International Science Journal of Engineering & Agriculture
2026; 5(3): 30-47
https://isg-journal.com/isjea
doi: 10.46299/j.isjea.20260503.04
ISSN: 2720-6319

INTERNATIONAL SCIENCE GROUP

Проектування безпечної системи управління командною роботою з багаторівневою автентифікацією на базі Django

Юрій Тулашвілі
Луцький національний технічний університет, Луцьк, Україна
ORCID: 0000-0002-0780-9529

Віктор Кошелюк
Луцький національний технічний університет, Луцьк, Україна
ORCID: 0000-0002-4136-5087

Богдан Морозюк
Луцький національний технічний університет, Луцьк, Україна
ORCID: 0009-0002-8342-2518

Анотація: Зростання масштабів віддаленої командної роботи та активне використання web-платформ у сфері розробки програмного забезпечення актуалізують проблему забезпечення безпечного доступу до корпоративних інформаційних ресурсів. У сучасних системах управління командною взаємодією особливого значення набувають механізми автентифікації користувачів, контроль доступу до даних і захист від поширених кіберзагроз. Недостатній рівень безпеки подібних платформ може призводити до компрометації облікових записів, втрати конфіденційної інформації та порушення цілісності робочих процесів. У зв'язку з цим виникає потреба у створенні web-орієнтованих систем, які поєднують функціональність управління командною роботою з сучасними засобами багаторівневого захисту. Метою дослідження є проектування безпечної системи управління командною роботою на базі Django із реалізацією багаторівневої автентифікації та механізмів контролю доступу до інформаційних ресурсів. У межах роботи проаналізовано сучасні підходи до побудови захищених web-застосунків, визначено ключові вимоги до архітектури системи та обґрунтовано вибір технологічного стеку для реалізації серверної частини програмного забезпечення. Методологічну основу дослідження становлять методи системного аналізу, об'єктно-орієнтованого проектування, функціонального моделювання та технології web-програмування. Для реалізації програмного рішення використано Django і Django REST Framework, що забезпечують підтримку модульної архітектури, REST-взаємодії та інтегрованих механізмів безпеки. У системі реалізовано багаторівневу автентифікацію, рольову модель доступу, JWT-механізми авторизації, а також засоби захисту від атак типу CSRF, XSS та SQL Injection. У результаті дослідження сформовано архітектурну модель web-системи, орієнтованої на підтримку командної взаємодії в багатокористувацькому середовищі. Розроблене програмне рішення забезпечує централізоване управління користувачами, контроль доступу до функціональних модулів, журналювання дій та захищену передачу даних між клієнтською і серверною частинами системи. Проведене тестування показало підвищення рівня безпеки доступу до інформаційних ресурсів та стабільність роботи системи при одночасній роботі кількох користувачів. Отримані результати підтверджують доцільність використання Django для створення безпечних систем управління командною роботою. Подальші дослідження можуть бути спрямовані на інтеграцію адаптивних механізмів автентифікації, використання поведінкового аналізу для виявлення аномальної активності користувачів та впровадження мікросервісної архітектури для підвищення масштабованості й відмовостійкості системи.

Ключові слова: Django; багаторівнева автентифікація; інформаційна безпека; web-система управління командною роботою; контроль доступу.

1. Вступ

За останні роки web-системи командної взаємодії перетворилися з допоміжних інструментів координації роботи на критично важливий елемент інфраструктури IT-компаній. Більшість процесів, пов'язаних із плануванням задач, управлінням програмними проєктами, комунікацією між учасниками команди та контролем робіт, сьогодні реалізується через мережеві платформи з постійним віддаленим доступом. Така трансформація значно спростила організацію distributed-команд, однак одночасно створила новий спектр ризиків, пов'язаних із безпекою корпоративних даних і стабільністю функціонування web-середовищ [1, 2].

Проблема захисту систем управління командною роботою набуває особливої актуальності в умовах зростання кількості кіберінцидентів, пов'язаних із компрометацією облікових записів користувачів. Згідно з сучасними аналітичними звітами у сфері кібербезпеки, значна частина атак на корпоративні сервіси пов'язана не зі складними технічними експлойтами, а з використанням слабких механізмів автентифікації, повторного використання паролів або недостатнього контролю доступу до інформаційних ресурсів. Для систем командної взаємодії така проблема є критичною, оскільки компрометація одного облікового запису потенційно відкриває доступ до внутрішньої документації, проєктних матеріалів, каналів комунікації та інших чутливих даних організації [3, 4].

Поширення віддалених і гібридних форматів роботи лише посилило зазначені ризики. Якщо раніше значна частина корпоративної взаємодії здійснювалася в межах локальної інфраструктури підприємства, то сьогодні доступ до систем управління проєктами часто виконується з різних мережевих середовищ, особистих пристроїв або зовнішніх каналів зв'язку. За таких умов класичні підходи до автентифікації, що базуються виключно на логіні та паролі, втрачають ефективність. Навіть складні паролі не гарантують належного рівня захисту в ситуаціях, коли користувачі стають об'єктами фішингових атак або коли облікові дані потрапляють до відкритих баз витоків інформації [5, 6].

У сучасній практиці розробки програмного забезпечення проблема безпеки дедалі частіше розглядається не як окремий функціональний модуль, а як фундаментальний архітектурний принцип. Концепція security-by-design передбачає інтеграцію механізмів захисту ще на етапі проектування системи, що дозволяє зменшити кількість потенційних вразливостей та уникнути ситуацій, коли засоби безпеки впроваджуються вже після завершення основної розробки. Для web-систем управління командною роботою такий підхід є особливо важливим, оскільки вони функціонують у багатокористувацькому середовищі з постійним обміном інформацією між клієнтськими та серверними компонентами [7, 8].

Окремої уваги заслуговує проблема багаторівневої автентифікації. На відміну від традиційних моделей доступу, багаторівневі механізми перевірки користувача передбачають використання кількох незалежних факторів підтвердження особи. Це можуть бути одноразові коди, токени, мобільні застосунки підтвердження доступу або інші засоби верифікації. Практика використання multi-factor authentication демонструє суттєве зниження ризику несанкціонованого доступу навіть у випадках компрометації паролів. Водночас інтеграція таких механізмів у системи командної взаємодії потребує врахування не лише аспектів безпеки, а й факторів зручності користування, продуктивності та безперервності робочих процесів [9, 10].

Аналіз наукових праць і сучасних програмних платформ свідчить про те, що більшість досліджень у сфері web-безпеки концентрується на окремих аспектах захисту: безпеці API, криптографічних механізмах, виявленні атак або управлінні ролями користувачів. Значно менше уваги приділяється комплексному проектуванню систем, у яких механізми автентифікації, контроль доступу та інструменти управління командною роботою формують єдине інтегроване середовище. Існуючі комерційні рішення, орієнтовані на Agile-команди, забезпечують стандартний набір функцій управління задачами, однак мають обмежені можливості адаптації до специфічних вимог безпеки окремих організацій [11, 12, 13].

Крім того, централізовані корпоративні платформи часто характеризуються складністю масштабування та залежністю від зовнішньої інфраструктури. Для невеликих або середніх IT-команд використання подібних рішень не завжди є економічно виправданим, особливо у випадках, коли виникає потреба інтеграції додаткових механізмів захисту або спеціалізованих сценаріїв контролю доступу. Саме тому актуальним напрямом залишається проектування гнучких web-систем, які можуть адаптуватися до конкретних умов використання без суттєвого ускладнення архітектури програмного забезпечення [14, 15].

У цьому контексті використання Django є доцільним з кількох причин. По-перше, фреймворк підтримує модульний принцип побудови програмного забезпечення, що спрощує інтеграцію окремих механізмів безпеки в загальну структуру системи. По-друге, Django містить вбудовані засоби захисту від поширених web-загроз, зокрема CSRF, XSS та SQL Injection атак. По-третє, використання ORM-моделі, централізованого управління сесіями та role-based access control дозволяє реалізувати безпечну взаємодію між користувачами та серверною частиною застосунку без необхідності створення великої кількості додаткових компонентів.

Теоретичну основу дослідження становлять підходи до проектування інформаційних систем, моделі захищеної клієнт-серверної взаємодії, принципи багаторівневої автентифікації та методи управління доступом у багатокористувацьких середовищах. Важливим аспектом є також поєднання принципів web-програмування з практиками DevSecOps, відповідно до яких питання безпеки розглядаються як невід'ємна частина життєвого циклу програмного забезпечення.

Метою статті є проектування безпечної системи управління командною роботою на базі Django із реалізацією багаторівневої автентифікації, механізмів розмежування доступу та захищеної взаємодії між компонентами web-застосунку. Для досягнення поставленої мети передбачено аналіз сучасних підходів до побудови захищених web-систем, визначення архітектурних вимог до програмного рішення та реалізацію моделі безпечної взаємодії користувачів у багатокористувацькому середовищі.

Практичне значення роботи полягає у можливості використання запропонованого підходу під час створення корпоративних систем підтримки командної роботи, платформ управління Agile-проєктами та інших web-сервісів, що потребують підвищеного рівня захисту інформаційних ресурсів. Отримані результати можуть бути використані не лише у сфері програмної інженерії, але й у дослідженнях, пов'язаних із захистом distributed-систем, побудовою DevSecOps-архітектур та автоматизацією процесів моніторингу інформаційної безпеки.

2. Об'єкт і предмет дослідження

Об'єктом дослідження є процеси функціонування та захисту web-орієнтованих систем управління командною роботою, що використовуються в середовищі IT-проєктів для координації взаємодії між учасниками команди, розподілу задач, контролю доступу до інформаційних ресурсів і підтримки спільної роботи в режимі віддаленої або гібридної взаємодії. У сучасних умовах подібні системи фактично виконують роль централізованого цифрового середовища, у межах якого здійснюються основні управлінські та комунікаційні процеси команди.

Особливістю таких систем є поєднання двох взаємопов'язаних компонентів: функціонального та безпекового. З одного боку, платформа повинна забезпечувати ефективне управління задачами, підтримку Agile-процесів, збереження проєктної документації та швидкий обмін інформацією між користувачами. З іншого боку, система має гарантувати захист корпоративних даних, контроль привілеїв доступу та стійкість до зовнішніх і внутрішніх загроз. Практика експлуатації подібних web-рішень показує, що саме баланс між зручністю використання та рівнем безпеки є одним із найскладніших аспектів їх проектування.

У технологічному аспекті об’єкт дослідження являє собою багаторівневу web-систему, побудовану за клієнт-серверною архітектурою. Її структура передбачає наявність серверної частини, бази даних, механізмів автентифікації користувачів, модулів контролю доступу та клієнтського інтерфейсу. Серверний рівень забезпечує обробку запитів, виконання бізнес-логіки та взаємодію з базою даних, тоді як клієнтська частина відповідає за візуалізацію інформації та організацію взаємодії користувача із системою.

У межах дослідження серверна архітектура базується на використанні Django, який забезпечує підтримку модульного принципу розробки, ORM-механізмів, системи керування сесіями та вбудованих засобів захисту web-застосунків. Вибір саме Django обумовлений тим, що цей фреймворк поєднує відносно високу швидкість розробки із достатнім рівнем гнучкості для реалізації складних механізмів автентифікації та управління ролями користувачів. Як система зберігання даних використовується PostgreSQL, що дозволяє забезпечити централізоване управління інформаційними ресурсами та підтримку складних структур взаємозв’язків між об’єктами системи.

Ключовими характеристиками досліджуваного об’єкта є рівень захищеності даних, масштабованість, стабільність роботи при багатокористувацькому доступі, ефективність механізмів автентифікації та швидкість обробки запитів. Важливим параметром також виступає здатність системи підтримувати гнучке розмежування прав доступу залежно від ролі користувача, типу задачі або контексту використання інформаційних ресурсів. Для корпоративних систем командної взаємодії це має принципове значення, оскільки різні категорії користувачів потребують різного рівня доступу до проєктних даних.

Предметом дослідження є методи та технології проектування безпечних web-систем управління командною роботою із застосуванням багаторівневої автентифікації, рольових моделей доступу та механізмів захисту інформаційних ресурсів у багатокористувацькому середовищі. Особливу увагу приділено інтеграції багатофакторної автентифікації в архітектуру системи, реалізації role-based access control та захисту серверної частини від типових web-загроз.

Серед зарубіжних аналогів досліджуваного об’єкта можна виділити Jira, Asana, Trello та Monday.com. Ці системи забезпечують підтримку управління проєктами, командної координації та інтеграції із зовнішніми сервісами. Водночас їх архітектура переважно орієнтована на універсальність використання, що обмежує можливості глибокої адаптації механізмів безпеки до специфічних вимог окремих організацій.

Практичний досвід використання існуючих платформ виявляє низку проблем, які особливо помітні в умовах інтенсивної експлуатації. Насамперед йдеться про залежність від централізованої хмарної інфраструктури, складність інтеграції нестандартних механізмів автентифікації та недостатню гнучкість у налаштуванні політик доступу. Крім того, у багатьох системах механізми багатофакторної автентифікації реалізовані як додаткові модулі, а не як інтегрований елемент архітектури безпеки, що ускладнює адміністрування та підвищує ризик конфігураційних помилок.

Ще однією проблемою є зростання навантаження на серверні компоненти при одночасній роботі великої кількості користувачів, особливо у випадках активного використання механізмів журналювання, моніторингу активності та перевірки автентифікаційних подій. У реальних умовах експлуатації це може призводити до зниження продуктивності системи та збільшення часу обробки запитів.

Таким чином, необхідність проектування безпечної системи управління командною роботою з інтегрованими механізмами багаторівневої автентифікації обумовлена не лише зростанням кількості кіберзагроз, але й практичними обмеженнями існуючих платформ, які не завжди забезпечують достатній рівень адаптивності, масштабованості та захисту інформаційного середовища.

3. Мета та задачі дослідження

Розвиток web-технологій і перехід значної частини IT-команд до віддалених або гібридних моделей роботи суттєво змінили вимоги до систем управління командною взаємодією. Якщо раніше подібні платформи розглядалися переважно як інструмент координації задач і внутрішньої комунікації, то сьогодні вони фактично виконують функції централізованого середовища зберігання та обробки корпоративної інформації. За таких умов проблема інформаційної безпеки виходить за межі окремого технічного аспекту й стає одним із ключових критеріїв ефективності системи в цілому.

Проведений аналіз сучасних платформ управління командною роботою показав, що більшість існуючих рішень забезпечує достатній рівень функціональності для організації Agile-процесів, однак механізми захисту інформаційного середовища часто залишаються стандартизованими та обмежено адаптованими до специфіки конкретної організації. У практиці використання таких систем найбільш проблемними виявляються питання гнучкого розмежування доступу, централізованого управління ролями користувачів та інтеграції багаторівневих механізмів автентифікації без суттєвого ускладнення архітектури програмного забезпечення.

Додатковою проблемою є залежність багатьох корпоративних платформ від традиційної моделі автентифікації, що базується лише на використанні логіна та пароля. У сучасних умовах така модель вже не забезпечує достатнього рівня захисту, особливо в середовищах із великою кількістю користувачів і постійним віддаленим доступом до інформаційних ресурсів. Компрометація облікових даних, фішингові атаки або помилки конфігурації можуть призвести не лише до втрати доступу до окремих облікових записів, а й до порушення цілісності всієї інформаційної інфраструктури команди.

Практичний досвід експлуатації існуючих систем також демонструє складність адаптації типових механізмів безпеки до потреб конкретних організацій. У багатьох випадках інтеграція додаткових рівнів автентифікації або реалізація нестандартних політик доступу вимагає використання сторонніх сервісів, що ускладнює адміністрування системи та підвищує залежність від зовнішньої інфраструктури. Крім того, при збільшенні кількості активних користувачів суттєво зростає навантаження на серверні компоненти, особливо у випадках використання механізмів журналювання подій, моніторингу активності та перевірки прав доступу в реальному часі.

У зв’язку з цим виникає потреба у створенні безпечної web-системи управління командною роботою, у якій механізми автентифікації та контролю доступу будуть інтегровані в архітектуру програмного забезпечення як базові елементи функціонування системи, а не як додаткові модулі безпеки.

Метою дослідження є проектування безпечної системи управління командною роботою на базі Django із реалізацією багаторівневої автентифікації, механізмів role-based access control та засобів захисту інформаційних ресурсів у багатокористувацькому середовищі.

Досягнення поставленої мети передбачає вирішення комплексу взаємопов’язаних задач. Насамперед необхідно провести аналіз сучасних підходів до побудови web-систем управління командною взаємодією та дослідити особливості реалізації механізмів інформаційної безпеки в корпоративних середовищах. Окрему увагу доцільно приділити вивченню моделей багатофакторної автентифікації, їх переваг, обмежень та можливостей інтеграції в системи командної роботи.

Наступним етапом є формування функціональних і нефункціональних вимог до програмного рішення, зокрема вимог щодо продуктивності, масштабованості, стабільності та захищеності системи. Важливим завданням дослідження виступає проектування архітектури web-застосунку із використанням Django, що дозволить реалізувати модульну структуру програмного забезпечення та централізоване управління механізмами автентифікації й контролю доступу.

У межах практичної реалізації необхідно забезпечити підтримку багаторівневої автентифікації, механізмів управління ролями користувачів, журналювання подій та захисту REST API від поширених web-загроз. Додатково передбачається реалізація засобів моніторингу активності користувачів і перевірка ефективності роботи системи в умовах багатокористувацького навантаження.

Завершальним етапом дослідження є тестування розробленого програмного рішення та оцінювання його ефективності з точки зору підвищення рівня інформаційної безпеки, стабільності роботи й адаптивності до потреб сучасних IT-команд.

Таким чином, реалізація поставлених задач має забезпечити створення web-системи, у якій механізми захисту інтегруються в загальну архітектуру управління командною роботою та формують цілісне безпечне інформаційне середовище для підтримки корпоративної взаємодії.

4. Аналіз літератури

Питання проектування безпечних web-систем управління командною роботою впродовж останніх років набуло значної актуальності у зв'язку зі стрімким розвитком distributed-середовищ, поширенням віддалених моделей співпраці та зростанням кількості кіберзагроз, спрямованих на корпоративні інформаційні ресурси. Сучасні наукові дослідження у цій сфері здебільшого зосереджуються на трьох взаємопов'язаних напрямах: проектуванні систем командної взаємодії, реалізації механізмів багаторівневої автентифікації та забезпеченні захисту web-застосунків від типових атак.

Одним із базових напрямів досліджень є вдосконалення архітектури інформаційних систем управління проєктами. У роботах [16, 17] значна увага приділяється принципам побудови багаторівневих інформаційних систем, організації модульної структури програмного забезпечення та питанням інтеграції клієнт-серверних компонентів у web-середовищі. Автори наголошують, що сучасні системи управління даними повинні проектуватися з урахуванням масштабованості, гнучкості та можливості адаптації до змінних умов експлуатації. Водночас у дослідженні основна увага приділена загальним принципам побудови інформаційних систем, тоді як механізми багаторівневої автентифікації та захисту корпоративної взаємодії розглядаються лише частково.

Окремий напрям досліджень пов'язаний із технологіями web-програмування та реалізацією серверної логіки інформаційних систем. У праці Vasudhar Sai Thokala [18] розглянуто сучасні підходи до побудови web-застосунків із використанням HTML, CSS, JavaScript та серверних технологій. Автор аналізує методи організації клієнт-серверної взаємодії, принципи реалізації REST-архітектури та механізми обробки запитів у web-системах. Практична цінність дослідження полягає у висвітленні підходів до побудови інтерактивних web-платформ, однак питання безпеки та захисту інформаційних ресурсів розглядаються переважно на базовому рівні.

Проблеми проектування захищених інформаційних систем детально досліджуються у роботах Hashmi et al. [19]. Автори розглядають методи побудови інформаційних систем із підвищеним рівнем безпеки, аналізують принципи управління доступом та організації захисту даних у багатокористувацькому середовищі. Значна увага приділяється питанням розмежування прав користувачів, контролю доступу до інформаційних ресурсів і централізованого адміністрування системи. Водночас дослідження орієнтоване переважно на загальні архітектурні принципи безпеки й не враховує специфіку web-орієнтованих платформ управління командною роботою.

Також серед зарубіжних досліджень вагоме місце займають роботи, присвячені використанню багатофакторної автентифікації в корпоративних системах. Зокрема, у дослідженні Siddiqui et al. [20] розглядаються принципи побудови безпечної архітектури програмного забезпечення, у межах яких механізми автентифікації інтегруються

безпосередньо в структуру програмної системи. Автор наголошує, що реалізація security-by-design є більш ефективною порівняно з підходами, де механізми безпеки додаються вже після завершення основної розробки. Запропоновані підходи є важливими для проектування web-систем із багаторівневим контролем доступу, однак у роботі не розглядаються практичні аспекти реалізації багатофакторної автентифікації в системах командної взаємодії.

Помітний внесок у розвиток web-технологій зробили дослідження, присвячені використанню фреймворка Django для побудови корпоративних web-застосунків. У роботі Manoj Kumar & Dr Rainu Nandal [21] детально розглядаються практики проектування масштабованих web-систем на базі Django, методи організації REST API, управління сесіями користувачів та інтеграції механізмів безпеки в архітектуру застосунку. Автори підкреслюють ефективність використання ORM-моделі, вбудованих засобів захисту від CSRF та XSS-атак, а також роль модульного підходу в забезпеченні гнучкості програмного забезпечення. Разом із тим робота має переважно прикладний характер і не містить комплексного аналізу проблем багаторівневої автентифікації в системах підтримки командної роботи.

Аналіз наукових джерел показує, що сучасні дослідження охоплюють окремі аспекти проектування інформаційних систем, web-програмування та інформаційної безпеки, однак комплексний підхід до побудови безпечної системи управління командною роботою із використанням багаторівневої автентифікації залишається недостатньо дослідженим. Більшість існуючих робіт або концентрується на функціональних можливостях систем управління проєктами, або розглядає питання безпеки окремо від архітектури корпоративної взаємодії.

Крім того, недостатньо уваги приділяється проблемам інтеграції механізмів багатофакторної автентифікації у web-середовища з великою кількістю одночасних користувачів, а також питанням балансування між рівнем захисту та продуктивністю системи. Це обумовлює необхідність подальших досліджень, спрямованих на проектування web-платформ, у яких механізми безпеки інтегруються в архітектуру системи як невід’ємна складова процесів управління командною взаємодією.

5. Методи досліджень

Методологія дослідження формувалася з урахуванням специфіки проектування web-орієнтованих інформаційних систем, які одночасно повинні забезпечувати підтримку командної взаємодії та високий рівень захисту інформаційних ресурсів. На відміну від традиційних web-застосунків, системи управління командною роботою функціонують у режимі постійної багатокористувацької взаємодії, що створює додаткові вимоги до архітектури безпеки, механізмів автентифікації та контролю доступу. Саме тому в межах дослідження використовувався комплекс взаємодоповнювальних методів, орієнтованих не лише на реалізацію функціональності системи, але й на аналіз її поведінки в умовах реального навантаження та потенційних кіберзагроз.

Перший етап дослідження передбачав застосування методів системного аналізу для вивчення предметної області та визначення основних процесів, характерних для платформ командної взаємодії. У межах цього етапу аналізувалися особливості організації Agile-процесів, типові сценарії роботи користувачів, структура корпоративної взаємодії та ризики, пов’язані з несанкціонованим доступом до інформаційних ресурсів. Системний підхід дозволив розглядати web-систему як єдине інформаційне середовище, у якому механізми автентифікації, обробки запитів, управління ролями користувачів і журналювання подій функціонують у тісному взаємозв’язку.

Для формалізації функціональної структури системи використовувалася методологія IDEF0. Її застосування дало можливість описати основні бізнес-процеси та встановити логіку взаємодії між компонентами програмного середовища. На контекстному рівні IDEF0-моделі головною функцією системи визначено забезпечення безпечної командної взаємодії в

багатокористувацькому середовищі. Вхідними параметрами моделі виступають облікові дані користувачів, HTTP-запити до серверної частини та операції доступу до ресурсів системи. До керуючих впливів віднесено політики безпеки, правила автентифікації, обмеження доступу та внутрішні механізми контролю прав користувачів. На рисунку 1 продемонстровано контекстна діаграма (A-0).

Рис 1. IDEF0 – Контекстна діаграма (A-0).

Механізмами реалізації функціональних процесів виступають серверна платформа на базі Django, REST API, модулі автентифікації та система керування базою даних PostgreSQL. Результатом роботи системи є надання користувачам контрольованого доступу до функціональних модулів, підтримка командної взаємодії та забезпечення цілісності інформаційного середовища.

На етапі декомпозиції IDEF0-моделі, що представлено на рисунку 2, окремо виділялися процеси реєстрації користувачів, багаторівневої автентифікації, перевірки прав доступу, обробки REST-запитів, моніторингу активності та журналювання подій. Такий підхід дозволив виявити критичні точки взаємодії між компонентами системи, у яких існує найбільша ймовірність виникнення вразливостей або конфігураційних помилок. Особливу увагу приділено процесам, пов’язаним із перевіркою автентичності користувачів і передачею токенів доступу між клієнтською та серверною частинами застосунку.

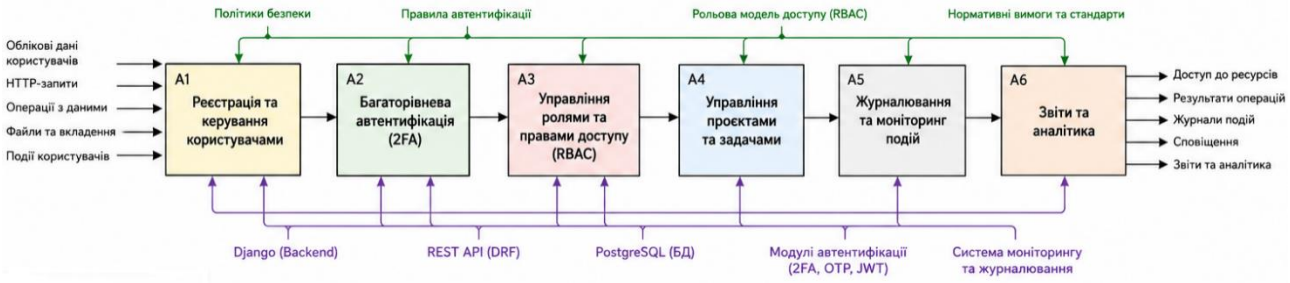

Рис 2. IDEF0 – Декомпозиція діаграми (рівень 1).

Для моделювання поведінки системи в динаміці використовувалися UML sequence diagrams. На відміну від статичних моделей, діаграми послідовностей дозволили відобразити логіку взаємодії між користувачем, сервером автентифікації, базою даних і REST API в часовому аспекті. У межах дослідження sequence diagram використовувалися для опису сценаріїв входу до системи, підтвердження другого фактора автентифікації, перевірки JWT-токенів та доступу до функціональних модулів відповідно до ролі користувача. На рисунку 3 проілюстровано процес багаторівневої автентифікації та доступу до системи (sequence diagram).

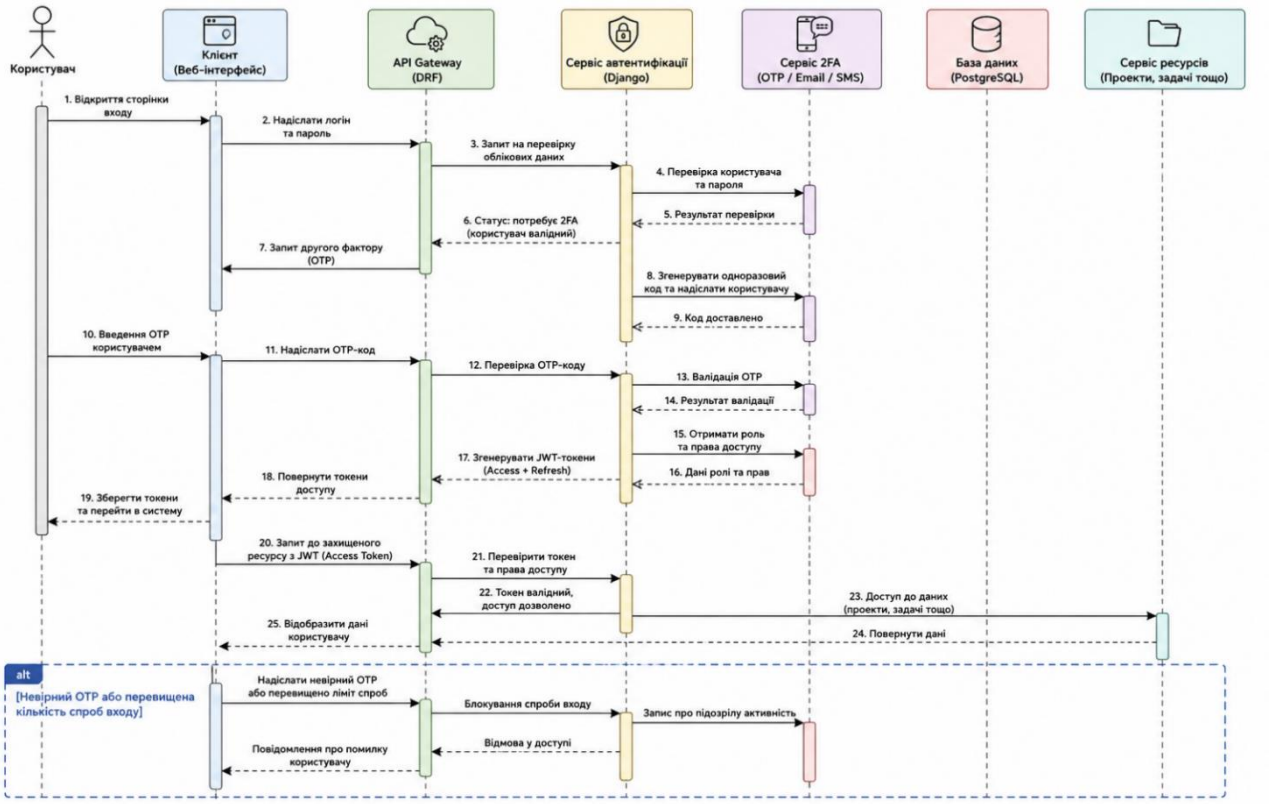

Рис 3. Sequence diagram.

Процес автентифікації моделювався як багатокрокова взаємодія між клієнтським інтерфейсом і серверною частиною системи. Після введення користувачем логіна та пароля серверний модуль виконує перевірку облікових даних і генерує одноразовий код підтвердження. Далі код передається користувачу через додатковий канал автентифікації, після чого виконується перевірка другого фактора. Лише після успішного завершення всіх етапів система формує JWT-токен доступу, який використовується для подальшої взаємодії з REST API. Окремо моделювалися сценарії помилкового введення даних, перевищення допустимої кількості спроб входу та автоматичного блокування доступу при виявленні підозрілої активності.

У процесі проектування програмного забезпечення застосовувався об’єктно-орієнтований підхід, який дозволив забезпечити модульність архітектури та спростити подальше масштабування системи. Основою серверної частини став Django, що підтримує принцип розділення логіки застосунку на окремі компоненти. Використання ORM-механізмів забезпечило централізовану взаємодію з базою даних та зменшило ризики помилок під час формування SQL-запитів.

Для організації взаємодії між клієнтською та серверною частинами використовувався Django REST Framework. REST-підхід дозволив забезпечити стандартизований обмін даними, підтримку асинхронної обробки запитів і можливість інтеграції системи з іншими корпоративними сервісами. Окрему увагу приділено механізмам JWT-автентифікації, які забезпечують безпечну передачу маркерів доступу без необхідності постійного збереження сесій на сервері.

Для оцінювання ефективності системи використовувалися методи сценарного та навантажувального тестування. Сценарне тестування передбачало моделювання типових ситуацій взаємодії користувачів із системою: створення задач, зміна статусів проєктів, авторизація користувачів із різними рівнями доступу, повторні спроби входу та робота з REST API. Такий підхід дозволив оцінити коректність функціонування механізмів role-based access control і перевірити стійкість системи до помилкових або несанкціонованих дій користувачів.

Навантажувальне тестування проводилося шляхом генерації великої кількості одночасних HTTP-запитів до серверної частини системи. Аналізувалися показники часу відповіді сервера, стабільність роботи API, рівень використання пам’яті та процесорних ресурсів. Окремо оцінювалася поведінка системи при активному використанні механізмів журналювання та перевірки автентифікаційних подій, оскільки саме ці процеси суттєво впливають на продуктивність web-застосунків у багатокористувацькому середовищі.

Важливою складовою дослідження стало тестування безпеки системи. Для цього використовувалися елементи penetration testing та методи аналізу типових web-вразливостей. Перевірялася стійкість системи до CSRF, XSS та SQL Injection атак, а також коректність роботи механізмів перевірки JWT-токенів. Додатково оцінювалася ефективність журналювання подій і можливість виявлення підозрілої активності користувачів у режимі реального часу.

У ролі дослідницької вибірки використовувалися типові сценарії роботи IT-команди в умовах багатокористувацької взаємодії. Для моделювання поведінки користувачів були сформовані тестові групи з різними рівнями доступу: адміністратори, менеджери проєктів і учасники команди. Це дозволило дослідити особливості функціонування системи в умовах різних ролей та різного рівня привілеїв доступу.

Надійність отриманих результатів забезпечувалася повторюваністю тестових сценаріїв, використанням стандартизованих методів перевірки web-безпеки та поєднанням функціонального, навантажувального й безпекового тестування. Комплексне застосування IDEF0-моделювання, UML sequence diagram, REST-архітектури та методів аналізу безпеки дозволило не лише спроєктувати web-систему з багаторівневою автентифікацією, а й оцінити її поведінку в умовах реального використання.

Таким чином, використана методологія дослідження забезпечила можливість комплексного аналізу функціональних і захисних характеристик системи управління командною роботою та створила основу для подальшого вдосконалення механізмів автентифікації, масштабування та моніторингу інформаційної безпеки в корпоративних web-середовищах.

6. Результати досліджень

У межах проведеного дослідження було реалізовано прототип безпечної web-системи управління командною роботою, орієнтований на використання в середовищі IT-команд із підвищеними вимогами до контролю доступу та захисту корпоративних даних. Основна увага під час проектування приділялася не лише функціональності системи, але й інтеграції механізмів інформаційної безпеки в базову архітектуру застосунку. Такий підхід дозволив розглядати автентифікацію, управління ролями користувачів і моніторинг активності не як окремі допоміжні модулі, а як невід’ємну складову логіки функціонування системи.

Практична реалізація програмного рішення виконувалася із використанням Django та Django REST Framework. Серверна частина системи забезпечує обробку REST-запитів, управління механізмами автентифікації та централізовану взаємодію з базою даних. Як система зберігання інформації використовувалася PostgreSQL, що дозволило реалізувати структуроване збереження даних про користувачів, задачі, ролі доступу та журнали активності.

У результаті проектування сформовано багаторівневу архітектуру, яка включає такі функціональні компоненти: клієнтський web-інтерфейс; API Gateway для обробки REST-запитів; модуль автентифікації користувачів; систему role-based access control; модуль журналювання подій; підсистему моніторингу активності; централізовану базу даних.

Розроблена архітектура забезпечила розділення функціональних компонентів системи, що спростило адміністрування та підвищило масштабованість програмного рішення. У

практичному аспекті це дозволяє модернізувати окремі модулі без необхідності суттєвого втручання в інші компоненти системи.

Одним із ключових результатів дослідження стала реалізація механізму багаторівневої автентифікації. На відміну від традиційних web-застосунків, де перевірка користувача обмежується введенням логіна та пароля, у запропонованій системі реалізовано додатковий рівень підтвердження особи через OTP-код. Практичне тестування показало, що такий підхід суттєво знижує ризик несанкціонованого доступу до корпоративних ресурсів навіть у випадку компрометації первинних облікових даних.

Послідовність взаємодії між компонентами системи відображено у sequence diagram, де показано процес передавання запитів між клієнтським інтерфейсом, сервером автентифікації, модулем OTP та REST API. Побудована діаграма дозволила формалізувати логіку багатокрокової перевірки користувача та визначити критичні точки взаємодії між компонентами системи. Особливу увагу було приділено сценаріям помилкової автентифікації, перевищення кількості спроб входу та автоматичного блокування підозрілої активності.

У процесі реалізації системи також було впроваджено механізм role-based access control. Для різних категорій користувачів сформовано окремі політики доступу до інформаційних ресурсів. Зокрема, адміністратор системи отримує доступ до управління користувачами та журналами безпеки, менеджер проєкту – до модулів координації задач і командної взаємодії, а учасник команди – лише до функцій, необхідних для виконання власних робочих операцій.

Результати функціонального тестування підтвердили коректність реалізованої моделі доступу. Система успішно блокувала спроби доступу до ресурсів, які не відповідали рівню привілеїв користувача, а всі події порушення політик доступу автоматично фіксувалися у журналі активності. Це дозволило підвищити контроль за діями користувачів і спростити процес аудиту безпеки.

Для оцінювання ефективності запропонованої архітектури було проведено навантажувальне тестування системи. Метою тестування стало визначення стабільності роботи web-застосунку в умовах одночасної взаємодії великої кількості користувачів із REST API. Результати навантажувального тестування наведено в таблиці 1.

Таблиця 1. Результати навантажувального тестування

Кількість активних користувачів	Середній час відповіді	Успішність виконання запитів	Використання CPU
50	118 мс	99.8 %	32 %
100	176 мс	99.4 %	46 %
250	305 мс	98.9 %	67 %
500	552 мс	97.3 %	82 %

Аналіз результатів показав, що навіть при суттєвому навантаженні система зберігає стабільність роботи та забезпечує високий рівень успішної обробки запитів. Зростання часу відповіді при збільшенні кількості активних користувачів є прогнозованим, однак отримані показники залишаються прийнятними для корпоративного середовища командної взаємодії.

Окремий етап дослідження був присвячений тестуванню механізмів інформаційної безпеки. У процесі penetration testing перевірялася стійкість системи до найбільш поширених web-загроз: CSRF, XSS та SQL Injection атак. Використання вбудованих засобів захисту Django у поєднанні з JWT-автентифікацією продемонструвало достатній рівень захищеності серверної частини застосунку. В таблиці 2 відображено результати тестування механізмів багаторівневої автентифікації.

Таблиця 2. Результати тестування механізмів багаторівневої автентифікації

Тип сценарію автентифікації	Кількість тестових спроб	Успішна автентифікація	Заблоковані спроби	Середній час автентифікації
Коректний логін + OTP	200	198	2	2.1 с
Невірний пароль	150	0	150	1.3 с
Невірний OTP-код	120	0	120	2.4 с
Повторне використання OTP	80	0	80	1.9 с
Перевищення ліміту спроб входу	60	0	60	1.7 с
Авторизація користувача з роллю “Менеджер”	100	100	0	2.0 с
Авторизація користувача з роллю “Адміністратор”	70	70	0	2.2 с

Результати, наведені у таблиці 2, демонструють стабільність роботи механізмів багаторівневої автентифікації в умовах різних сценаріїв використання системи. Під час експериментального тестування коректна автентифікація користувачів забезпечувалася у більшості випадків, а всі спроби входу з неправильними параметрами були успішно заблоковані системою. Особливу увагу привертають результати перевірки OTP-механізму. Повторне використання одноразових кодів або введення некоректного OTP призводило до автоматичної відмови в доступі без порушення стабільності роботи серверної частини. Це підтверджує ефективність реалізованого підходу до захисту процесу автентифікації. Крім того, результати експерименту свідчать про коректне функціонування механізму автоматичного блокування після перевищення допустимої кількості спроб входу. Такий підхід дозволяє знизити ефективність brute-force атак і підвищує загальний рівень захищеності інформаційного середовища. Отримані показники середнього часу автентифікації залишаються прийнятними для корпоративних web-систем командної взаємодії та демонструють, що інтеграція другого фактора автентифікації не створює критичного впливу на швидкодію системи.

Практичні результати тестування також підтвердили ефективність механізмів блокування brute-force атак. Після перевищення допустимої кількості невдалих спроб входу система автоматично обмежувала доступ користувача та створювала запис у журналі безпеки. Це дозволило не лише мінімізувати ризики несанкціонованого доступу, але й забезпечити можливість подальшого аналізу підозрілої активності.

Суттєвим результатом дослідження стало впровадження централізованого журналювання подій. Система автоматично фіксує спроби входу, зміни ролей користувачів, помилки автентифікації та звернення до захищених ресурсів. На практиці це створює основу для побудови механізмів аудиту безпеки та спрощує процес реагування на інциденти.

Отримані результати свідчать про те, що інтеграція механізмів багаторівневої автентифікації безпосередньо в архітектуру web-застосунку є більш ефективним підходом порівняно з використанням зовнішніх модулів безпеки. У межах проведеного дослідження це дозволило знизити складність адміністрування системи, покращити контроль доступу та забезпечити централізоване управління процесами автентифікації.

Практичне значення роботи полягає в можливості використання запропонованої архітектури як основи для створення корпоративних платформ командної взаємодії в організаціях із підвищеними вимогами до інформаційної безпеки. Результати дослідження можуть бути адаптовані для IT-компаній, фінансових структур, освітніх платформ та інших середовищ, де критичне значення має захист корпоративних даних і контроль дій користувачів.

Таким чином, проведене дослідження підтвердило ефективність використання Django для проектування безпечних web-систем управління командною роботою. Реалізований підхід

забезпечує поєднання масштабованості, функціональності та високого рівня захисту інформаційного середовища, що створює передумови для подальшого практичного впровадження розробленого програмного рішення.

7. Перспективи подальшого розвитку досліджень

Подальший розвиток досліджень у сфері безпечних систем управління командною роботою визначається не лише технологічними змінами у web-розробці, але й трансформацією самої моделі корпоративної взаємодії. Поширення distributed-команд, віддалених форматів співпраці та постійне зростання обсягів корпоративних даних формують нові вимоги до архітектури інформаційних систем. У таких умовах платформи командної взаємодії перестають бути виключно інструментом координації задач і поступово перетворюються на централізоване середовище управління інформаційними потоками організації. Саме тому питання інтеграції механізмів безпеки безпосередньо в архітектуру web-застосунку набуває стратегічного значення.

Проведене дослідження підтвердило, що використання Django як основи для проектування захищених корпоративних платформ є доцільним з точки зору подальшого масштабування системи та адаптації її до змінних умов експлуатації. Важливою перевагою запропонованого підходу є модульність архітектури, яка дозволяє інтегрувати нові функціональні компоненти без суттєвого порушення логіки роботи вже реалізованих підсистем. У практичному аспекті це створює можливість поетапного розвитку системи залежно від потреб організації та рівня складності інформаційного середовища.

Одним із найбільш перспективних напрямів подальших досліджень є розвиток адаптивних механізмів автентифікації. У більшості сучасних корпоративних систем багатофакторна автентифікація реалізується за статичним принципом, коли однакова процедура перевірки застосовується до всіх користувачів незалежно від контексту доступу. Проте в умовах динамічного цифрового середовища такий підхід не завжди є ефективним. Перспективною є реалізація adaptive authentication, де система аналізує поведінкові характеристики користувача, тип пристрою, місце входу, часові параметри та історію попередньої активності. У разі виявлення нетипових ознак доступу рівень перевірки може автоматично посилюватися.

Інтеграція подібних механізмів є особливо актуальною для систем командної взаємодії, оскільки саме вони функціонують у середовищах із великою кількістю одночасних користувачів та значною інтенсивністю інформаційного обміну. Використання поведінкової аналітики дозволить не лише підвищити рівень безпеки, але й знизити навантаження на користувача в типових сценаріях роботи за рахунок адаптивного управління процедурами автентифікації.

Ще одним перспективним напрямом є інтеграція принципів Zero Trust Security у структуру web-систем командної роботи. Традиційна модель інформаційної безпеки базується на понятті довіреного внутрішнього середовища, однак сучасні умови експлуатації корпоративних платформ фактично нівелюють межі між внутрішньою та зовнішньою інфраструктурою. У випадку використання віддаленого доступу або хмарних сервісів кожен запит до інформаційних ресурсів повинен розглядатися як потенційно небезпечний незалежно від статусу користувача.

Практичне впровадження концепції Zero Trust у системи управління командною роботою відкриває можливість реалізації динамічних політик доступу, які враховуватимуть не лише роль користувача, але й поточний контекст його дій. Наприклад, доступ до критичних ресурсів може автоматично обмежуватися при зміні геолокації, нетиповій поведінці або різкому зростанні кількості запитів до API. Подібний підхід значно підвищує стійкість системи до внутрішніх та зовнішніх загроз.

Окрему перспективу становить використання технологій машинного навчання для моніторингу активності користувачів та автоматичного виявлення аномалій. У процесі функціонування корпоративної web-системи накопичується значний масив даних щодо поведінки користувачів, структури їх взаємодії з інформаційними ресурсами та характеру звернень до серверної частини. Аналіз таких даних дозволяє формувати поведінкові моделі, які можуть використовуватися для раннього виявлення потенційних кіберінцидентів.

На відміну від класичних систем моніторингу, які переважно реагують на вже відомі шаблони атак, поведінкова аналітика дозволяє виявляти нетипові дії навіть за відсутності конкретних сигнатур загроз. Це особливо важливо для корпоративних платформ командної взаємодії, де значна частина ризиків пов'язана саме з компрометацією облікових записів або внутрішніми помилками користувачів.

Подальші дослідження також можуть бути спрямовані на вдосконалення продуктивності системи в умовах масштабного навантаження. Хоча результати тестування продемонстрували стабільність роботи реалізованого прототипу, зростання кількості користувачів та інтеграція додаткових механізмів моніторингу можуть суттєво впливати на швидкодію серверної частини. У цьому контексті перспективними є дослідження, пов'язані з використанням мікросервісної архітектури, контейнеризації та розподіленої обробки запитів.

Важливим напрямом розвитку залишається інтеграція системи з корпоративними сервісами зовнішньої автентифікації. Підтримка OAuth 2.0, OpenID Connect або SAML дозволить реалізувати механізми єдиного входу та централізованого управління обліковими записами користувачів. Для великих організацій це є не лише технічною перевагою, але й важливим фактором зниження адміністративних витрат.

Практична цінність подальшого розвитку запропонованого підходу полягає в можливості створення універсальної корпоративної платформи, здатної поєднувати підтримку Agile-процесів, гнучке управління доступом та інтелектуальні механізми захисту інформаційного середовища. З урахуванням тенденцій розвитку distributed-команд та ускладнення сучасних кіберзагроз подібні системи можуть стати основою цифрової інфраструктури організацій, для яких критично важливими є безпечна взаємодія та централізований контроль інформаційних ресурсів.

8. Висновки

Проведене дослідження було спрямоване на вирішення актуальної проблеми забезпечення безпечної командної взаємодії в сучасних web-орієнтованих інформаційних середовищах. Зростання кількості distributed-команд, використання хмарних сервісів та активне впровадження Agile-підходів суттєво змінили характер корпоративної взаємодії, однак одночасно збільшили кількість ризиків, пов'язаних із захистом інформаційних ресурсів. У цих умовах системи управління командною роботою перестають виконувати виключно координаційну функцію та фактично стають ядром корпоративної цифрової інфраструктури. Саме тому питання інтеграції механізмів інформаційної безпеки в архітектуру подібних платформ набуває не лише технічного, а й організаційного значення.

У межах дослідження було встановлено, що значна частина існуючих платформ командної взаємодії реалізує механізми безпеки фрагментарно, орієнтуючись переважно на базову автентифікацію користувачів та стандартні моделі доступу. Такий підхід є недостатнім для корпоративних середовищ, де обробляються критичні дані та підтримується постійна взаємодія великої кількості користувачів. Аналіз практики використання сучасних web-систем підтвердив, що компрометація облікових записів, недостатній контроль прав доступу та відсутність централізованого моніторингу залишаються одними з найбільш поширених причин виникнення інцидентів інформаційної безпеки.

У процесі роботи було спроєктовано та реалізовано прототип безпечної системи управління командною роботою на базі Django. Використання цієї платформи дозволило

сформувати модульну структуру програмного забезпечення, у межах якої механізми автентифікації, управління ролями користувачів, журналювання подій та REST-взаємодії інтегруються в єдину архітектуру. Практичне значення такого підходу полягає в тому, що система безпеки перестає бути зовнішнім компонентом і функціонує як частина базової логіки web-застосунку.

Одним із ключових результатів дослідження стало впровадження багаторівневої автентифікації користувачів. Проведене тестування показало, що використання OTP-підтвердження у поєднанні з JWT-механізмами суттєво підвищує стійкість системи до несанкціонованого доступу та автоматизованих атак. Водночас отримані результати свідчать, що інтеграція додаткових рівнів перевірки не створює критичного впливу на швидкодію системи та не призводить до суттєвого погіршення користувацького досвіду. Це є важливим практичним аспектом, оскільки корпоративні інформаційні системи повинні забезпечувати баланс між захищеністю та ефективністю взаємодії користувачів.

Під час навантажувального тестування було підтверджено стабільність роботи серверної частини системи в умовах значної кількості одночасних запитів. Отримані показники часу відповіді та рівня успішної обробки REST-запитів свідчать про доцільність використання REST-архітектури для побудови корпоративних платформ командної взаємодії. Водночас результати експериментів дозволили виявити залежність продуктивності системи від інтенсивності процедур журналювання та перевірки автентифікаційних подій. Це демонструє, що питання оптимізації безпекових механізмів залишатиметься актуальним у процесі подальшого масштабування системи.

Важливим результатом дослідження стало також впровадження role-based access control, що дозволило забезпечити контрольований доступ до функціональних модулів відповідно до ролей користувачів. Практична перевірка показала ефективність такого підходу для корпоративних середовищ, де різні категорії користувачів працюють із неоднаковим рівнем доступу до інформаційних ресурсів. Крім того, реалізація централізованого журналювання подій створила основу для подальшого розвитку механізмів аудиту безпеки та моніторингу активності користувачів.

Разом із позитивними результатами дослідження було виявлено низку проблемних аспектів, які потребують додаткового аналізу. Насамперед це стосується зростання навантаження на серверну інфраструктуру при використанні складних процедур автентифікації та постійного контролю активності користувачів. У реальних умовах експлуатації корпоративних систем такі процеси можуть впливати на швидкодію застосунку, особливо в середовищах із великою кількістю одночасних користувачів або значним обсягом API-взаємодії.

Не менш важливим є питання адаптації систем безпеки до динамічного характеру сучасних кіберзагроз. Класичні моделі автентифікації та статичні політики доступу поступово втрачають ефективність через зростання складності атак і використання компрометованих облікових записів. Саме тому подальший розвиток дослідження доцільно пов’язувати із впровадженням adaptive authentication, поведінкової аналітики та принципів Zero Trust Security. Подібні механізми дозволять перейти від реактивної моделі захисту до системи постійного аналізу ризиків і динамічного управління доступом.

Практична цінність отриманих результатів полягає в можливості використання запропонованої архітектури як основи для створення корпоративних платформ у середовищах, де критичне значення має безпечна взаємодія користувачів та централізований контроль інформаційних ресурсів. Результати дослідження можуть бути адаптовані для IT-компаній, фінансових структур, освітніх платформ і організацій із distributed-командами.

Таким чином, проведене дослідження підтвердило ефективність інтеграції багаторівневої автентифікації та механізмів контролю доступу в архітектуру web-систем управління командною роботою. Запропонований підхід демонструє можливість поєднання функціональності, масштабованості та сучасних механізмів інформаційної безпеки в межах

єдиного програмного середовища, що створює основу для подальшого розвитку корпоративних web-платформ нового покоління.

Список літератури:

1) Woldman, T.: Hands-on microservices with Django: build cloud-native and reactive applications with Python using Django 5. Packt Publishing, Place of publication not identified (2024).

2) Melé, A., Melchiorre, P.: Django 5 by example: build powerful and reliable Python web applications from Scratch: updated to Django 5.2 LTS. Packt, Birmingham (2025).

3) Enhanced Reliability In Iot With Sdn By Multifactor Authentication Approach. (2024). Nanotechnology Perceptions, 20(S14). https://doi.org/10.62441/nano-ntp.v20is14.158.

4) Klivan, S., Höltervennhoff, S., Huaman, N., Krause, A., Simko, L., Acar, Y., & Fahl, S. (2023). “We’ve Disabled MFA for You”: An Evaluation of the Security and Usability of Multi-Factor Authentication Recovery Deployments. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (pp. 3138–3152). ACM. CCS ’23: ACM SIGSAC Conference on Computer and Communications Security. https://doi.org/10.1145/3576915.3623180.

5) Ling, X., Wu, L., Zhang, J., Qu, Z., Deng, W., Chen, X., Qian, Y., Wu, C., Ji, S., Luo, T., Wu, J., & Wu, Y. (2023). Adversarial attacks against Windows PE malware detection: A survey of the state-of-the-art. Computers & Security, 128, 103134. https://doi.org/10.1016/j.cose.2023.103134

6) Cerny, T., Walker, A., Svacina, J., Bushong, V., Das, D., Frajtak, K., Bures, M., Tisnovsky, P.: Mapping Study on Constraint Consistency Checking in Distributed Enterprise Systems. In: Proceedings of the International Conference on Research in Adaptive and Convergent Systems. pp. 167–174. ACM, Gwangju Republic of Korea (2020). https://doi.org/10.1145/3400286.3418257.

7) Ghaffari, F., Bertin, E., Crespi, N., Hatin, J.: Distributed ledger technologies for authentication and access control in networking applications: A comprehensive survey. Computer Science Review. 50, 100590 (2023). https://doi.org/10.1016/j.cosrev.2023.100590.

8) Adesokan, A., Kinney, R., & Tsiropoulou, E. E. (2024). CROWDMATCH: Optimizing Crowdsourcing Matching through the Integration of Matching Theory and Coalition Games. Future Internet, 16(2), 58. https://doi.org/10.3390/fi16020058.

9) Otta, S. P., Panda, S., Gupta, M., & Hota, C. (2023). A Systematic Survey of Multi-Factor Authentication for Cloud Infrastructure. Future Internet, 15(4), 146. https://doi.org/10.3390/fi15040146.

10) Kruzikova, A., Muzik, M., Knapova, L., Dedkova, L., Smahel, D., Matyas, V.: Two-factor authentication time: How time-efficiency and time-satisfaction are associated with perceived security and satisfaction. Computers & Security. 138, 103667 (2024). https://doi.org/10.1016/j.cose.2023.103667.

11) Rahaman, M. S., Tisha, S. N., Song, E., & Cerny, T. (2023). Access Control Design Practice and Solutions in Cloud-Native Architecture: A Systematic Mapping Study. Sensors, 23(7), 3413. https://doi.org/10.3390/s23073413

12) Venčkauskas, A., Kukta, D., Grigaliūnas, Š., & Brūzgienė, R. (2023). Enhancing Microservices Security with Token-Based Access Control Method. Sensors, 23(6), 3363. https://doi.org/10.3390/s23063363

13) Samuel, B., & Kasturi, K. (2024). A secure authentication and collaborative data-sharing model based on a blockchain network in the cloud. Journal of Control and Decision, 11(4), 730–745. https://doi.org/10.1080/23307706.2023.2293965

14) Petcu, A., Pahontu, B., Frunzete, M., & Stoichescu, D. A. (2023). A Secure and Decentralized Authentication Mechanism Based on Web 3.0 and Ethereum Blockchain Technology. Applied Sciences, 13(4), 2231. https://doi.org/10.3390/app13042231

15) Mostafa, A. M., Ezz, M., Elbashir, M. K., Alruily, M., Hamouda, E., Alsarhani, M., & Said, W. (2023). Strengthening Cloud Security: An Innovative Multi-Factor Multi-Layer Authentication Framework for Cloud User Authentication. Applied Sciences, 13(19), 10871. https://doi.org/10.3390/app131910871

16) Zhu, Y., Wang, J., Li, B., Zhao, Y., Zhang, Z., Xiong, Y., & Chen, S. (2024). MicroIRC: Instance-level Root Cause Localization for Microservice Systems. Journal of Systems and Software, 216, 112145. https://doi.org/10.1016/j.jss.2024.112145

17) Onile, A. E., Petlenkov, E., Levron, Y., & Belikov, J. (2024). Smartgrid-based hybrid digital twins framework for demand side recommendation service provision in distributed power systems. Future Generation Computer Systems, 156, 142–156. https://doi.org/10.1016/j.future.2024.03.018

18) Vasudhar Sai Thokala. (2023). Enhancing Test-Driven Development (TDD) and BDD Methodologies in Full-Stack Web Applications. International Journal of Science and Research Archive, 10(1), 1119–1129. https://doi.org/10.30574/ijsra.2023.10.1.0815

19) Hashmi, I. F., Iqbal, Z., Munir, E., Kryvinska, N., Ivanochko, I., & Sampedro, G. A. (2024). SAAC: Secure Access Control Management Framework for Multi-User Smart Home Systems. IEEE Access, 12, 133339–133355. https://doi.org/10.1109/access.2024.3446180

20) Siddiqui, F., Khan, R., Sezer, S., McLaughlin, K., Masing, L., Dorr, T., Schade, F., Becker, J., Ahlbrecht, A., Zaeske, W., Durak, U., Adler, N., Sailer, A., Weber, R., Wilhelm, T., Nemeth, G., Morales, V., Gomez, P., Keramidas, G., Antonopoulos, C.P., Mavropoulos, M., Kelefouras, V., Antonopoulos, K., Voros, N., Panagiotou, C., Karadimas, D.: XANDAR: A holistic Cybersecurity Engineering Process for Safety-critical and Cyber-physical Systems. In: 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring). pp. 1–5. IEEE, Helsinki, Finland (2022). https://doi.org/10.1109/VTC2022-Spring54318.2022.9860859.

21) Manoj Kumar, & Dr Rainu Nandal. (2024). Python's Role in Accelerating Web Application Development with Django. International Research Journal on Advanced Engineering and Management (IRJAEM), 2(06), 2092–2105. https://doi.org/10.47392/irjaem.2024.0307

Designing a secure teamwork management system with multi-level authentication based on Django

Yurii Tulashvili
Department of Computer Science, Lutsk National Technical University, Lutsk, Ukraine
ORCID: 0000-0002-0780-9529

Viktor Kosheliuk
Department of Computer Science, Lutsk National Technical University, Lutsk, Ukraine
ORCID: 0000-0002-4136-5087

Bohdan Morozyuk
Department of Computer Science, Lutsk National Technical University, Lutsk, Ukraine
ORCID: 0009-0002-8342-2518

Abstract: The growth of remote teamwork and the active use of web platforms in the field of software development make the problem of ensuring secure access to corporate information resources urgent. In modern team interaction management systems, user authentication mechanisms, data

access control, and protection against widespread cyber threats are of particular importance. Insufficient security on such platforms can lead to account compromises, the loss of confidential information, and the compromise of work process integrity. In this regard, there is a need to create web-oriented systems that combine team management functionality with modern multi-level protection mechanisms. The purpose of the study is to design a secure teamwork management system based on Django, implementing multi-level authentication and access control mechanisms for information resources. The work analyzes modern approaches to building secure web applications, identifies key requirements for the system architecture, and justifies the selection of a technological stack for implementing the server-side of the software. The methodological basis of the study is system analysis, object-oriented design, functional modeling, and web programming technologies. To implement the software solution, Django and Django REST Framework were used, which provide support for a modular architecture, RESTful interactions, and integrated security mechanisms. The system implemented multi-level authentication, a role-based access model, JWT-based authorization mechanisms, and protection against attacks such as CSRF, XSS, and SQL Injection. As a result of the study, an architectural model of a web system focused on supporting team interaction in a multi-user environment was formed. The developed software solution provides centralized user management, access control to functional modules, action logging, and secure data transfer between the client and server parts of the system. The testing showed an increase in the security of access to information resources and in the system's stability when several users work simultaneously. The results obtained confirm the feasibility of using Django to create secure teamwork management systems. Further research could focus on integrating adaptive authentication mechanisms, using behavioral analysis to detect anomalous user activity, and implementing a microservice architecture to improve system scalability and fault tolerance.

Keywords: Django; multi-level authentication; information security; web-based teamwork management system; access control.