

МОДЕЛІ ТА ЗАСОБИ УПРАВЛІННЯ ЗАДАЧАМИ У ВЕБ-ОРІЄНТОВАНИХ СИСТЕМАХ

Вінницький національний технічний університет

Анотація. У роботі розглянуто підходи до побудови моделей управління задачами у веб-орієнтованих системах. Проаналізовано основні принципи організації задач, зокрема їх створення, класифікацію, призначення виконавців, зміну станів та контроль строків виконання. Розглянуто засоби реалізації веб-систем керування задачами на основі клієнт-серверної архітектури, програмних інтерфейсів взаємодії, механізмів автентифікації та баз даних. Обґрунтовано доцільність використання ролівої моделі доступу для розмежування функцій користувачів у межах робочих просторів. Запропоновано узагальнену структуру веб-орієнтованої системи управління задачами, що забезпечує підтримку життєвого циклу задачі від її створення до завершення та аналізу результатів.

Ключові слова: веб-орієнтована система, модель управління задачами, життєвий цикл задачі, ролівий доступ, клієнт-серверна архітектура, REST API.

Abstract. The paper considers approaches to developing task management models in web-oriented systems. The main principles of task organization are analyzed, including task creation, classification, assignment of performers, state changes, and deadline control. The tools for implementing web-based task management systems based on client-server architecture, application programming interfaces, authentication mechanisms, and databases are considered. The feasibility of using a role-based access model to differentiate user functions within workspaces is substantiated. A generalized structure of a web-oriented task management system is proposed, which supports the task life cycle from creation to completion and result analysis.

Keywords: web-oriented system, task management model, task life cycle, role-based access, client-server architecture, REST API.

Актуальність дослідження

В умовах активного розвитку цифрових технологій значна частина навчальної, проектної та організаційної діяльності переноситься у веб-середовище. Це зумовлює потребу в інструментах, які дозволяють не лише зберігати інформацію про завдання, а й керувати процесом їх виконання, координувати дії учасників, контролювати строки та оцінювати результати роботи.

Управління задачами є важливою складовою багатьох веб-орієнтованих систем, оскільки саме задачі відображають конкретні дії, які мають бути виконані користувачами. Для ефективної роботи такої системи необхідно визначити модель задачі, її стани, зв'язки з користувачами, пріоритетами, строками виконання та робочими просторами.

На практиці часто використовуються готові сервіси та методики для організації задач і командної роботи, однак вони не завжди відповідають потребам конкретного навчального або робочого процесу. Одні підходи орієнтовані на просте візуальне планування та переміщення задач між етапами виконання за канбан-моделлю [1], інші — на складніше управління розробкою за Scrum-підходом [2], що може бути надмірним для невеликих команд. Тому актуальним є дослідження узагальнених моделей і засобів, які можуть бути покладені в основу власної веб-орієнтованої системи управління задачами.

Постановка проблеми та її зв'язок з науковими і практичними завданнями

Проблема дослідження полягає у визначенні такої моделі управління задачами, яка забезпечує зрозумілу структуру даних, ефективну взаємодію користувачів і можливість подальшого розвитку веб-орієнтованої системи. Для цього необхідно врахувати не лише основні властивості задачі, а й механізми її переходу між станами, правила доступу користувачів, засоби фільтрації та способи відображення результатів.

З наукової точки зору важливим є формалізований опис задачі як об'єкта системи, що має визначений набір атрибутів і зв'язків з іншими сутностями. З практичної точки зору така модель повинна бути придатною для реалізації у вигляді програмного продукту, який може

використовуватися в навчальних групах, малих командах, проектних середовищах або організаційних підрозділах.

Практичне значення полягає в тому, що правильно побудована модель управління задачами дозволяє спростити планування роботи, зменшити кількість неузгодженостей між учасниками, підвищити прозорість виконання завдань і забезпечити зручний контроль за поточним станом процесів.

Аналіз досліджень та публікацій

У сучасних дослідженнях і практичних розробках управління задачами часто розглядається як частина ширших систем управління проектами, робочими процесами або командною взаємодією. Такі системи базуються на поняттях задачі, виконавця, ролі, статусу, пріоритету, строку виконання та історії змін.

Технічна реалізація веб-орієнтованих систем управління задачами зазвичай ґрунтується на клієнт-серверній архітектурі. Для взаємодії між частинами системи широко використовується REST-підхід, який дає змогу подавати основні об'єкти системи як ресурси та виконувати операції над ними за допомогою HTTP-запитів [3].

У межах такої взаємодії основні HTTP-методи можуть застосовуватися для виконання типових операцій над задачами та пов'язаними сутностями. Зокрема, GET використовується для отримання даних, POST — для створення нових записів, PUT або PATCH — для оновлення інформації, а DELETE — для видалення ресурсів [5].

Для забезпечення безпечної роботи застосовуються механізми автентифікації та авторизації, зокрема токен-орієнтовані підходи [4]. Це дозволяє перевіряти особу користувача під час звернення до захищених ресурсів і контролювати доступ до функцій системи.

Невирішені раніше частини проблеми

Попри наявність значної кількості інструментів для керування задачами, актуальною залишається проблема адаптації таких засобів до конкретних умов використання. Готові сервіси часто мають фіксовану логіку роботи або велику кількість функцій, які не завжди потрібні користувачам. Це може ускладнювати впровадження системи в невеликих командах або навчальних групах.

Недостатньо уваги приділяється узгодженню між логічною моделлю задачі та її програмною реалізацією. Наприклад, задача може розглядатися лише як запис із назвою та статусом, хоча на практиці вона має ширший життєвий цикл, включає коментарі, вкладення, історію змін, зв'язок із виконавцями та обмеження за правами доступу.

Також потребує уточнення питання вибору засобів реалізації, які дозволяють поєднати простоту інтерфейсу, безпеку, масштабованість і можливість інтеграції з іншими сервісами. Саме тому доцільним є дослідження моделей і засобів управління задачами як взаємопов'язаних складових веб-орієнтованої системи.

Мета статті

Метою статті є дослідження моделей та засобів управління задачами у веб-орієнтованих системах, визначення основних компонентів такої системи та обґрунтування підходу до організації задач, користувачів, ролей, статусів і засобів контролю виконання.

Виклад основного матеріалу та отримані наукові результати

Веб-орієнтовану систему управління задачами доцільно розглядати як програмне середовище, у якому користувачі можуть створювати, редагувати, призначати, виконувати та аналізувати задачі. Основою такої системи є модель задачі, що визначає її структуру, властивості та зв'язки з іншими об'єктами. Задача як об'єкт системи може описуватися такими основними атрибутами: назва, опис, автор, виконавець, статус, пріоритет, дата створення, термін виконання, належність до робочого простору та історія змін. Такий набір атрибутів дозволяє не лише зберігати інформацію про завдання, а й відстежувати його виконання у часі.

Життєвий цикл задачі може містити кілька станів: створена, запланована, у процесі виконання, на перевірку, завершена або відхилена. Перехід між станами має виконуватися відповідно до прав користувача та логіки робочого процесу. Наприклад, виконавець може змінити статус задачі на «у процесі виконання», тоді як завершення або перевірка можуть бути доступні менеджеру чи адміністратору.

Важливим елементом моделі є робочий простір, який об'єднує користувачів і задачі в межах певного навчального, робочого або проектного процесу. У межах одного робочого простору можуть існувати різні ролі користувачів. Адміністратор відповідає за налаштування простору та керування

учасниками, менеджер — за створення й розподіл задач, а виконавець — за виконання призначених завдань.

Засоби управління задачами мають забезпечувати створення, редагування, видалення та перегляд задач, а також можливість пошуку й фільтрації за статусом, пріоритетом, виконавцем або строком виконання. Для підвищення інформативності системи доцільно використовувати календарне подання, списки, дошки станів і статистичні панелі. Зокрема, дошки станів дають змогу групувати задачі за етапами виконання, візуально контролювати перебіг роботи та швидко визначати поточний стан окремих завдань [8].

Технічно така система може бути реалізована на основі клієнт-серверної архітектури. Клієнтська частина відповідає за взаємодію з користувачем і відображення даних, а серверна — за обробку запитів, перевірку прав доступу, виконання бізнес-логіки та збереження інформації. Такий поділ дозволяє незалежно розвивати інтерфейс і серверну логіку.

Для обміну даними між клієнтом і сервером доцільно використовувати REST API. У такому випадку користувачі, задачі, робочі простори, коментарі та статуси подаються як окремі ресурси. Це спрощує структуру системи, полегшує тестування та дає змогу в майбутньому інтегрувати систему з мобільними застосунками або зовнішніми сервісами. Зберігання даних може здійснюватися у реляційній базі даних. Реляційна модель є зручною для опису зв'язків між користувачами, задачами, ролями, робочими просторами та коментарями. Вона також дозволяє забезпечити цілісність даних і виконувати складні запити для формування статистики. Для реалізації такого підходу може використовуватися PostgreSQL як система керування реляційними базами даних [6].

Безпека веб-орієнтованої системи управління задачами забезпечується автентифікацією користувачів і перевіркою їхніх прав доступу. Токен-орієнтована автентифікація дозволяє підтверджувати особу користувача під час звернення до захищених ресурсів, а рольова модель визначає, які дії може виконувати конкретний користувач у межах певного робочого простору. Під час розробки таких систем також необхідно враховувати вимоги безпеки веб-застосунків, зокрема захист механізмів автентифікації, перевірку авторизації та запобігання несанкціонованому доступу до даних [7].

Отримані результати полягають у визначенні узагальненої моделі управління задачами, яка поєднує модель задачі, модель користувачів, рольову модель доступу, життєвий цикл задачі та засоби візуалізації. Такий підхід дозволяє створювати веб-орієнтовані системи, придатні для організації навчальних, робочих і проектних процесів.

Наукова новизна полягає в узагальненні підходів до побудови моделей і засобів управління задачами у веб-орієнтованих системах з урахуванням життєвого циклу задачі, рольового доступу, клієнт-серверної взаємодії та можливостей подальшої інтеграції з іншими сервісами.

Висновки та перспективи подальших досліджень

У статті досліджено моделі та засоби управління задачами у веб-орієнтованих системах. Визначено, що ефективна система такого типу має включати модель задачі, робочі простори, ролі користувачів, механізми контролю строків, зміну статусів, фільтрацію та візуалізацію результатів.

Запропонований підхід дозволяє організувати повний життєвий цикл задачі від створення до завершення, забезпечити контроль доступу користувачів і підвищити прозорість командної роботи. Використання клієнт-серверної архітектури, REST API та реляційної бази даних створює основу для масштабованої та гнучкої веб-орієнтованої системи.

Перспективними напрямками подальших досліджень є розроблення адаптивних моделей пріоритетизації задач, інтеграція з календарними сервісами, удосконалення механізмів сповіщень, аналіз продуктивності користувачів і застосування інтелектуальних методів для прогнозування строків виконання завдань.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kanban Guide. The Official Guide to the Kanban Method [Електронний ресурс]. – Режим доступу: <https://kanbanguides.org/> (дата звернення: 05.05.2026).
2. Schwaber K., Sutherland J. The Scrum Guide [Електронний ресурс]. – Режим доступу: <https://scrumguides.org/> (дата звернення: 05.05.2026).
3. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures [Електронний ресурс] : doctoral dissertation / R. T. Fielding. – University of California, Irvine, 2000. – Режим доступу: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm> (дата звернення: 05.05.2026).
4. Auth0. JSON Web Tokens [Електронний ресурс]. – Режим доступу: <https://jwt.io/> (дата звернення: 05.05.2026).

5. MDN Web Docs. HTTP request methods [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Methods> (дата звернення: 05.05.2026).
6. PostgreSQL Global Development Group. PostgreSQL Documentation [Електронний ресурс]. – Режим доступу: <https://www.postgresql.org/docs/> (дата звернення: 05.05.2026).
7. OWASP Foundation. Web Application Security Testing Guide [Електронний ресурс]. – Режим доступу: <https://owasp.org/www-project-web-security-testing-guide/> (дата звернення: 05.05.2026).
8. Atlassian. What is a kanban board? [Електронний ресурс]. – Режим доступу: <https://www.atlassian.com/agile/kanban/boards> (дата звернення: 05.05.2026).

Новак Віталій Ігорович – студент групи 2ПІ-22б, факультет інформаційних технологій та комп’ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: vitalik.n1803@gmail.com

Науковий керівник: **Стахов Олексій Ярославович** – PhD, старший викладач кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця.

Novak Vitalii I. – student of group 2PI-22b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: vitalik.n1803@gmail.com

Scientific supervisor: **Stakhov Oleksii Y.** – PhD, Senior Lecturer of the Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia.