

**Широкопетлєва М.С., Черепанова Ю.Ю.**

*Харківський національний університет радіоелектроніки*

## **ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ ТЕСТУВАННЯ ЗНАНЬ З МОВИ SQL**

На сучасному етапі комп'ютерний контроль знань студентів є основою для отримання об'єктивної оцінки рівня їх навчальних досягнень. Великий інтерес представляє собою проблема тестування знань з застосуванням тестів у відкритій формі, особливо в зв'язку зі зростанням ролі дистанційного і самостійного навчання. Система тестування знань з мови запитів до бази даних може використовуватись у навчальних закладах для спрощення завдань перевірки і навчання студентів. Метою дослідження є визначення підходів до проектування програмної системи тестування знань з мови SQL.

Найбільш ефективним способом тестування знань структурованої мови запитів представляється тестування з використанням тестів у відкритій формі, а саме спосіб, при якому студенту пропонується вирішити завдання - написати текст запиту до бази даних. Система повинна оцінити вірність запиту.

Потрібно мати на увазі, що різні запити виконують різну роботу з даними. Найбільш безпечні зчитують и повертають інформацію. Ці запити важливо перевіряти на помилки та перевіряти отримані дані з потрібними. Такі запити, як INSERT, UPDATE, DELETE, CREATE, DROP, впливають на дані, змінюючи їх, тому такі запити більш небезпечні при виникненні помилки. При роботі з запитами потрібно розділити обробку помилок різних запитів і передбачити різний порядок рядків у отримуваних даних при виконанні запиту.

Для програмної системи перевірки знань з мови SQL необхідно створити набір вимог за кожною тематичною одиницею та алгоритм порівняння результуючих вибірок (еталонної та запропонованої користувачем).

Для перевірки запитів на модифікацію слід передбачити виконання двох ізольованих транзакцій (для еталонного та запропонованого запитів) для унеможливлення довільної зміни початкових даних, а результати виконання завдання перевіряти за допомогою перевірки даних таблиць за таким критерієм: повний збіг даних таблиці при виконанні запиту на вибірку усіх даних з упорядкуванням по ключовому полю. Слід передбачити можливу зміну даних після кожного сеансу роботи студентів з системою. Таким чином потрібно відновлювати тестову базу даних після кожного сеансу роботи студента. Важливо також зберігати певний звіт про тестування студентів певної групи, і звіт про роботу окремих студентів цієї групи.

Для реалізації веб-сервісу тестування знання мови запитів до бази даних у вигляді веб-серверу та веб-клієнта було вирішено використовувати клієнт-серверну архітектуру. Веб-система розроблена з використанням платформи ASP.NET MVC. Для реалізації серверної частини був використаний принципи проектування SOLID. Серверна частина виконує функцію обробки введених клієнтом даних та надання клієнту інформації відповідно до його запитів.

На клієнтському боці для створення дизайну і динамічної роботи веб-сайту слід використовувати JQuery бібліотеку, що значно спрощує роботу з JavaScript. Клієнт являє собою web-додаток, створений також з використанням ASP.NET MVC та він виконує роль динамічної навігації між сторінками та взаємодії з клієнтом.

Для зберігання даних пропонується використання бази даних. В якості СУБД обрано Microsoft SQL Server, що краще всього інтегрується з ASP.NET проектами, при використанні технологій Entity Framework та LINQ та моделлю Code First.

Слід зауважити, що існують різні діалекти мови SQL та спеціальні функції, реалізовані у конкретних СУБД. При реалізації системи тестування слід передбачити можливість вибору типу СУБД для виконання запиту, але цей підхід вимагає необхідність підключення різних СУБД для коректної перевірки запитів, що в свою чергу підвищує вимоги до апаратного забезпечення серверної частини. На даний час в реалізованій системі тестова база використовує СУБД Microsoft SQL Server, та у майбутньому планується розширити перелік СУБД.

В базі даних зберігається вся динамічна інформація, що може бути редагована адміністраторами та викладачами, а саме: дані користувачів; групи студентів; завдання; групи тестових завдань, критерії оцінки груп, питання з еталонними запитами; відповіді та оцінки.

Сервер включає основний функціонал обробки даних, відновлення тестової бази даних. Певний функціонал та доступ до даних з бази даних винесено до окремих бібліотек класів. Також в окрему бібліотеку класів винесені ресурси локалізації. На рисунку 1 зображена діаграма компонентів.

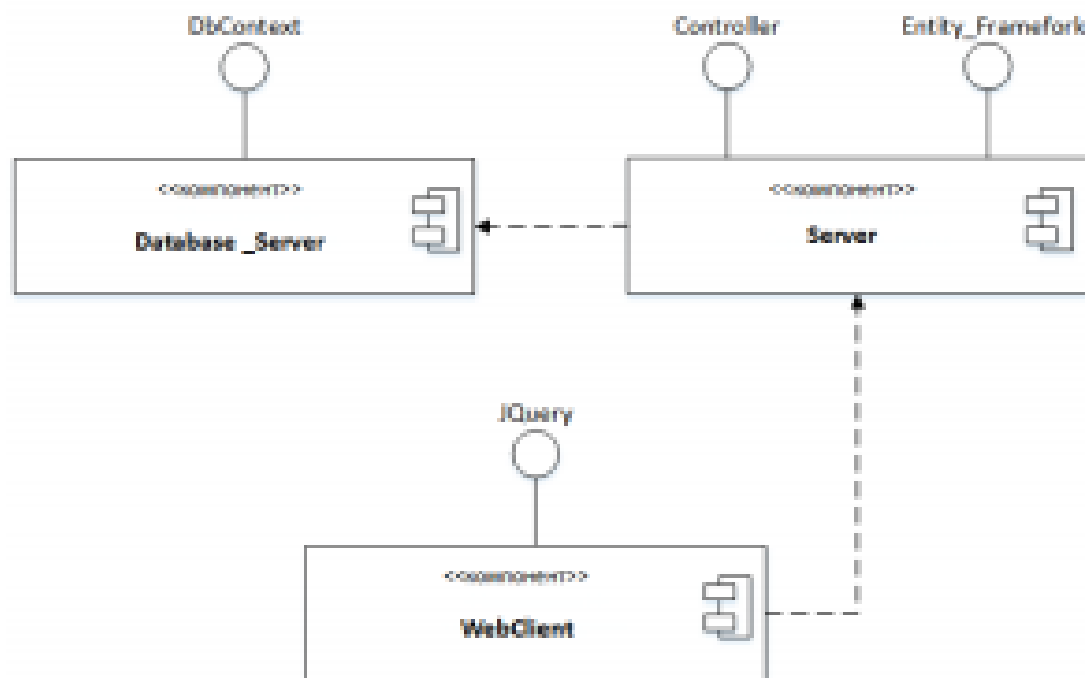


Рис. 1 – Діаграма компонентів

Пропонується використовувати компонент серверу, базу даних та компонент «веб-клієнт». При програмній реалізації з використанням платформи .Net, компонент «база даних» використовує `DbContext`, що потрібен для роботи `EntityFramework`. Компонент «Сервер» використовує «`Controller`» для формування сторінок та обробки запитів від клієнта і обробки потрібних для відображення даних, а також використовує бібліотеку `EntityFramework`. Веб-клієнт, в свою чергу, використовує бібліотеку `jQuery`. Усі компоненти взаємодіють між собою: клієнт робить запит до серверу, сервер запитує дані у серверу БД, а тоді повертає їх клієнту.

В системі наявні такі ролі для зареєстрованих користувачів: Адміністратор, Викладач та Студент. Кожна з ролей має окремий набір функцій.

Роль Адміністратор надає можливість керувати користувачами та модифікувати їх дані (окрім оцінок і результатів тестування). Крім того Адміністратор має можливість керувати тестовою базою даних, змінюючи чи відновлюючи її. Роль Викладач дозволяє створювати та редагувати групи студентів, а також надає можливість додавання і редагування завдань, формування контрольних робіт, перегляд результатів тестування та формування звітності за цими результатами. Роль Студент надає доступ до функціоналу вибору та проходження тестування, а також перегляду власних результатів у загальній формі та після конкретного тестування.

На рисунку 2 наведена діаграма послідовностей, що відображає взаємодію об'єктів під час процесу відкриття користувачем сторінки з даними з бази, редагування даних та збереження у системі.

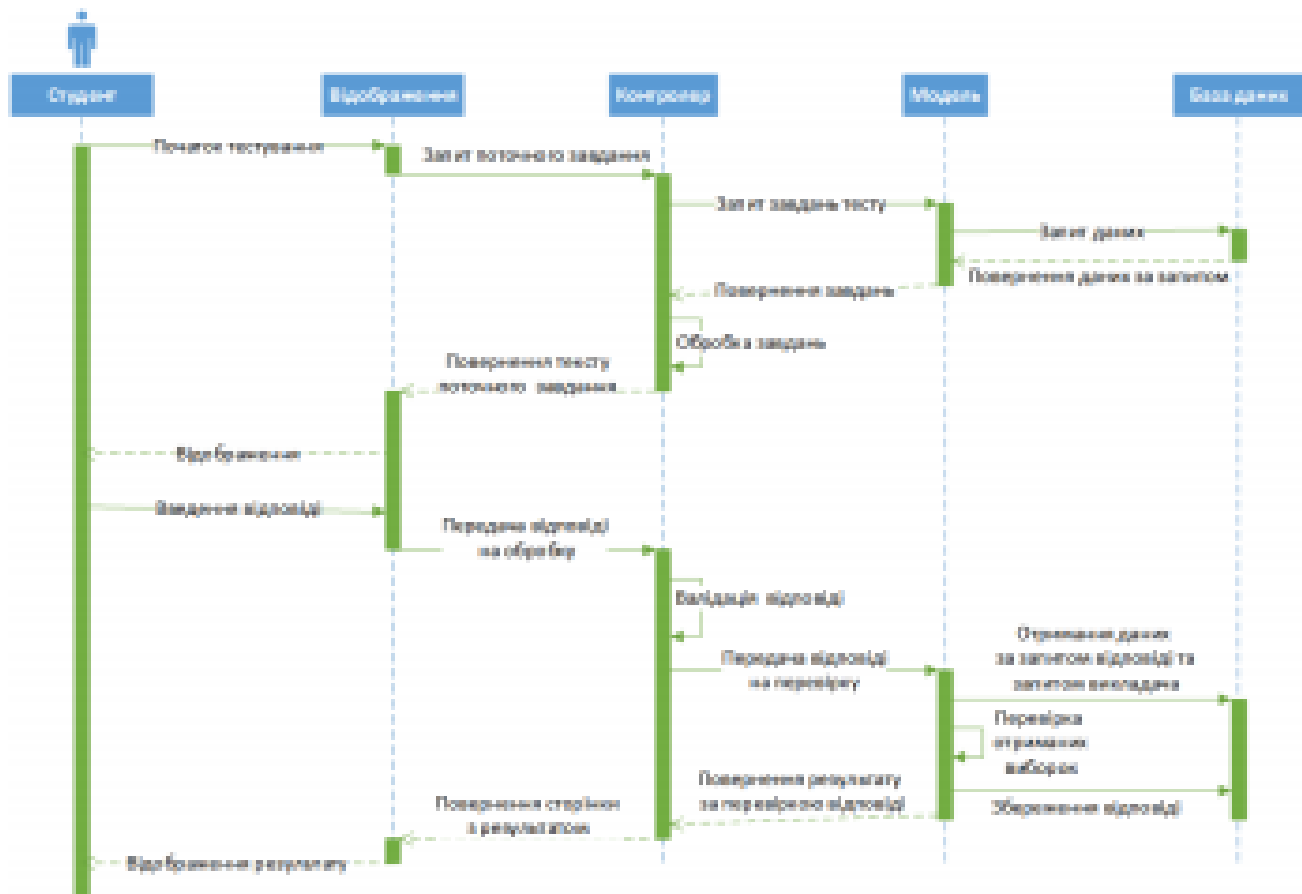


Рис. 2 – Діаграма послідовності

В подальшому розвиток системи можливий у декількох напрямках: додавання нового функціоналу з тестування видів запитів щодо забезпечення цілісності даних (indexes, constraints), додавання підтримки додаткових СУБД.