

УДК 004.9

Березоручька О.В.¹, Вичужанін В.В.²

¹ студ. гр. АД-191 НУ «Одеська політехніка»

² проф. НУ «Одеська політехніка»

РОЗРОБКА NOSQL БАЗИ ДАНИХ ДЛЯ СППР УПРАВЛІННЯ ПРОГРАМНИМИ ПРОЄКТАМИ

SQL та NoSQL бази даних розрізняються підходами до організації даних, що впливає на їх застосування в системах підтримки прийняття рішень (СППР). SQL бази даних зберігають інформацію в таблицях зі строго визначеними схемами, що забезпечує високу цілісність і підтримку складних запитів. NoSQL бази даних пропонують гнучкішу організацію даних у вигляді документів, колекцій або графів, що дозволяє створювати динамічні, вкладені структури без жорсткої схеми.

Для СППР управління програмними проєктами важлива здатність швидко додавати нові типи даних і поля, наприклад, історію змін, без порушення цілісності бази. Динамічна структура NoSQL спрощує це

завдання, уникаючи необхідності зміни всієї бази або складних міграцій, які часто потрібні в SQL-системах. У SQL базах такі зміни потребують корекції таблиць та зв'язків (через primary і foreign keys), що може знижувати продуктивність і вимагати додаткових тестів.

У цьому дослідженні будуть розглянуті можливості MongoDB як приклад ефективної NoSQL бази даних для СППР. MongoDB забезпечує зберігання пов'язаних даних у вкладених структурах, що дозволяє, наприклад, зберігати завдання разом із коментарями та виконавцями в одному документі. Така організація знижує потребу в приєднаннях (joins), характерних для SQL, оптимізуючи процес читання й підвищуючи продуктивність навіть при великих обсягах даних. MongoDB також дозволяє зберігати дані з неоднорідною структурою, що особливо корисно для сутностей із різними підтипами (наприклад, різні типи завдань), оскільки зберігаються лише ті поля, які мають значення. Це зменшує використання пам'яті та спрощує структуру даних, підвищуючи ефективність системи.

NoSQL бази даних забезпечують високу горизонтальну масштабованість, дозволяючи розподіляти дані між кількома серверами та обробляти тисячі одночасних з'єднань [3]. Це особливо актуально для команд із численними користувачами та великими обсягами даних, які працюють у реальному часі. Треба зазначити, що забезпечення безпеки даних у MongoDB є важливим аспектом для розроблюваної системи, що працює з конфіденційною інформацією про проекти, команди та клієнтів. MongoDB підтримує шифрування даних у спокої (at-rest) та під час передачі (in-transit), що забезпечує захист від несанкціонованого доступу під час мережевої передачі та запобігає витокам при зберіганні.

Для зберігання великих файлів, таких як вкладення, використовується механізм GridFS, який розбиває файли на частини та зберігає їх як окремі документи, що забезпечує доступність і ефективне зберігання [2]. Альтернативно, MongoDB може бути інтегрована з зовнішніми сховищами, такими як Amazon S3, для роботи з великими даними, що потребують специфічного зберігання.

Основними сутностями СППР для управління програмними проектами є Company, User, Project, Task і Team, кожна з яких містить специфічні поля для підтримки структурованості та гнучкості даних (рис. 1).

Company зберігає базову інформацію про компанію, включаючи назву, контактний email, загальний опис, а також списки працівників і проектів, що пов'язує компанію з учасниками та завданнями. User представляє користувача системи з контактними даними (ім'я, прізвище, телефон, унікальний email) і зашифрованим паролем із вимогами безпеки. User також містить роль у системі (наприклад, адміністратор) і професійну позицію (наприклад, QA чи Dev), що полегшує розподіл функцій і доступу. Project

охоплює назву, опис, команду, завдання, статус і вкладення, забезпечуючи контроль за робочим процесом проекту. Task деталізує завдання з полями для назви, опису, відповідального працівника, статусу, пріоритету, дедлайну, необхідних навичок, коментарів, витраченого часу та оцінки, що дозволяє оцінювати результати. Team об'єднує працівників у рамках конкретного проекту. Comment містить інформацію про автора, зміст, час створення та відповідне завдання, забезпечуючи зручну комунікацію. Така структура підтримує комплексне управління проектними даними та оптимізує процес прийняття рішень.

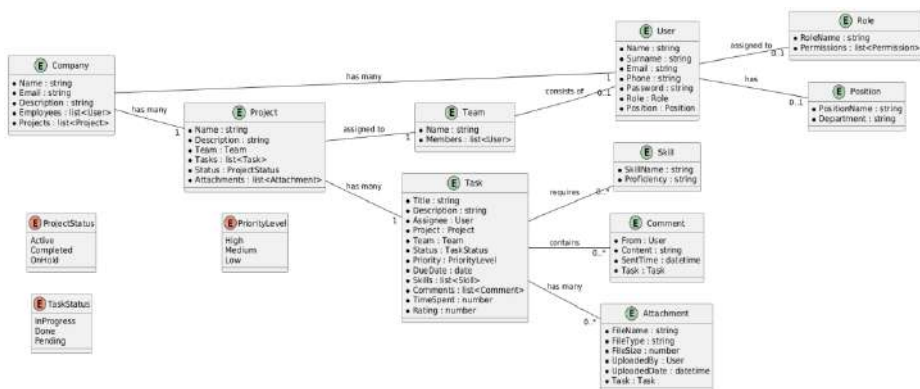


Рисунок 1 – ER-діаграма сутностей системи.

Система підтримки прийняття рішень для управління проектами реалізується на платформі .NET, що забезпечує надійну основу для розробки з високою продуктивністю та інтеграцією. Серверна частина використовує ASP.NET Core Web API, який дозволяє ефективно обробляти запити, управляти доступом за ролями та забезпечувати стабільний хостинг API. Інтерфейс створюється за допомогою Blazor, що об'єднує клієнтську та серверну частини в межах однієї екосистеми .NET, спрощуючи розробку та обслуговування. Для алгоритмів машинного навчання застосовано ML.NET, а MongoDB використовується як основна база даних для гнучкого зберігання інформації, що відповідає динамічній структурі проектів.

Для інтеграції .NET із MongoDB використовується офіційна бібліотека MongoDB.Driver, що забезпечує стабільну та ефективну роботу з базою даних. Бібліотека підтримує повний набір CRUD-операцій (створення, читання, оновлення, видалення) і дає змогу налаштовувати запити для підвищення продуктивності. Підключення здійснюється через URL сервера з параметрами для захищеного з'єднання, а класи для кожної сутності спрощують управління даними через API за допомогою методів

InsertOneAsync, FindAsync, ReplaceOneAsync та DeleteOneAsync. Оптимізація взаємодії включає налаштування пулу з'єднань для стабільності при високих навантаженнях і використання асинхронних методів `async/await` для зниження затримок, що забезпечує масштабованість і стабільність системи при значних обсягах даних та багатокористувацькій роботі [1].

У процесі розробки NoSQL-бази даних для СППР управління програмними проектами було обґрунтовано вибір MongoDB завдяки її гнучкості, масштабованості та можливості динамічного розширення структури даних. MongoDB підтримує вкладені структури, одночасні підключення та адаптується до змінних вимог проєктів. Інтеграція з .NET через `MongoDB.Driver` забезпечує стабільну архітектуру, оптимізуючи обробку даних і відповідність високим вимогам до продуктивності. Розроблені сутності та алгоритми ефективно інтегруються в архітектуру системи, підтримуючи прийняття зважених рішень у реальному часі, що критично важливо для управління сучасними програмними проєктами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Carter L. Beginning mongodb atlas with .NET: flexible and scalable document storage for .NET developers. Apress L. P., 2023.
2. Le T. T., Lam Pham X. Towards NoSQL databases: experiences from actual projects. IBDAP, Bangkok, Thailand, 1–2 September 2022. URL: <https://doi.org/10.1109/ibdap55587.2022.9907664>.
3. MongoDB: powerful and scalable data storage. O'Reilly Media, Incorporated, 2019. 425 p.