узагальнення даних, архівування даних і створення звітів.

Кожен рівень включає в себе безліч інших компонентів. Ці компоненти побудовані на інформації, що міститься в описі системи.

Розглянемо модель варіантів використання системи. У цій моделі система взаємодіє з зовнішніми об'єктами (actors) під час завантаження і завершення роботи, при складанні та відсилання звітів на основі спів-лайливих даних, а також при збиранні та збереженні даних. Кожен з наявних варіантів використання можна описати за допомогою просто-го природної мови. Такий опис допомагає ідентифікувати об'єкти в системі і зрозуміти, що система повинна робити. У таблиці 1 представлений варіант використання Реєстрація.

Таблиця 1. Опис варіанту використання Реєстрація

Система	TecPublisher
Варіант використання	Реєстрація
Учасники	Система TecPublisher, користувач
Дані	Користувач вводить в вікно реєстрації (рис.4), ім'я та пароль. Ці дані передаються системі, вона порівнює їх з зберігаються в БД ім'ям користувача і паролем.
Вхідні дані	Ім'я користувача - текст довжиною 10 символів, може містити тільки числа і / або букви. Пароль- текст довжиною не менше 6 символів, але не більше 12. Може містити будь-які символи клавіатури.
Відповідь	У разі збігу імені користувача і пароля з зберігаються в БД відкриється вікно системи з головним меню, в іншому випадку повідомлення про відмову в реєстрації

Коментарі	На введення пароля дається три спроби, при перевищенні числа спроб видається повідомлення і користувач блокується. Користувачі розрізняються за рівнем доступу - привілейований і звичайний. Залежно від типу користувача деякі пункти меню системи можуть бути недоступні (рис.5).
-----------	---

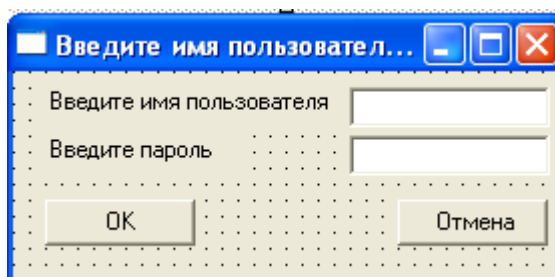


Рис.1. Вікно реєстрації в системі

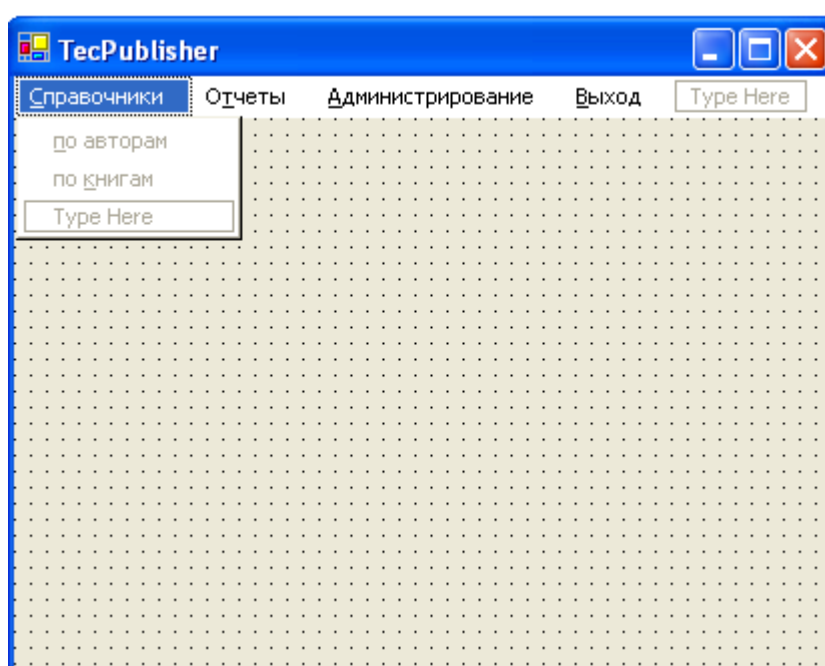


Рис.2. Головна форма системи

## Розроблення проекту тестування в рамках програми TestLog

### 2.1. Основні етапи створення тестового проекту (Test Project)

- Створення проекту (Projects) для тестування системи.
- Створення наборів тестів (test suite) для:
  - функціонального тестування (base testing);
  - GUI-тестування;
  - конфігураційний тестування (configuration testing);
  - інсталяційного тестування (installation testing);
- Опис оточення для тестування системи (Test Configuration)
- Опис необхідних ресурсів (Resources)
- Визначення необхідного штату (Testers)

### 2.2. Приклад розробки проекту тестування в програмі TestLog. Створення нового проекту.

Для створення нового проекту необхідно запустити програму TestLog і виконати команду File-New Database ..., у вікні, (рис.3) необхідно вказати розташування бази даних для тестування проекту.

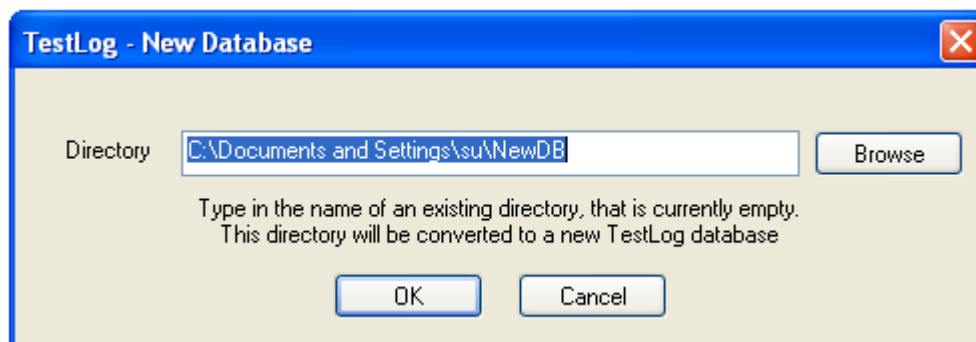


Рис.3. Створення нової бази даних тестування

Для початку роботи з системою необхідно визначити штат тестувальників (Test Team member). Розділ Testers визначає членів команди тестувальників, які будуть проводити тестування. Запис, з інформацією про тестівщика, створюється для кожного члена команди. Потім ці записи будуть використані при закріпленні тестувальника за конкретними тестами, які він буде виконувати.

Для створення нового запису необхідно:

1. Клацнути на кнопці Create tester
2. Клацнути правою кнопкою на папці Testers у вікні дерева проекту і вибрати команду New Tester (рис.7)
3. Вибрати команду Create-Tester

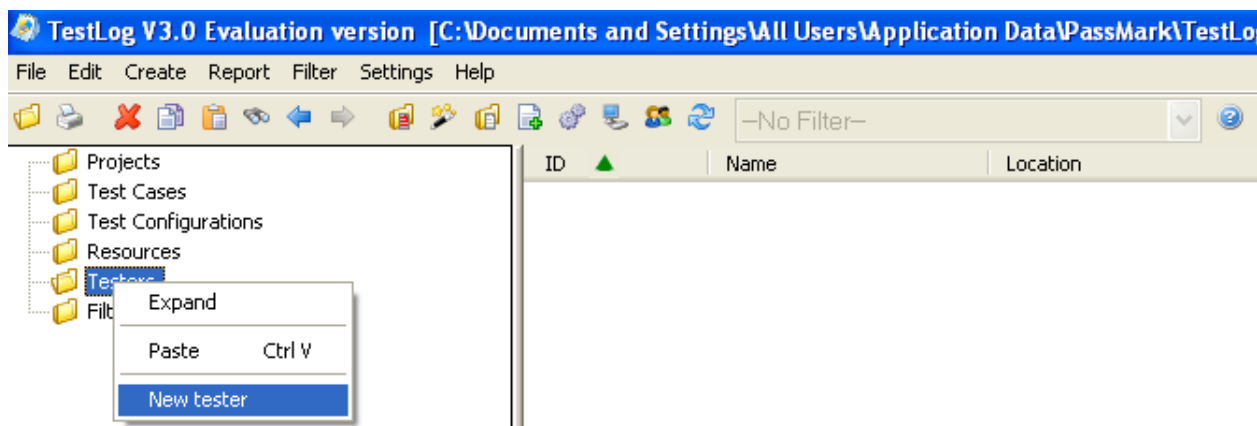


Рис.4. Додавання нового тестувальника

У вікні додавання нового тестувальника (рис.5) необхідно ввести інформацію про тестувальника: Tester ID (ідентифікатор тестувальника), Testers name (ПІБ тестувальника), Location (місцезнаходження). У разі віддаленого доступу необхідно вибрати режим доступу для тестувальника, а також пароль доступу.

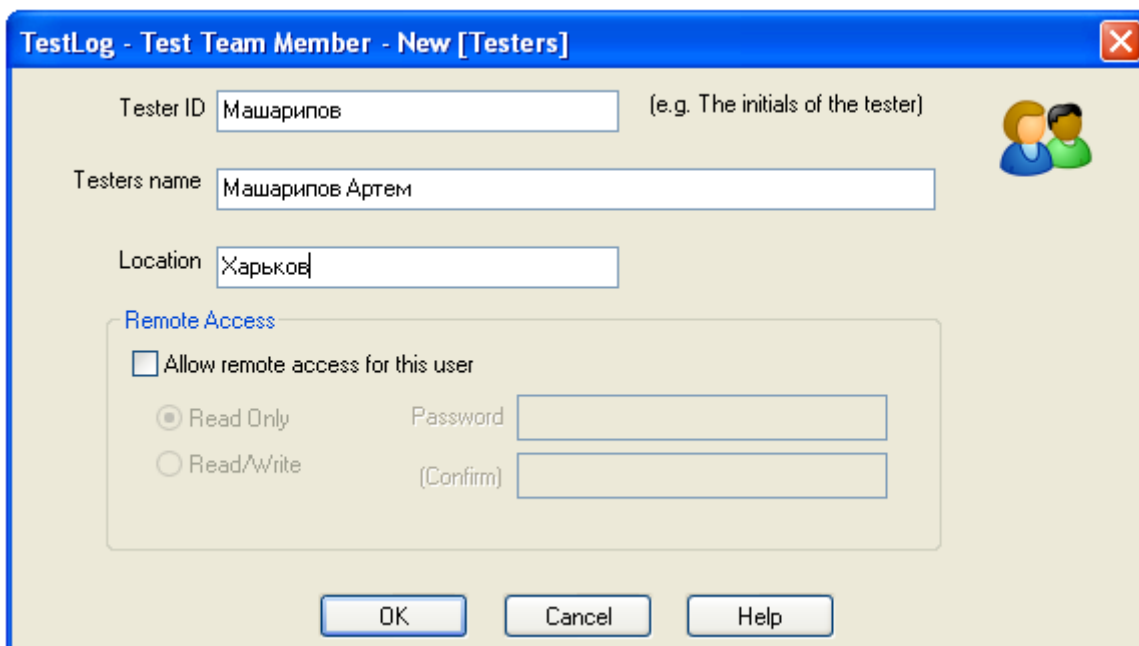


Рис.5. Вікно додавання нового тестувальника

База даних TestLog може містити кілька проектів. Рекомендується створювати окремі бази даних для різних проектів з тестування. Проект може містити безліч тестових наборів (test suites). Тестовий набір (test suite) є контейнером для тестових випадків (test cases). Тестові набори (test suite) дозволяють організувати тестові випадки в більш зручній короткій формі. Рекомендується створювати тестові набори для кожної логічної області планованих тестів.

Для створення нового проекту необхідно:

1. Клацнути на кнопці Create a new project в головному вікні програми.
2. Скористатися командою New project контекстного меню папки Projects (рис.6).
3. Виконати команду меню Create-Project

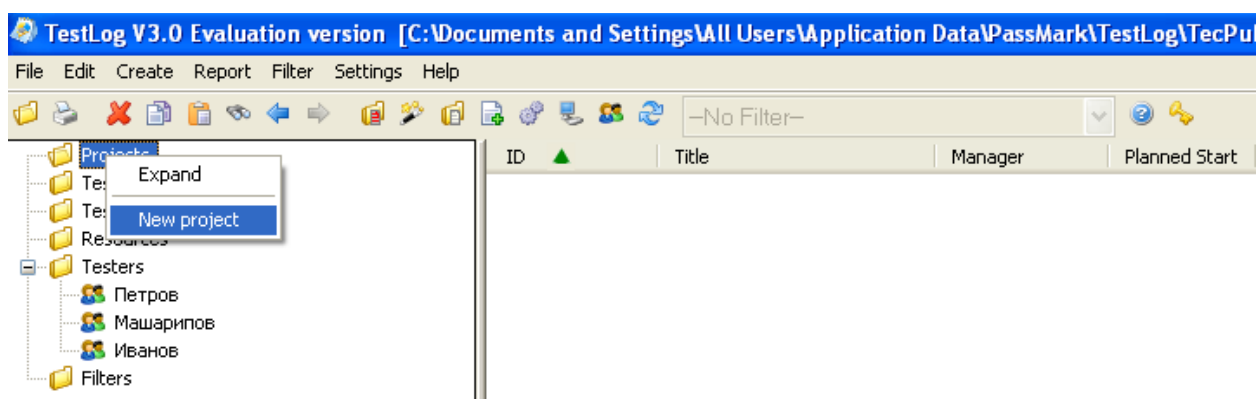


Рис.6. Додавання нового тестового проекту (test project)

Після виконання команди, з'являється вікно тестового плану (Test Plan) проекту в якому вказуються основні параметри нового проекту (рис.7).

TestLog - Project Test Plan - Edit [TEC1.tgp]

Project ID:  Manager:

Project Title:

Planned Start:  Planned Finish:

Actual Start:   Actual Finish:

Overview & Scope:

Resources & Staff Required:

Risks & Minimisation Actions:

Additional Project Notes:

References to Other Documents:

Buttons: OK, Cancel, Help

Рис.7. Тестовый план для нового проекта тестування

Тестове оточення Test Configuration визначає специфічне програмне забезпечення, яке використовується для більшості тестів. У загальному випадку Test Configuration визначає операційну систему і необхідний мінімум програмного забезпечення. Одні або кілька тестових оточень можуть використовуватися для різних тестових випадків.

Для створення Test Configuration необхідно:

1. Клацнути на кнопці Create test configuration в головному вікні програми.

2. Виконати команду New test Configuration контекстного меню папки Test Configurations (рис.8)

3. Виконати команду меню Create-Test configuration

У вікні визначення тестового оточення системи (Test Configuration) (рис.8) визначаються необхідні вимоги до ОС, на яких буде тестуватися система.

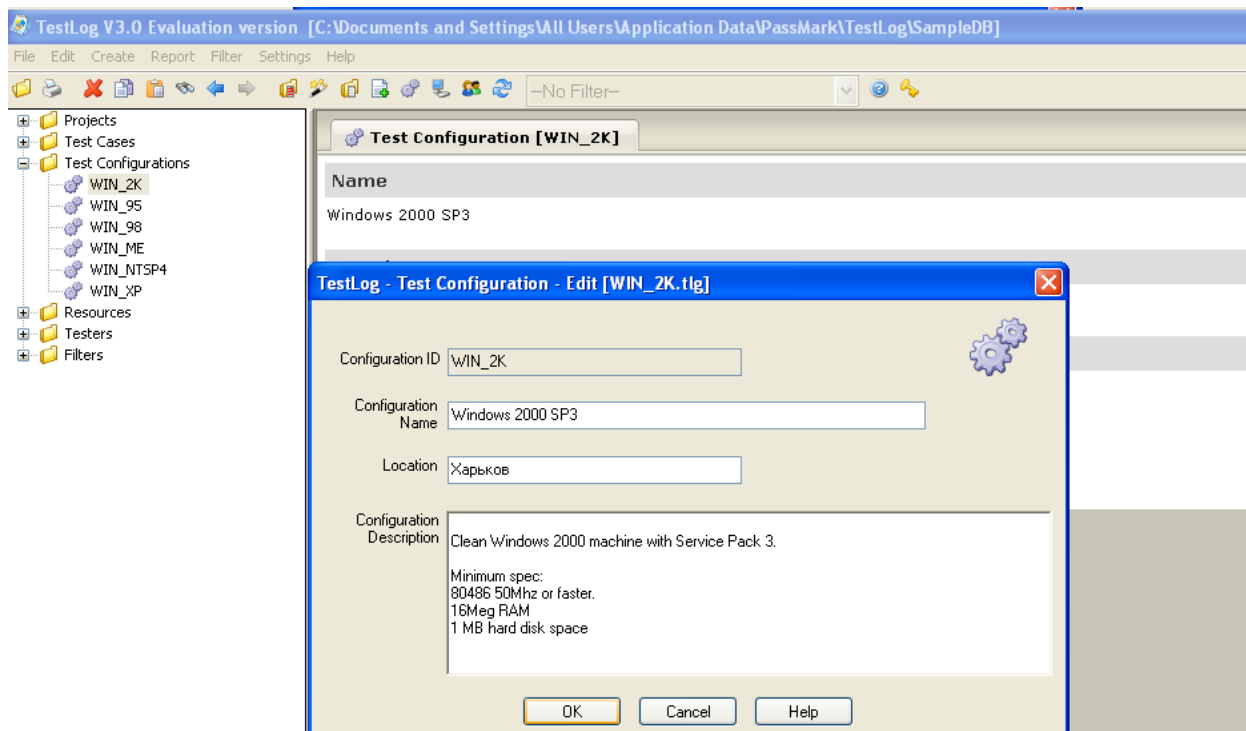


Рис.8. Вікно визначення тестового оточення для тестування системи

Ресурси Resource визначають специфічні вимоги до апаратних ресурсів для кожного тестового випадку.

Для створення ресурсу необхідно:

1. Клацнути на кнопці Create test resource в головному вікні програми.
2. Виконати команду New test resource контекстного меню папки Test Configurations (рис.9)
3. Виконати команду меню Create-Test resource

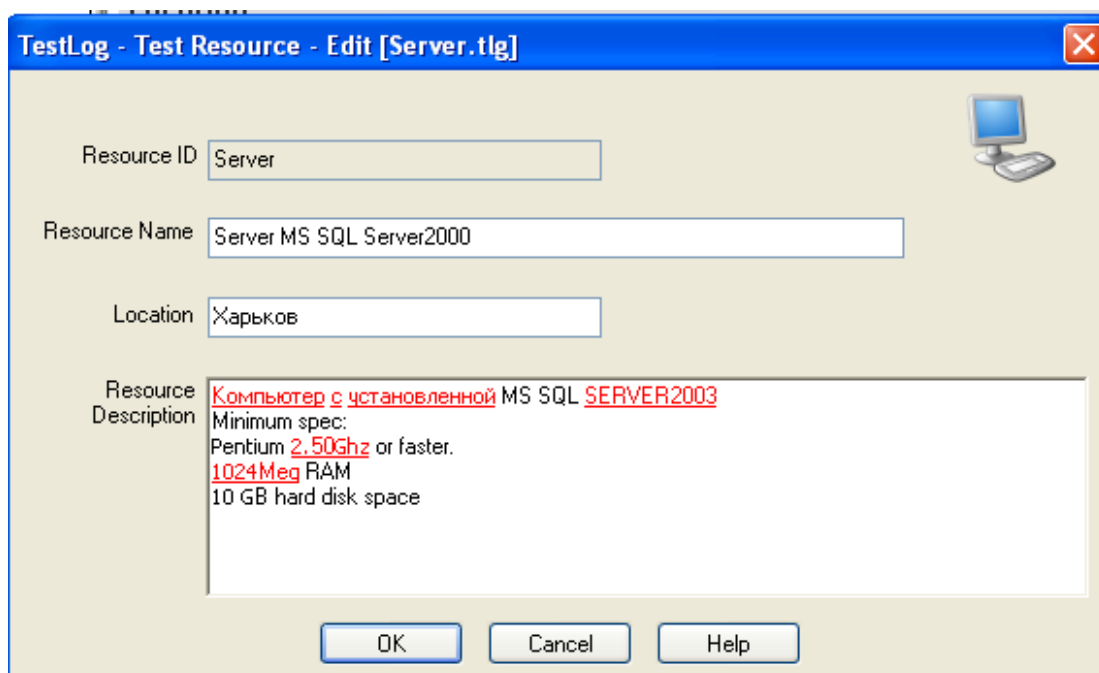


Рис.9. Вікно ресурсів



Для створення тестового набору Test Suite необхідно:

1. Клацнути на кнопці Create test suite в головному вікні програми.
2. Виконати команду New test suite контекстного меню папки Project Test Cases
3. Виконати команду меню Create-Test Suite

На рис.10 наведено вікно створення тестового набору для функціонального тестування системи TecPublisher

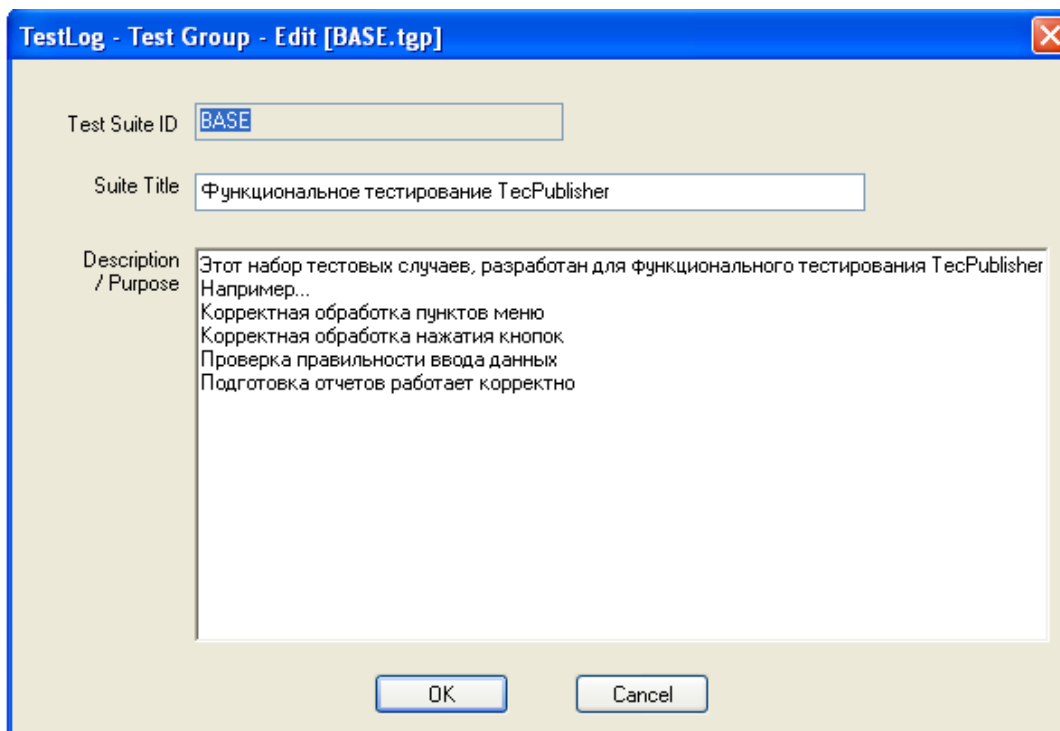


Рис.10. Вікно створення тестового набору Test Suite

Тестові випадки Test Cases найбільш важлива сутність в програмі TestLog. Тестові випадки містять велику кількість даних, необхідних для виконання тестів.

Для створення Test Cases необхідно:

1. Клацнути на кнопці Create test case в головному вікні програми.
2. Виконати команду New test case контекстного меню папки тісто-вого набору
3. Виконати команду меню Create-Test case

На рис.11 наведено вікно створення тестового випадку для інсталяційного тестування системи TecPublisher

The screenshot shows the 'TestLog - Test Case - Edit [INST\_001.tlg]' window. The interface includes the following fields and sections:

- Test-ID:** INST\_001
- Expected Duration:** 0 Hrs 30 Mins
- Title:** Основная установка
- Priority:** Medium
- Test Type:** Installability
- Created:** 00:00:00, 14-мар-2006
- Created By:** FERGUSXP
- Test Phase:** System test
- Updated:** 00:00:00, 11-июл-2006
- Update By:** ROCKLAKEXP
- Recommended Config. IDs:** WIN\_2K, WIN\_95, WIN\_98, WIN\_ME, << Add
- Requirements:** (empty field)
- Resource IDs Required:** (empty field) << Add
- Buttons:** Browse, Launch
- Prerequisites and Initial Condition:** "Чистая машина" без установленного TecPublisher. Настроена и сконфигурирована локальная сеть
- Test Description and Steps:**
  - Запустить install.exe. Проверить, что появляется окно 'Welcome to the TecPublisher Setup Wizard'
  - Нажать 'Next'. Проверить появление окна 'License Agreement'. В окне License Agreement... Проверить правильность имени продукта и его версии
- Expected Results:** Все описанные выше шаги должны выполняться без ошибок
- Notes Field 1:** (empty)
- Notes Field 2:** (empty)
- External Description Link\*:** (empty) Browse, Launch
- Footer:** \*a local document path or web page in the form http://
- Buttons:** OK, Cancel, Help

Рис.11. Вікно створення тестового випадку для виконання інсталяційного тестування системи

### Звіт з лабораторної роботи

Звіт з лабораторної роботи повинен містити роздруківки наступних сутностей, створених в програмі TestLog:

1. Сутність Project. Повинна містити: назву проекту, плануємо-мі дати початку і закінчення тестування, менеджера проекту, огляд проекту, ризики проекту.
2. Сутність Tester. Містить список прізвищ студентів, які беруть участь в тестуванні (по групі згідно з розподілом)
3. Сутність Test Configuration. Містить опис програмного забезпечення, необхідного для проведення тестування.
4. Сутність Resource. Містить опис необхідного для тестування апаратного забезпечення.
5. Сутність Test Suite. Містить тестові набори для:
  - a. Функціонального тестування (base testing);
  - b. GUI-тестування;
  - c. Конфігураційного тестування (configuration testing);
  - d. Інсталяційного тестування (installation testing);

6. Сутність Test Cases. Містить набір тестових випадків для кожного тестового набору, згідно індивідуального завдання.