## Функціональні вимоги до ПЗ

*Реєстрація та авторизація:* Застосунок повинен мати можливість реєстрації нових користувачів і авторизації вже зареєстрованих користувачів для доступу до функцій перегляду оголошень на інтерактивній мапі, публікації нових оголошень, комунікації між потенційним орендарем та орендодавцем.

*Додавання оголошення:* Застосунок повинен забезпечувати можливість створення оголошення з детальним описом, фото та відео з подальшою опцією просування, редагування та видалення. Усі оголошення проходять через модерацію працівниками компанії.

*Пошук оголошень:* Застосунок повинен мати інтерактивну деталізовану мапу міста на головній сторінці з підсвіченими будинками, у яких є оголошення, що задовільняє введені користувачем параметри пошуку.

*Керування обраними оголошеннями:* Користувач повинен мати можливість додавати оголошення до обраних та, за потреби, видаляти його зі списку.

*Чат між орендарем та орендодавцем:* Застосунок повинен мати можливість створювати діалоги між орендарем та орендодавцем стосовно конкретного оголошення.

## Нефункціональні вимоги до ПЗ

*Інтерфейс:* Застосунок повинен мати інтуїтивно зрозумілий та привабливий інтерфейс, що дозволяє легко керувати оголошеннями та здійснювати їх пошук відповідно до заданих фільтрів.

*Швидкість та продуктивність:* Застосунок повинен працювати швидко та ефективно, забезпечуючи миттєве завантаження оголошень на інтерактивній мапі та іншого функціоналу додатку.

*Сумісність:* Застосунок повинен бути сумісним з усіма актуальними версіями мобільних операційних систем Android та iOS та бути адаптованим під різні діагоналі екрану телефонів та планшетів.

*Безпека та відсутність посередників:* Застосунок повинен надсилати кожне оголошення на модерацію на перевірку коректності опису, зображень, відео та адреси. Орендодавець, а відповідно і його оголошення, може бути верифікованим за допомогою селфі та фотографії паспорту чи водійського посвідчення задля перевірки чи дана особа не є посередником між оголошенням та власниками житла.

## Взаємодія між компонентами ПЗ

Взаємодія між компонентами ПЗ відбувається з використанням API. Клієнтський рівень взаємодіє з серверним, надсилаючи запити на сервер та отримуючи відповіді. Це може бути реалізовано за допомогою RESTful API, який використовує HTTP протокол для передачі даних.

Серверний рівень обробляє запити, виконує необхідні операції та взаємодіє з базою даних для отримання необхідної інформації. Він також може взаємодіяти з іншими зовнішніми сервісами або API, які можуть знадобитися для додаткової функціональності.

Така архітектура дозволяє забезпечити зручну та безпечну взаємодію між користувачем та системою.

## Діаграма варіантів використання

Use-case діаграми використання UML є основною формою опису поведінки системи з точки зору користувача і визначають очікувану поведінку, а не конкретний спосіб реалізації. Вони дозволяють нам проектувати систему з орієнтацією на кінцевого користувача та передавати поведінку системи шляхом визначення зовнішніх взаємодій.

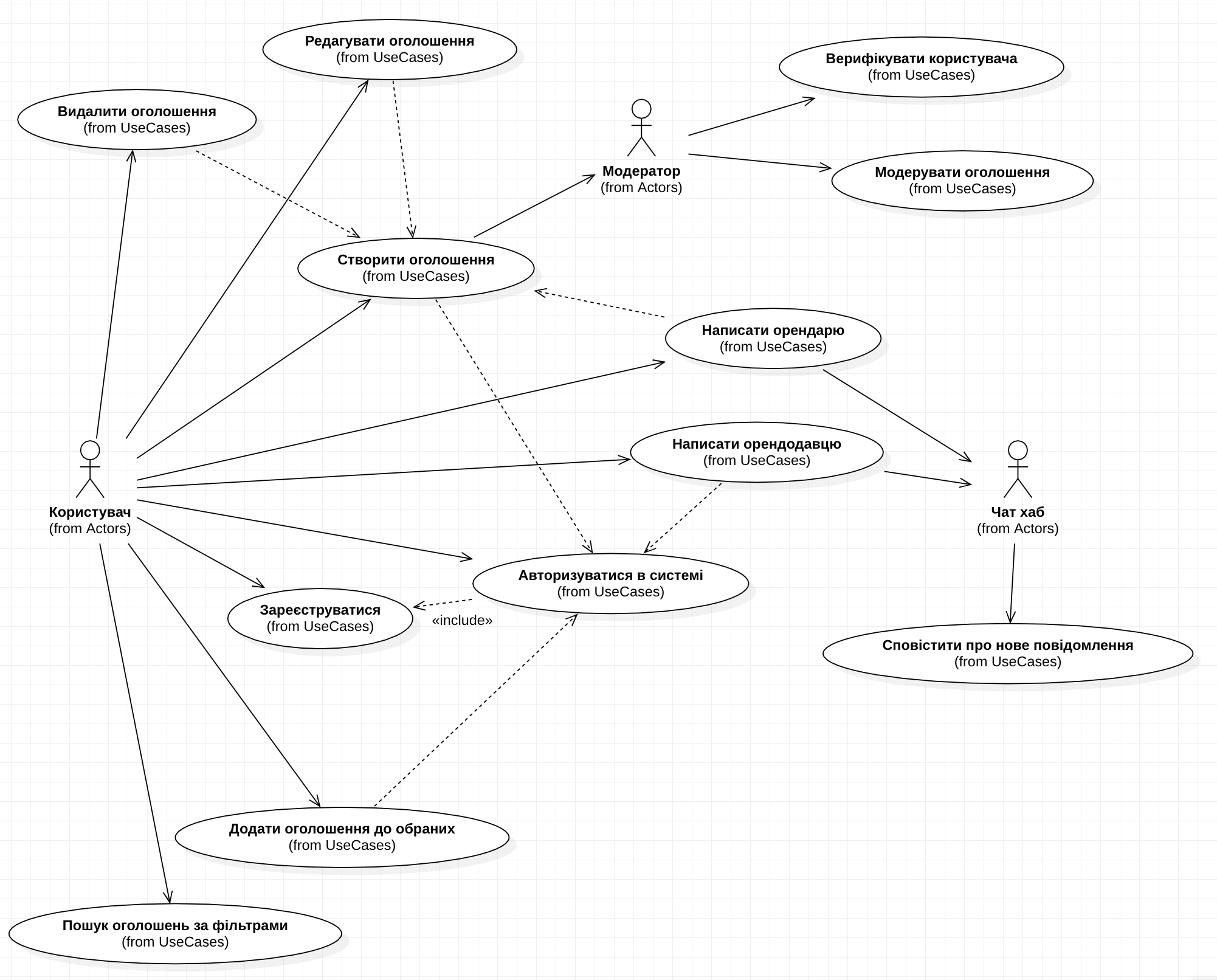


Рис. 3.1 (Use-case діаграма)

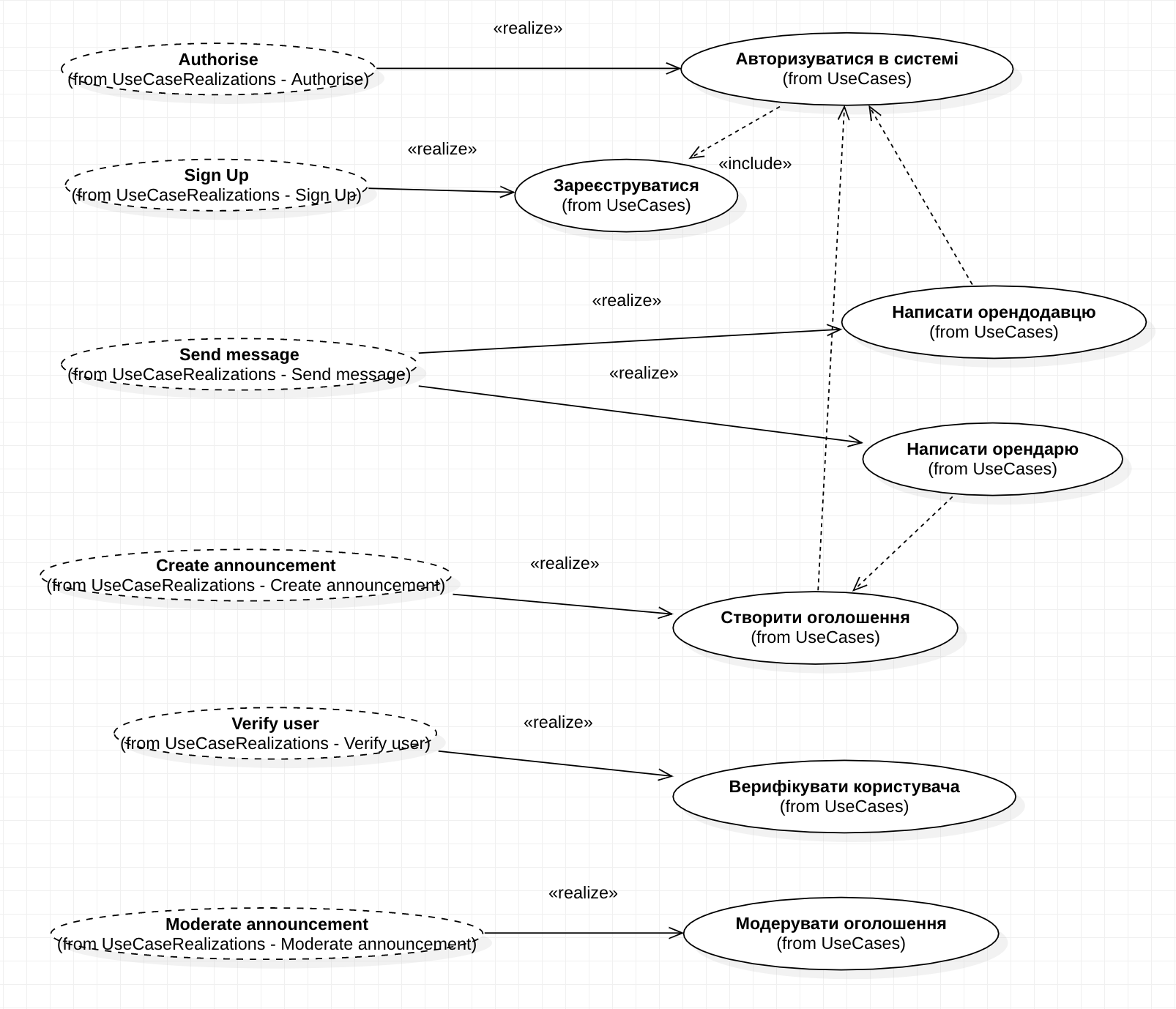


Рис. 3.2 (Діаграма трасування Traceabilities)

## Діаграми послідовностей

Діаграми послідовності в UML є важливими діаграмами для моделювання взаємодії об'єктів і деталей виконання операцій. Вони зосереджені на послідовності взаємодій та передають цю інформацію за допомогою вертикальної осі діаграми, відображаючи моменти надсилання повідомлень.

Діаграми послідовності широко використовуються для уточнення діаграм прецедентів, що дозволяють детально описати поведінкові аспекти системи. Вони передають інформацію у формі повідомлень, а взаємодія об'єктів показується через обмін цими повідомленнями в межах сценарію.

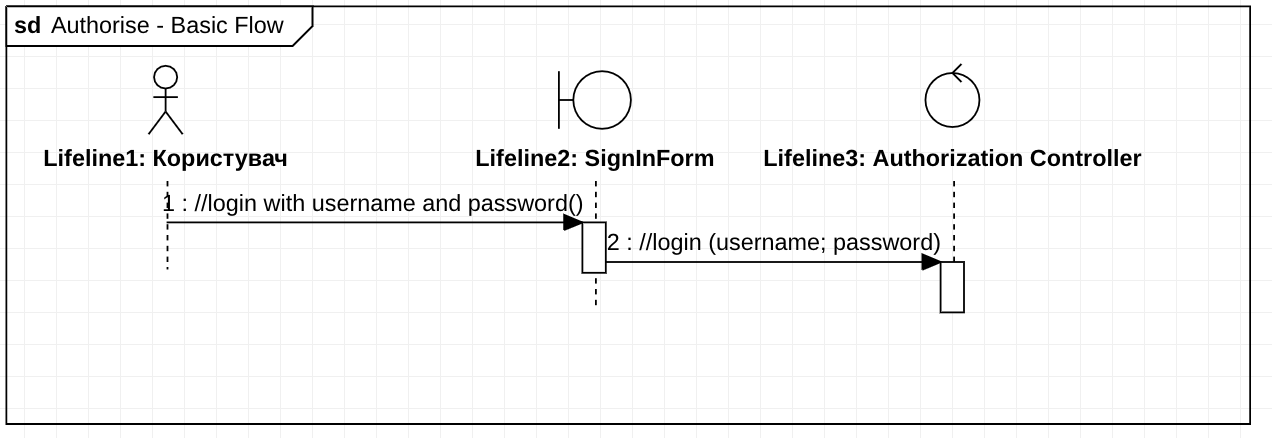


Рис. 3.3 (Діаграма послідовності авторизації)

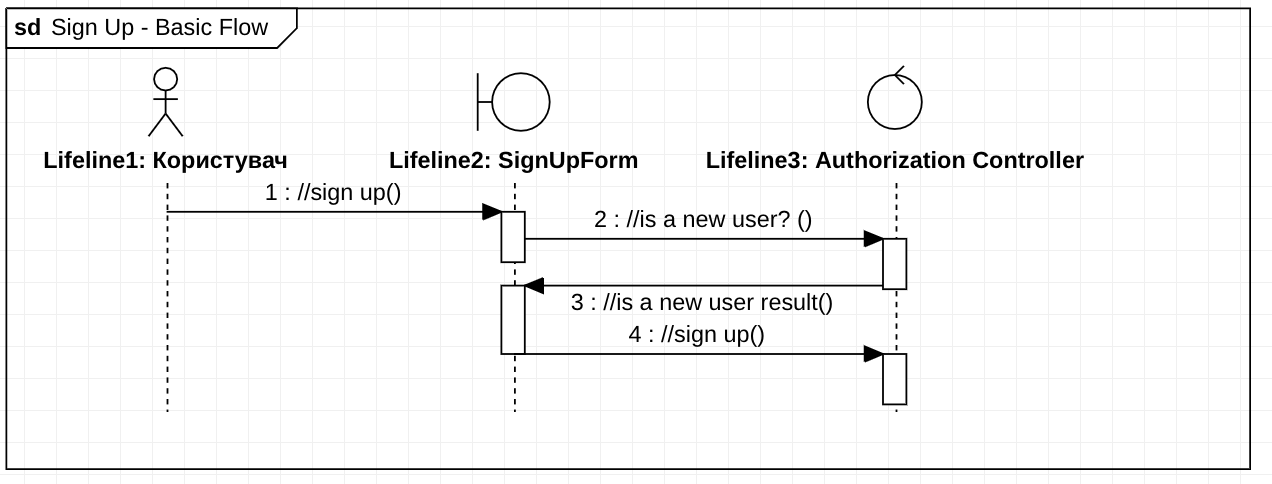


Рис. 3.4 (Діаграма послідовності реєстрації)

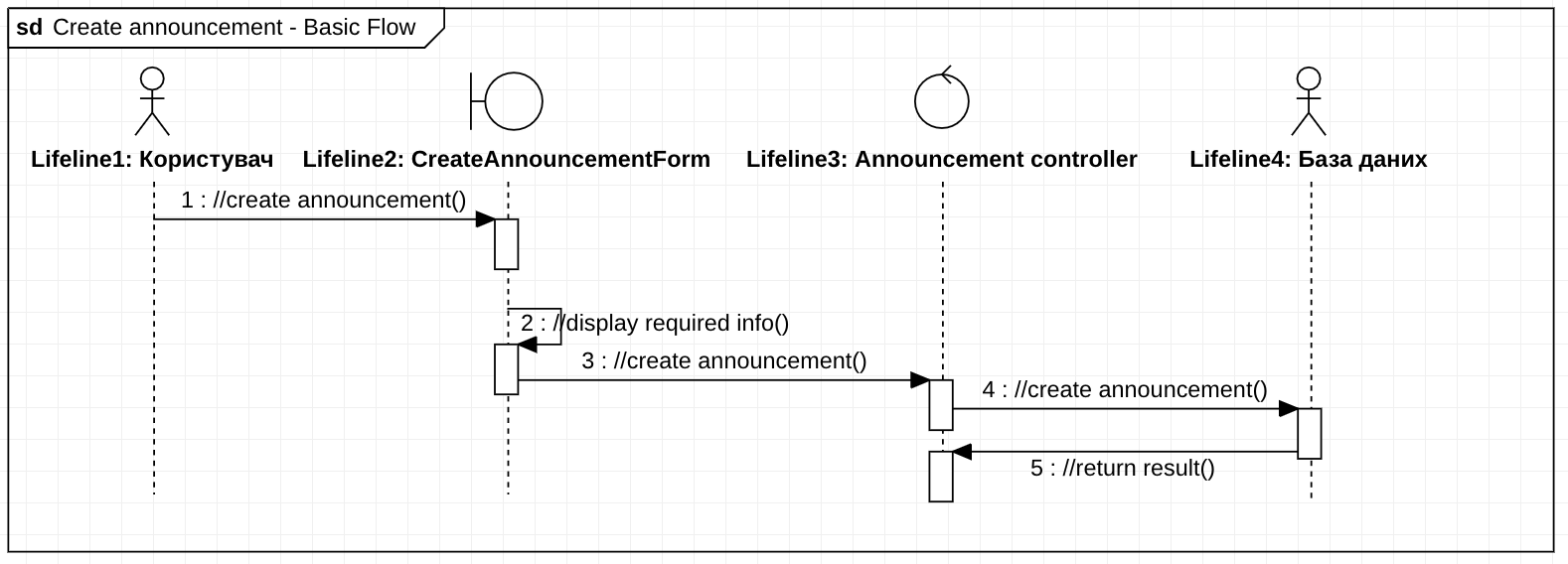


Рис. 3.5 (Діаграма послідовності створення оголошення)

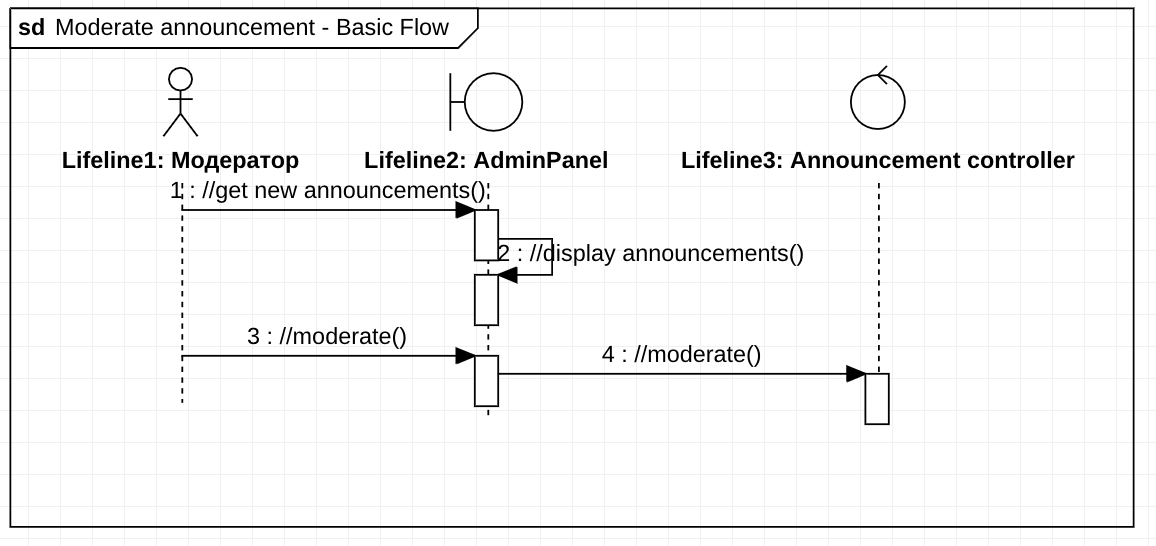


Рис. 3.6 (Діаграма послідовності модерації оголошення)

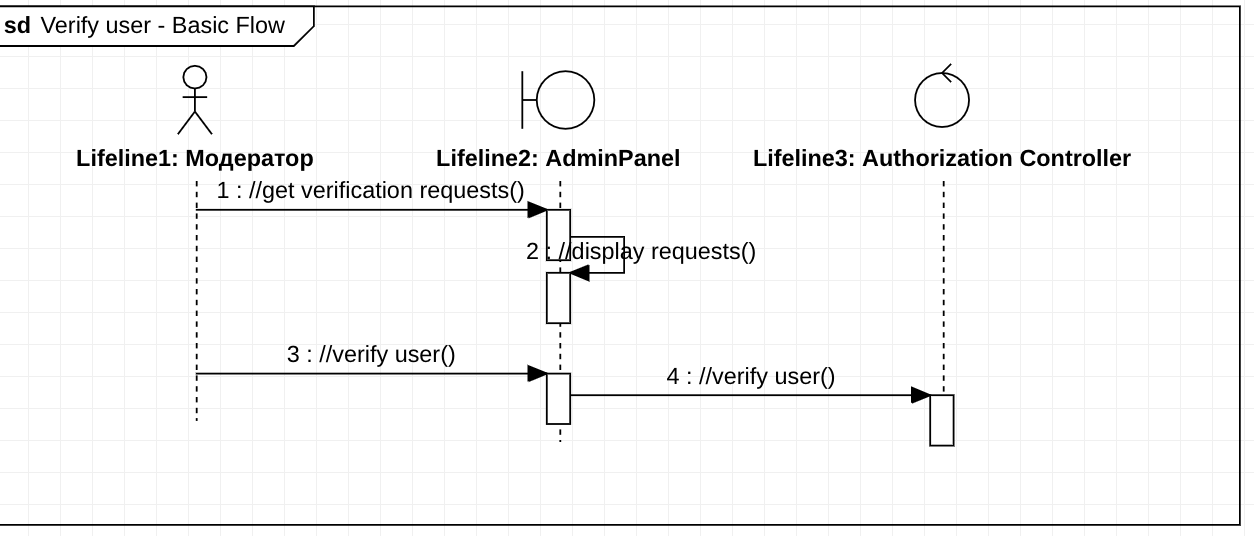


Рис. 3.7 (Діаграма послідовності верифікації користувача)

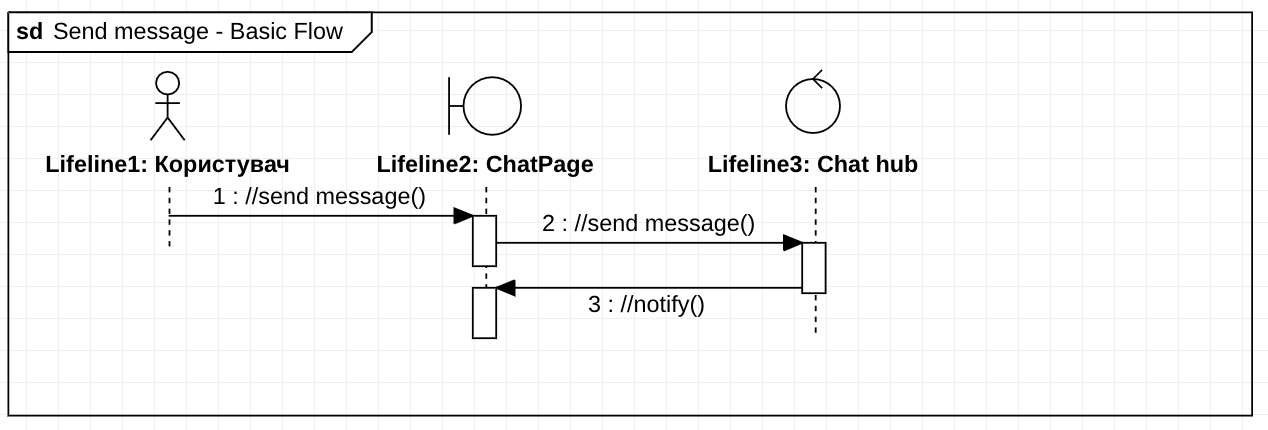


Рис. 3.8 (Діаграма послідовності надсилання повідомлення)

## Діаграма класів

Діаграма класів - це діаграма, яка моделює структуру системи шляхом відображення класів, їх атрибутів та методів, а також взаємозв'язків між класами.

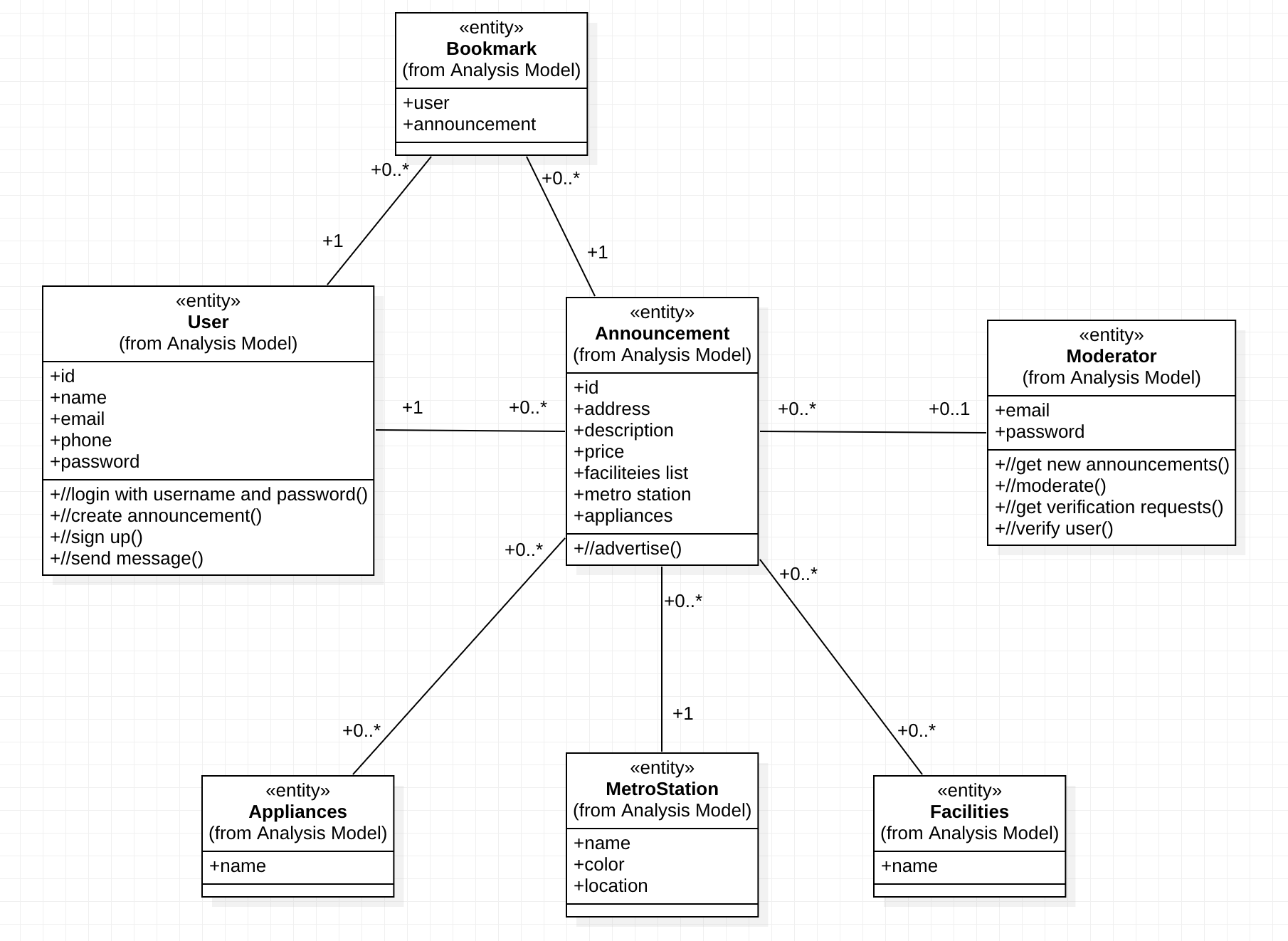


Рис. 3.9 (Діаграма класів)

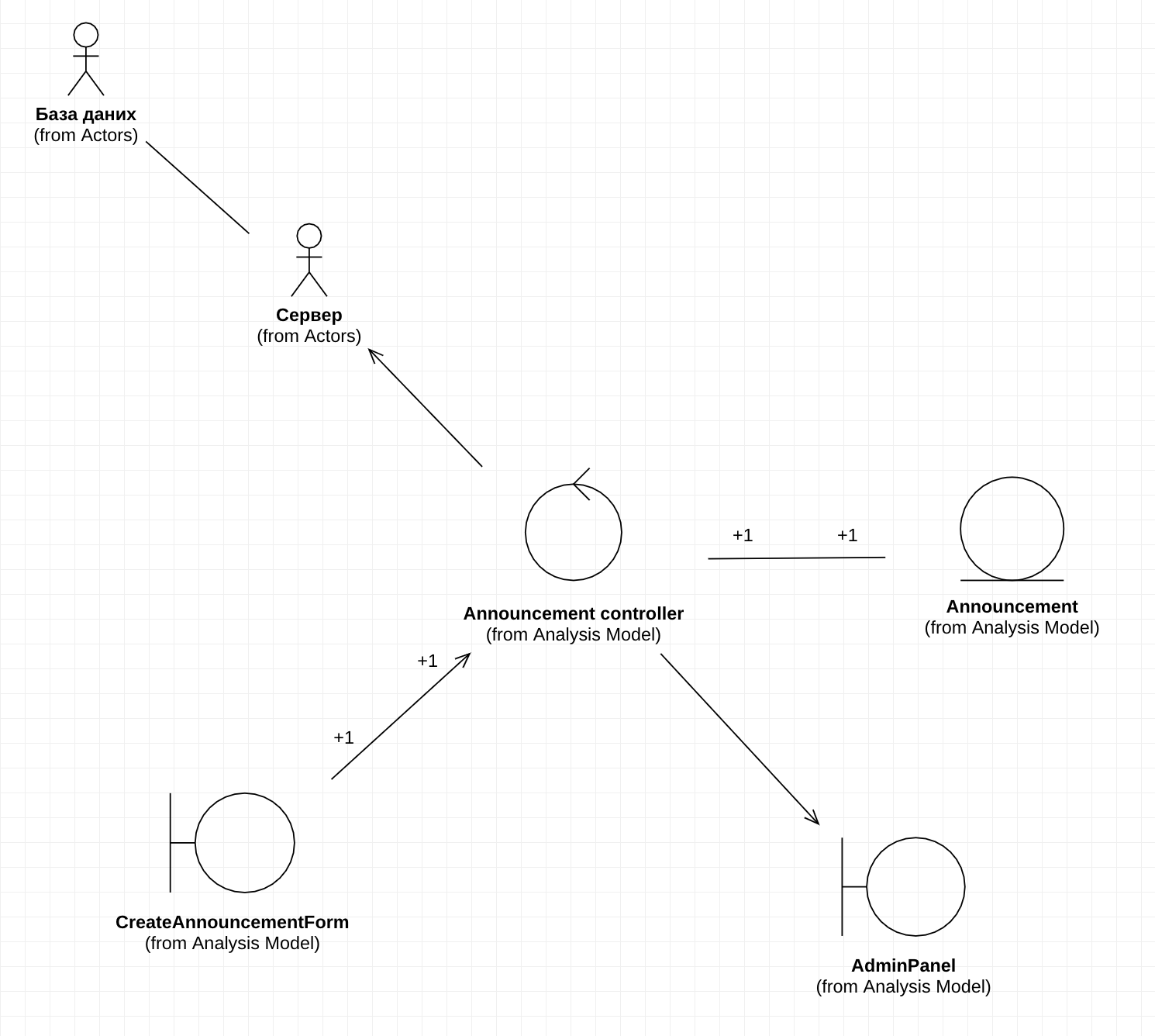


Рис. 3.10 (Діаграма класів VOPC (full))

## Діаграма розгортання

Діаграма розгортання (Deployment diagram) - це діаграма, яка моделює фізичне розташування апаратного та програмного забезпечення системи. Вона відображає структуру та конфігурацію апаратних компонентів (серверів, комп'ютерів, мережевих пристроїв) та програмних компонентів (програм, модулів, бібліотек) у системі. Діаграма розгортання показує зв'язки між цими компонентами та мережами, а також визначає способи комунікації між ними.

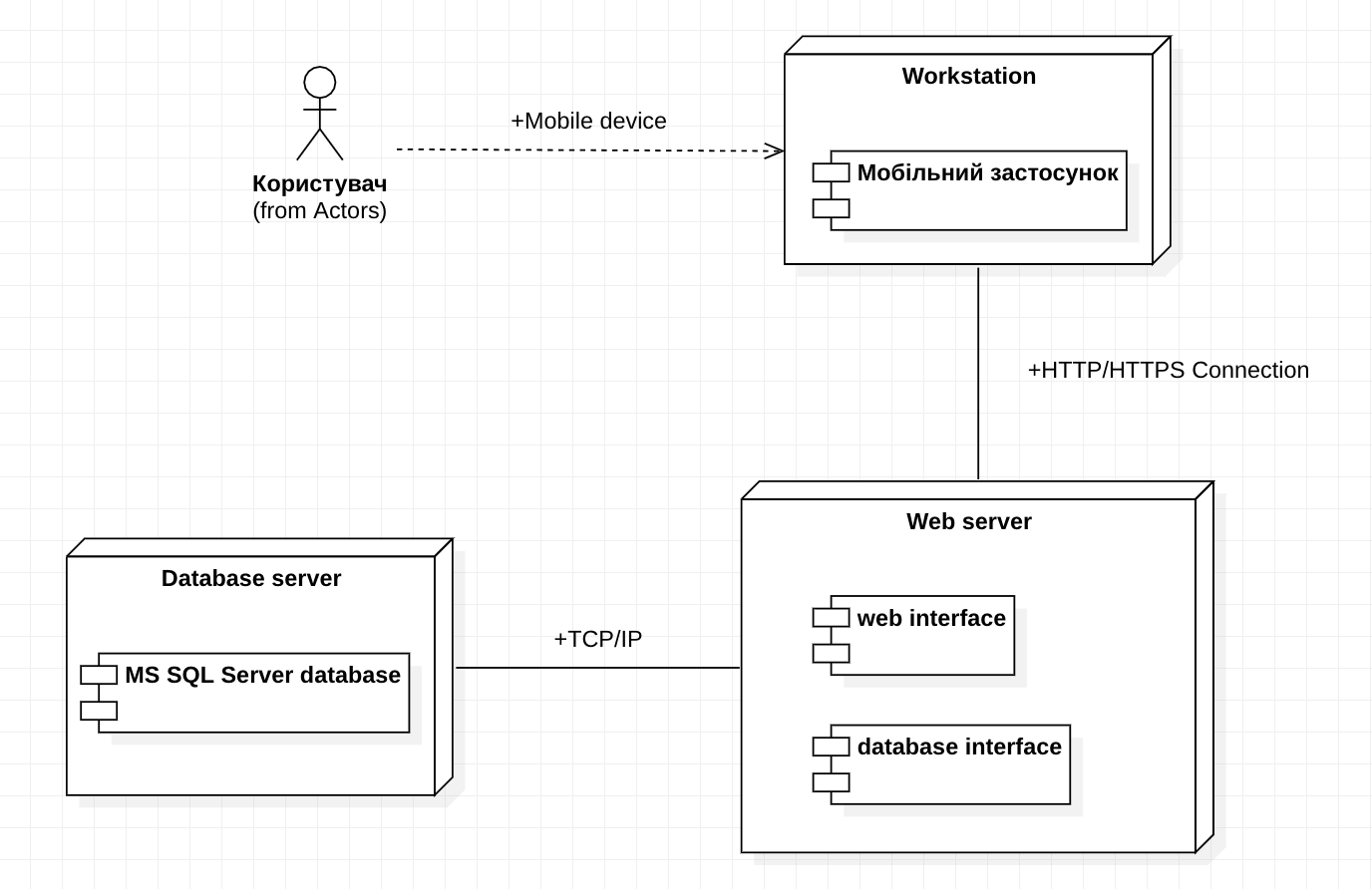


Рис. 3.11 (Діаграма розгортання)