

ЛАБОРАТОРНА РОБОТА 5

РОЗРОБКА UML–ДІАГРАМ ФІЗИЧНОГО РІВНЯ ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

Мета роботи – ознайомитись з методикою побудови UML-діаграм фізичного рівня проектування програмної системи (ПС).

5.1. Загальні відомості

Діаграма компонентів

Діаграма компонентів (component diagram) описує особливості фізичного представлення програмної системи. Вона дозволяє визначити архітектуру розроблюваної системи, встановивши залежності між програмними компонентами (вихідним ко-дом, програмними модулями тощо).

У розробці діаграм компонентів беруть участь як системні аналітики та архітек-тори, так і програмісти. Для опису фізичних програмних сутностей у мові UML за-стосовується спеціальний термін: компонент (*component*). Компонент реалізує де-який набір інтерфейсів і слугує для загального позначення елементів фізичного представлення моделі. Для графічного позначення компонента використовується спеціальний символ – прямокутник зі вставленими зліва двома дрібнішими прямо-кутниками. Усередині великого прямокутника записується ім'я компонента і при необхідності деяка додаткова інформація. Ім'я компонента підпорядковується зага-льним правилам іменування елементів моделі у мові UML і може складатися з будь-якої кількості букв, цифр та деяких інших символів.

Таким чином, компоненти на UML–діаграмах розгортання являють собою фізич-ні файли, які можуть бути безпосередньо виконані (наприклад з розширенням exe, dll та ін.) або такі, що використовуються для отримання кінцевої ПС, тобто це файли з вихідними текстами програм (з розширеннями php або cpp для мови C ++), Web-сторінки з розширенням html і файли довідки з розширенням hlp та ін.

На рис. 5.1 показаний приклад простої діаграми компонентів [4].

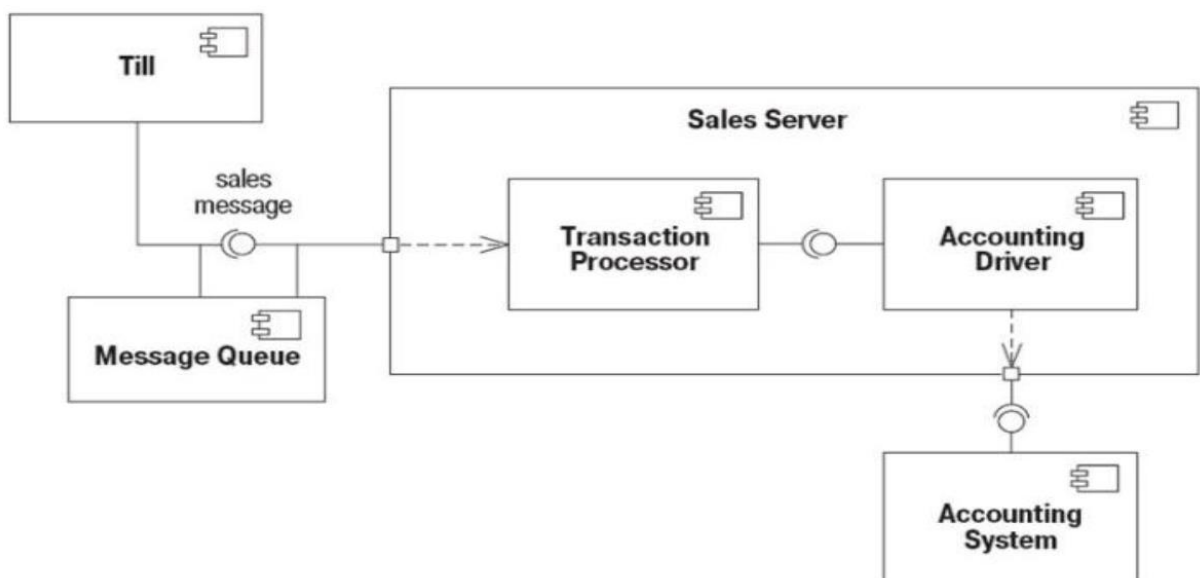


Рисунок 5.1 – Приклад діаграми компонентів [4]

У цьому прикладі компонент Till (Каса) може взаємодіяти з компонентом Sales Server (Сервер Продажу) за допомогою інтерфейсу sales message (повідомлення про продаж). Оскільки мережа ненадійна, то компонент Message Queue (Черга Повідомлень) встановлено так, щоб каса могла спілкуватися з сервером, коли мережа працює, і розмовляти з чергою повідомлень, коли мережа відключена. Тоді черга повідомлень зможе поговорити з сервером, коли мережа знову стане доступною. У результаті чергу повідомлень надає інтерфейс для розмови з касою, і вимагає такий же інтерфейс для розмови з сервером. Сервер розділений на два основних компоненти: Transaction Processor (Процесор транзакцій) реалізує інтерфейс повідомлень, а Accounting Driver (Драйвер рахунків) спілкується з Accounting System (Система обліку рахунків).

Діаграма розгортання

Діаграма розгортання (deployment diagram) призначена для візуалізації елементів і компонентів ПС, які існують лише на етапі її виконання (*runtime*). При цьому подаються тільки компоненти-екземпляри програми, які є виконуваними файлами або динамічними бібліотеками. Ті компоненти, які не використовуються на етапі виконання, на діаграмі розгортання не відображаються. Так, компоненти з вихідними текстами програм можуть бути присутніми тільки на діаграмі компонентів, а на діаграмі розгортання вони не вказуються.

Діаграма розгортання містить графічні зображення процесорів, пристроїв, процесів і зв'язків між ними. На відміну від діаграм логічного подання (напр., діаграми класів) діаграма розгортання є єдиною для ПС в цілому, оскільки повинна повністю відображати особливості її реалізації. Розробка діаграми розгортання, як правило, є останнім етапом специфікації моделі ПС.

При розробці діаграми розгортання переслідують такі цілі:

- визначити розподіл компонентів системи по її фізичних вузлах деякої обчислювальної мережі (network);
- показати фізичні зв'язки між усіма вузлами реалізації ПС на етапі її виконання;
- виявити вузькі місця системи і ре-конфігурувати її топологію для досягнення необхідної продуктивності.

Діаграми розгортання розробляються спільно системними архітекторами, інженерами-системотехніками та програмістами. Вузол (*node*) являє собою деякий фізично існуючий елемент ПС, що володіє певним обчислювальним ресурсом. Як обчислювальний ресурс вузла може розглядатися наявність деякого об'єму електронної або магнітооптичної пам'яті або процесора. В останній версії мови UML поняття вузла розширено і може включати в себе не тільки обчислювальні пристрої, а й інші механічні або електронні пристрої, такі як датчики, принтери, модеми, цифрові камери, сканери і маніпулятори. Графічно на діаграмі розгортання вузол зображується у формі тривимірного куба. Вузол має власне ім'я, яке вказується всередині його графічного символу.

Нижче наведені 2 діаграми розгортання компонентів, які являють собою ПС, яка розроблена з використанням: 1) компонентів технології JEE та 2) мови створення серверних сценаріїв PHP. Обидві ці ПС імплементують функціональність, яка представлена на наступній діаграмі сценаріїв використання (рис. 5.2).

Login / PW PW Recovery Admin User Authorization <<extend>>
<<include>> <<realize>>

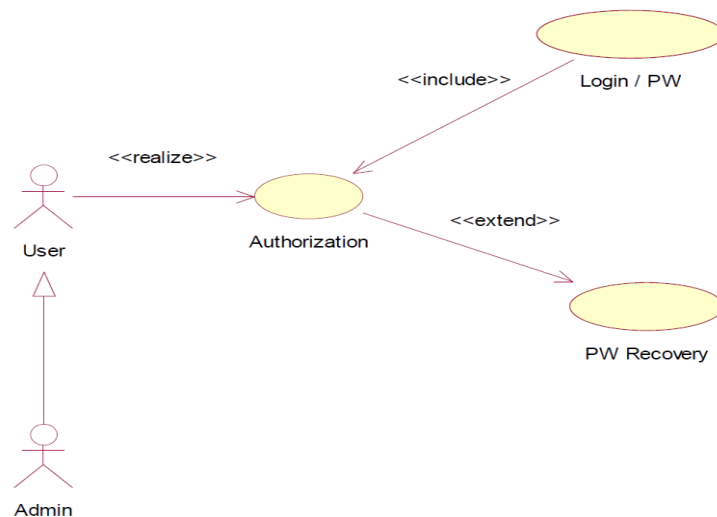


Рисунок 5.2 – Сценарій авторизації користувача у Web-системі

Реалізація на платформі Java

На вузлі *Client* розгорнуто два компоненти (рис. 5.3):

1. Web Browser – програма веб-браузер, що надає користувачеві доступ до системи.
2. JavaScript Framework – каркас javascript, що є необхідним для програмної реалізації певної бізнес-логіки на стороні клієнта.

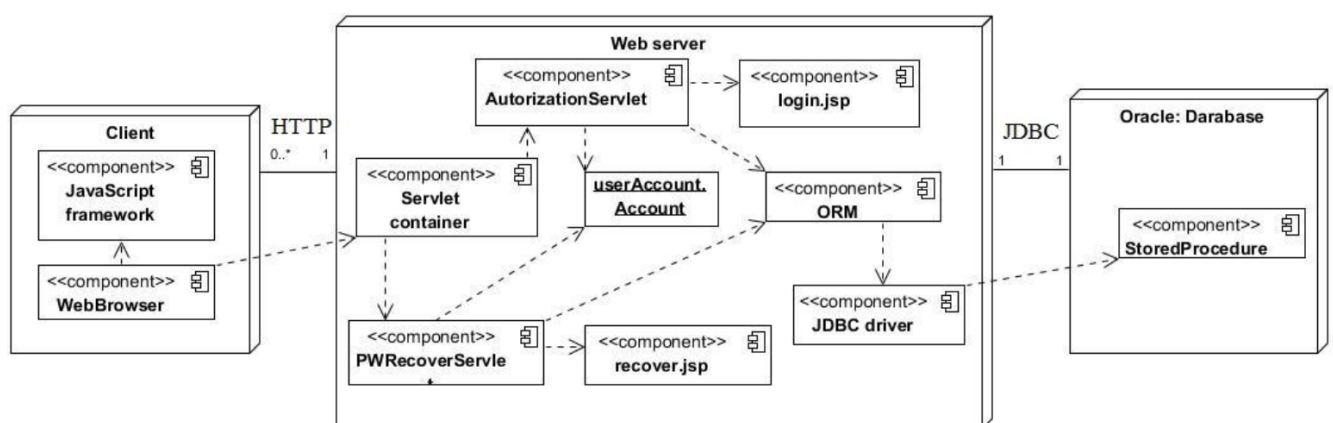


Рисунок 5.3 – Діаграма розгортання для реалізації на Java

Вузол Web server – це веб-сервер системи, клієнт взаємодіє з ним за допомогою протоколу HTTP. Розмірність відповідного зв'язку зазначено як «0..* : 1», тобто передбачено можливість одночасної роботи декількох клієнтів із 1 Web-сервером.

На цьому вузлі розгорнуто такі компоненти:

1. Servlet container – це контейнер java-сервлетів, що відповідає за виконання серверної логіки.
2. AuthorizationServlet – сервлет, що реалізує процес аутентифікації у системі.
3. PWRecoverServlet – сервлет, що реалізує процес відновлення пароля користувача.
4. ORM – компонент, що реалізує об'єктно-реляційне відображення (object-relational mapping) при роботі з базою даних системи.
5. login.jsp – це серверна jsp-сторінка, що відображає дані на етапі аутентифікації.
6. Recover.jsp – це jsp-серверна сторінка, що відображає дані на етапі відновлення пароля користувача.
7. userAccount – це програмний об'єкт, який створюється у процесі роботи та містить обліковий запис користувача.
8. JDBC Driver – це драйвер, що виконує підключення до бази даних. Цей об'єкт створюється при аутентифікації у системі або при відновленні пароля.

Вузол *Database* – це вузол розміщення бази даних системи, яка реалізована за допомогою конкретної СКБД Oracle, зв'язок з цим вузлом забезпечується за допомогою протоколу JDBC, при цьому розмірність цього зв'язку визначена як «1:1», тобто в системі, що розробляється, передбачено мати 1 Web-сервер і 1 сервер БД. У контейнері сервера БД передбачено також наявність компоненту *StoredProcedure* – це відповідні збережені процедури, що імплементують додаткову бізнес-логіку системи.

Реалізація на платформі PHP

Вузол *Client* має два компоненти (рис. 5.4):

1. Web Browser – програма веб-браузера, що надає користувачеві доступ до системи авторизації;
2. JavaScript Framework – каркас javascript, що необхідний для реалізації додаткової бізнес-логіки на стороні клієнта.

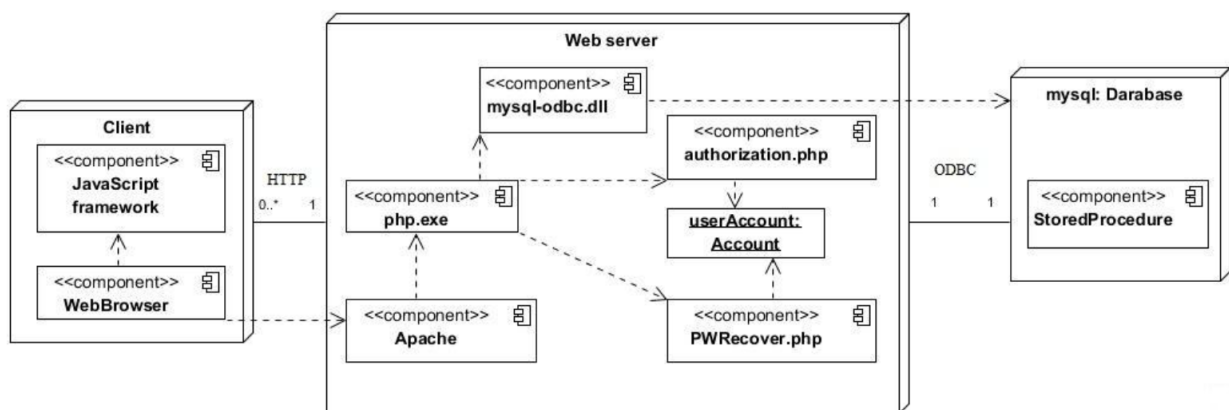


Рисунок 5.4 – Діаграма розгортання у разі реалізації на PHP

Вузол *Web server* – це веб-сервер системи, клієнт взаємодіє з ним за допомогою протоколу HTTP. Розмірність відповідного зв'язку зазначено як «0..* : 1», тобто передбачено можливість одночасної роботи декількох клієнтів із 1 Web-сервером.

На ньому розгорнуто такі компоненти:

1. Apache – це компонент веб-сервера, що забезпечує виконання PHP-скриптів (сторінок).
2. authorization.php – файл зі скриптом, що реалізує процес аутентифікації в системі.
3. PWRecover.php – файл зі скриптом, що реалізує процес відновлення пароля користувача.
4. php.exe – інтерпретатор мови PHP.
5. php-mysql.dll – це бібліотека драйверів, що дозволяє виконувати підключення php-сторінок до бази даних системи.
6. userAccount – це об'єкт, що містить обліковий запис користувача. Цей об'єкт створюється при аутентифікації в системі або при відновленні пароля.

Вузол *Database* – це вузол розміщення бази даних системи, яка реалізована в цьому проекті за допомогою конкретної СКБД MySQL, зв'язок з цим вузлом забезпечується за допомогою протоколу ODBC, при цьому розмірність цього зв'язку ви-значена як «1:1», тобто у системі, що розробляється, передбачено мати 21 Web-сервер і 1 сервер БД. У контейнері сервера БД передбачено також наявність компонента StoredProcedure – це відповідні збережені процедури, що імплементують додаткову бізнес-логіку системи.

Вузол Database – це база даних системи, сервер взаємодії з БД за допомогою протоколу JDBC. Компонент StoredProcedure – це хранимі процедури, що містяться у базі даних.

5.2. Виконання лабораторної роботи

1. Розробити діаграму компонентів для обраної предметної області.
2. Розробити діаграму розгортання для обраної предметної області.

5.3. Висновки до виконаної роботи та оформлення звіту

Зміст звіту:

- 1) титульний аркуш;
- 2) короткі теоретичні відомості;
- 3) результати практичного виконання, висновки та рекомендації.

Контрольні запитання

1. Для якого етапу проектування ПС використовуються діаграми компонентів та діаграми розгортання?
2. Для чого розробляється діаграма компонентів?
3. Для чого розробляється діаграма розгортання?

4. Що являють собою компоненти на UML–діаграмах розгортання?
5. Для чого застосовуються діаграми розгортання?
6. Що таке вузол? Як він зображується в UML?
7. У чому полягає різниця між діаграмами розгортання компонентів, розроблених з використанням технологій JEE та PHP?
8. Що таке вузол Web-server? Які компоненти розгорнуті на ньому?
9. З яких компонентів складається вузол Client?
10. Яку функціональність реалізує в обох діаграмах вузол Database?