**Architecture and Design**

**laboratory work 10**

**EXERCISE 01 – APPLYING architectural styles**

**Выбор Архитектурных Стилей**

Микросервисный стиль выбран для проекта JSE, потому что он обеспечивает:

* **Модульность**: Каждый сервис отвечает за определенную часть функциональности, что облегчает разработку, тестирование и развертывание.
* **Масштабируемость**: Систему можно легко масштабировать, добавляя или заменяя сервисы по мере необходимости.
* **Гибкость**: Компоненты могут взаимодействовать через асинхронные механизмы, такие как NATS Streaming, что позволяет разрабатывать независимые сервисы.

**Описание Архитектуры**

На диаграмме представлена архитектура JSE, в которой используются микросервисы для различных задач:

1. **JSE UI**: Это фронтенд проекта. Он может быть реализован как веб-приложение или мобильное приложение, предоставляющее пользователям доступ к функционалу JSE. Он взаимодействует с JSE Service для обработки запросов.
2. **JSE Service**: Этот сервис служит центральной точкой для обработки запросов из JSE UI и координирует взаимодействие между другими микросервисами. Он принимает запросы от фронтенда и распределяет их к соответствующим сервисам, таким как User Management Service или Jobs Service.
3. **User Management Service**: Этот сервис отвечает за управление пользователями, включая регистрацию, аутентификацию и управление профилями. Он взаимодействует с JSE DB для хранения данных о пользователях.
4. **Jobs Service**: Этот сервис отвечает за управление вакансиями. Он может обрабатывать добавление, редактирование и удаление вакансий, а также взаимодействовать с JSE DB для хранения информации о вакансиях.
5. **Admin Service**: Этот сервис предназначен для административных задач, таких как управление правами доступа и мониторинг системы. Он может взаимодействовать с другими сервисами для выполнения своих функций.
6. **Auth Service**: Этот сервис отвечает за аутентификацию пользователей. Он может выдавать токены или другие средства аутентификации, обеспечивая безопасность системы.
7. **JSE DB**: Это база данных, используемая для хранения информации о пользователях, вакансиях и административных данных. Она обеспечивает надежное хранилище для системы JSE.
8. **NATS Streaming**: Это система обмена сообщениями, которая позволяет сервисам взаимодействовать асинхронно. Сервисы могут отправлять сообщения и подписываться на них, что обеспечивает гибкость и надежность.

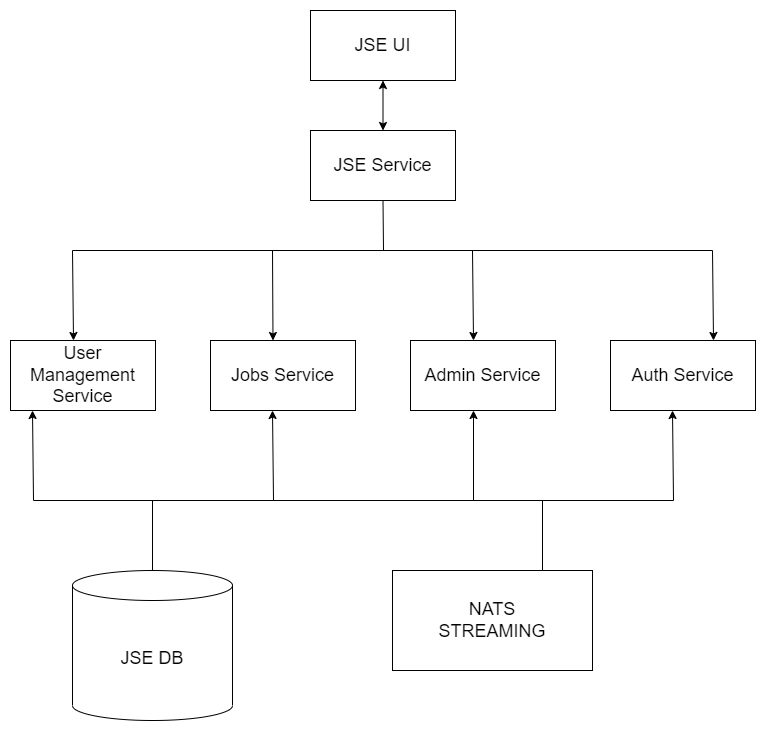
**Взаимодействие Между Компонентами**

На диаграмме показано взаимодействие между различными сервисами:

* **JSE UI** отправляет запросы к JSE Service, который координирует взаимодействие с другими сервисами.
* **JSE Service** взаимодействует с User Management Service, Jobs Service, Admin Service и Auth Service для обработки запросов, связанных с пользователями, вакансиями, администрированием и аутентификацией.
* **JSE DB** используется для хранения данных, а NATS Streaming обеспечивает асинхронное взаимодействие между сервисами.

**Графическое Представление Архитектуры**

На приведенной диаграмме отображена микросервисная архитектура JSE:



**Заключение**

Применение микросервисного стиля к проекту JSE обеспечивает модульность, гибкость и масштабируемость. Сервисы могут разрабатываться и развертываться независимо друг от друга, а взаимодействие осуществляется через асинхронные механизмы, такие как NATS Streaming и REST API. База данных JSE DB предоставляет надежное хранилище данных.