

Лабораторна робота №6-7

Мета: навчитися моделювати програмні системи за допомогою компонентних діаграм UML, розуміти структуру компонентів, їх взаємодію та залежності.

Завдання 6. Побудова діаграмами компонентів

Діаграма компонентів візуалізує високорівневу структуру системи, розбиваючи її на функціональні модулі та показуючи залежності між ними.

Для системи "TransferAnalytics" виділено наступні архітектурні рівні (пакети):

1. Клієнт (Client Side):

- Відповідає за взаємодію з користувачем (UI).
- Включає компоненти для відображення профілів гравців, графіків статистики та керування станом (State Management).
- Комунікує з сервером через HTTP/API Client.

2. Сервер (Server Side / Business Logic):

- Приймає запити через API Gateway / Routes.
- Transfer Manager: Керує логікою трансферів (перевірка бюджетів, зміна статусів).
- Player Manager: Відповідає за оновлення статистики та ринкової вартості.
- Auth Service: Забезпечує безпеку та авторизацію.

3. Зберігання даних (Data Access):

- Репозиторії для доступу до бази даних (MySQL/PostgreSQL).

4. Зовнішні сервіси (External Services):

- Інтеграція з зовнішніми футбольними API (наприклад, для отримання результатів матчів у реальному часі).

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка». 25.121.22.000–Пр6-7		
Розроб.	Свистанюк Н.О.				Звіт з лабораторної роботи		
Перевір.	Левківський В.Л.						
Керівник							
Н. контр.							
Зав. каф.					ФІКТ Гр. ІПЗ-22-3		
					Літ.	Арк.	Аркушів
						1	3

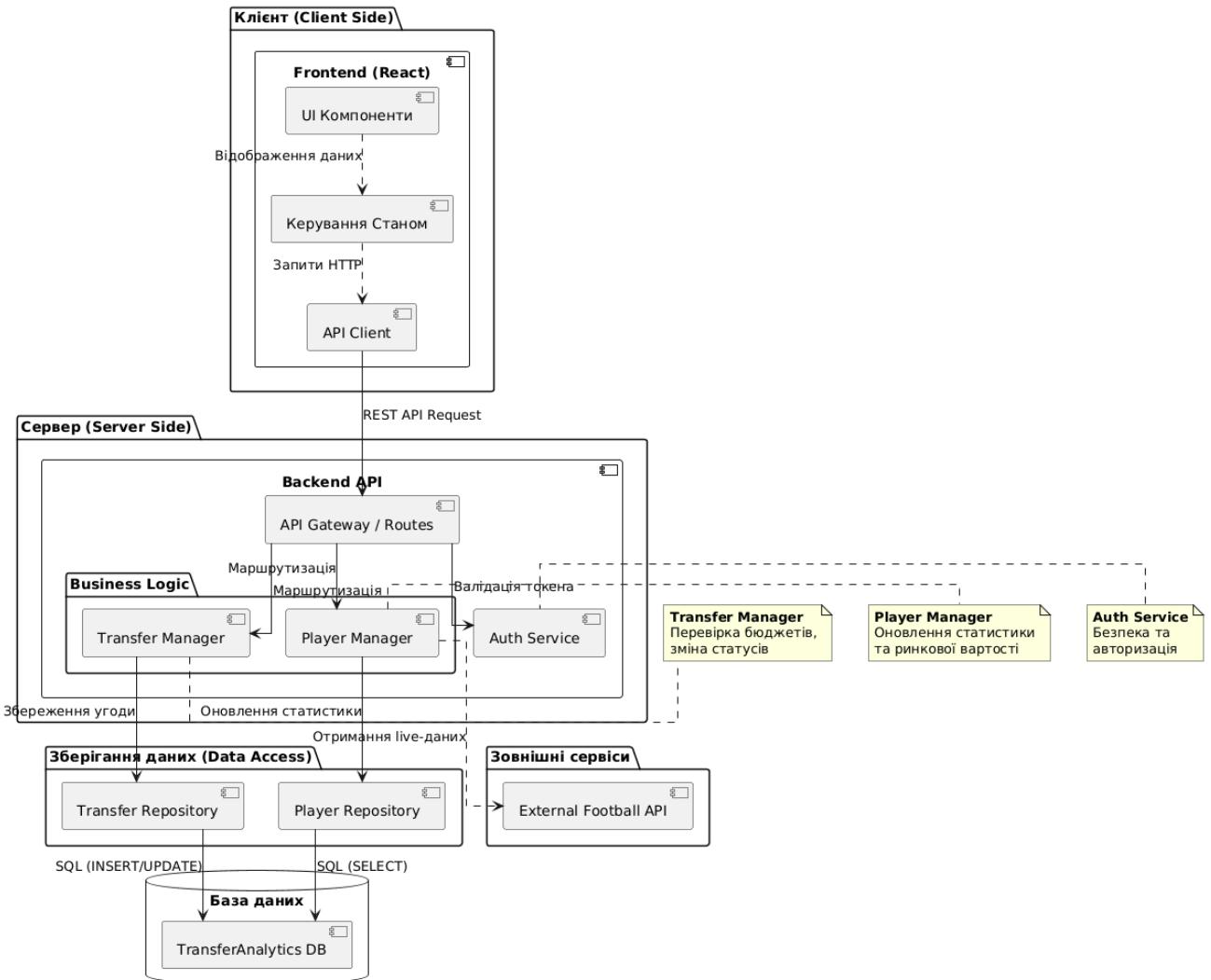


Рис.6.1.Діаграма компонентів

Завдання 7. Побудова діаграми розгортання

Діаграма розгортання моделює фізичне розміщення програмних артефактів.

Клієнтський пристрій (Client Node):

- Це персональний комп'ютер або мобільний пристрій користувача.
- На ньому у середовищі Веб-браузера виконується Frontend-додаток (Single Page Application, React Build). Він надсилає запити до сервера через захищений протокол HTTPS (порт 443).

Веб-сервер / Реверс-проксі (Web Server Node):

- Вхідна точка у серверну інфраструктуру.
- Тут працює сервер Nginx, який виконує дві функції:
 - Віddaє Статичні файли (зображення, стилі, скрипти), що розвантажує основний сервер додатку.
 - Працює як реверс-проксі, перенаправляючи API-запити на сервер додатку (на внутрішній порт 5000).

Сервер Додатку (App Server Node):

- Вузол, де зосереджена бізнес-логіка.
- Тут у середовищі Node.js Runtime виконується Backend API (на базі Express.js).

		Свистанюк Н.О.					Арк.
		Левківський В.Л.					
Змн.	Арк.	№ докум.	Підпис	Дата		ДУ «Житомирська політехніка». 25.121.22.000 – Прб-7	2

- Сервер має доступ до локальної Файлової системи (директорія reports/) для збереження та читання згенерованих PDF-звітів скаутів, що дозволяє не зберігати великі бінарні файли у базі даних.

Сервер Бази Даних (Database Node):

- Окремий виділений сервер для СУБД MySQL.
- Тут розгорнуто схему бази даних TransferAnalytics DB.
- Доступ до цього вузла дозволено виключно з Сервера Додатку по внутрішньому порту 3306, що забезпечує захист даних від зовнішнього втручання.

Зовнішні сервіси (External Node):

- Представлені компонентом Football Stats API. Сервер додатку періодично звертається до нього для синхронізації реальної статистики гравців.

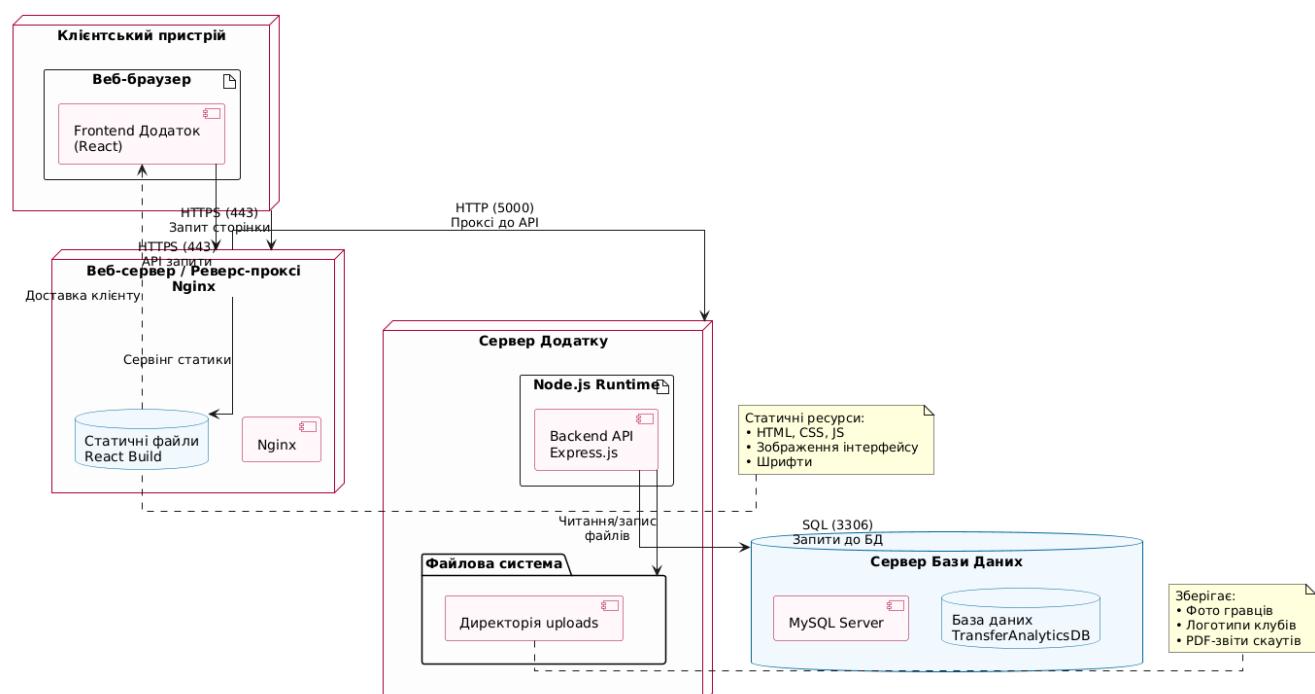


Рис.7.1.Діаграма розгортання

Висновок: в ході виконання лабораторної роботи №6-7 було набуто практичних навичок архітектурного моделювання програмних систем за допомогою мови UML. Створені діаграми слугують основою для подальшої розробки, розгортання та підтримки системи, забезпечуючи чітке розуміння її будови та взаємозв'язків між елементами.

Репозиторій: <https://github.com/Svistaniuk/SMAA>

		Свистанюк Н.О.				Арк.
		Левківський В.Л.				
Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка». 25.121.22.000 – Прб-7	3