



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №9
РІЗНІ ВИДИ ВЗАЄМОДІЇ ДОДАТКІВ: CLIENT-SERVER,
PEER-TO-PEER, SERVICE-ORIENTED ARCHITECTURE
Варіант 4

Виконав
студент групи ІА – 13:
Запотоцький І.А

Перевірив:
М’який М.Ю

Завдання:

1. Ознайомитися з короткими теоретичними відомостями.
2. Реалізувати частину функціоналу робочої програми у вигляді класів та їхньої взаємодії для досягнення конкретних функціональних можливостей.
3. Реалізувати взаємодію програми в одній з архітектур відповідно до обраної теми.

Варіант:

4)

Графічний редактор (proxy, prototype, decorator, bridge, flyweight, SOA) Графічний редактор повинен вміти створювати / редагувати растрові (або векторні на розсуд студента) зображення в 2-3 основних популярних форматах (bmp, png, jpg), мати панель інструментів для створення графічних примітивів, вибору кольорів, нанесення тексту, додавання найпростіших візуальних ефектів (ч/б растр, інфрачервоний растр, 2-3 на вибір учня), роботи з шарами.

Хід роботи

Сервісно-орієнтована архітектура (SOA) є підходом до розробки програмного забезпечення, який спрямований на створення великих та складних систем з використанням набору сервісів, які взаємодіють між собою. SOA розглядає компоненти програмної системи як сервіси, які можуть бути незалежно розгортані, масштабовані та використовувані для взаємодії через стандартизовані інтерфейси.

Головні принципи та характеристики SOA включають:

Сервіси як основна одиниця: SOA розглядає функціональність програмного забезпечення як набір сервісів, які можуть бути викликані та використані іншими компонентами системи.

Стандартизовані інтерфейси: Сервіси у SOA взаємодіють один з одним через стандартизовані інтерфейси, такі як HTTP або SOAP (Simple Object Access Protocol), що дозволяє різноманітним компонентам спілкуватися між собою.

Інкапсуляція: Кожен сервіс в SOA повинен бути самодостатнім та інкапсульованим, тобто мати внутрішню реалізацію, яка приховується від інших сервісів.

Ітероперабельність: SOA сприяє можливості взаємодії між сервісами, незалежно від того, на якій технології вони побудовані, що сприяє ітероперабельності в різноманітних середовищах.

Масштабованість та повторне використання: SOA дозволяє створювати розширювані системи, де сервіси можна масштабувати та повторно використовувати для різних завдань.

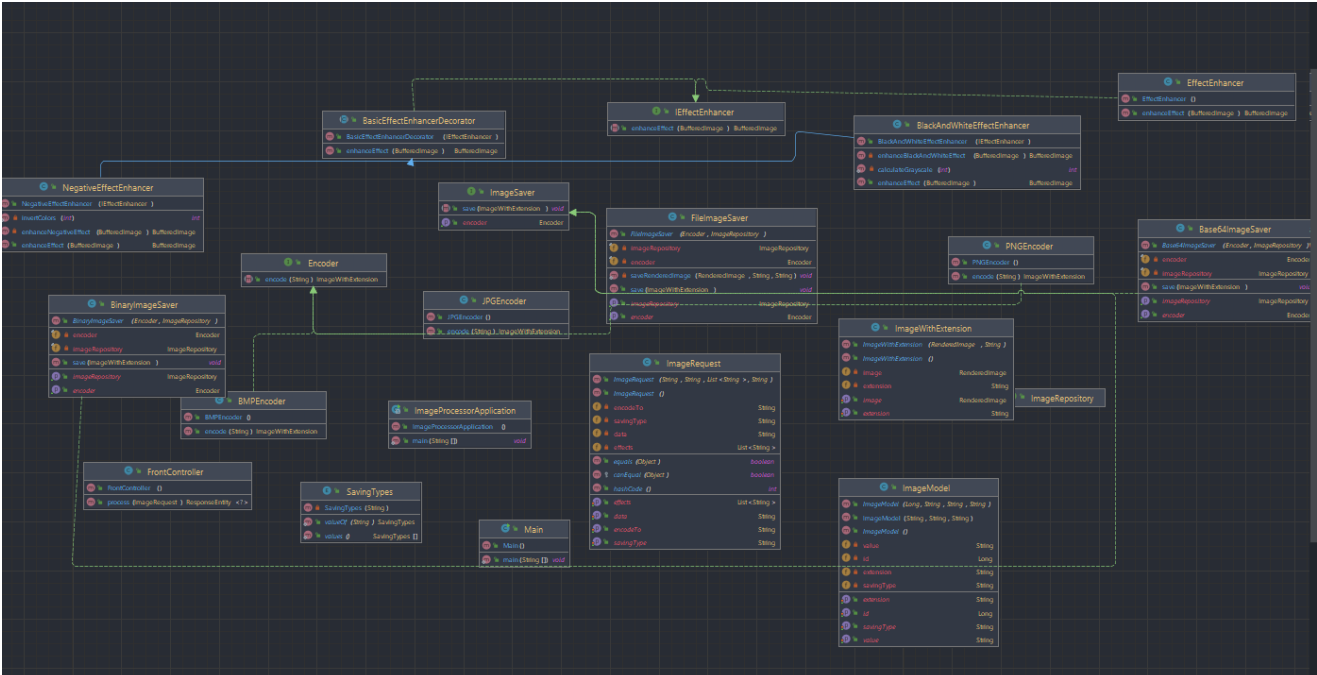
Легкість модифікації та адаптації: SOA сприяє гнучкості системи, що дозволяє легко вносити зміни або адаптувати її до нових вимог без значних змін у загальній структурі.

ЗОА дозволяє підприємствам побудовувати гнучкі та розширювані інформаційні системи, які легко адаптуються до змін у бізнес-потребах. Цей підхід розглядається як ефективний засіб покращення архітектури програмного забезпечення та сприяє побудові дієздатних, масштабованих та ефективних бізнес-систем.

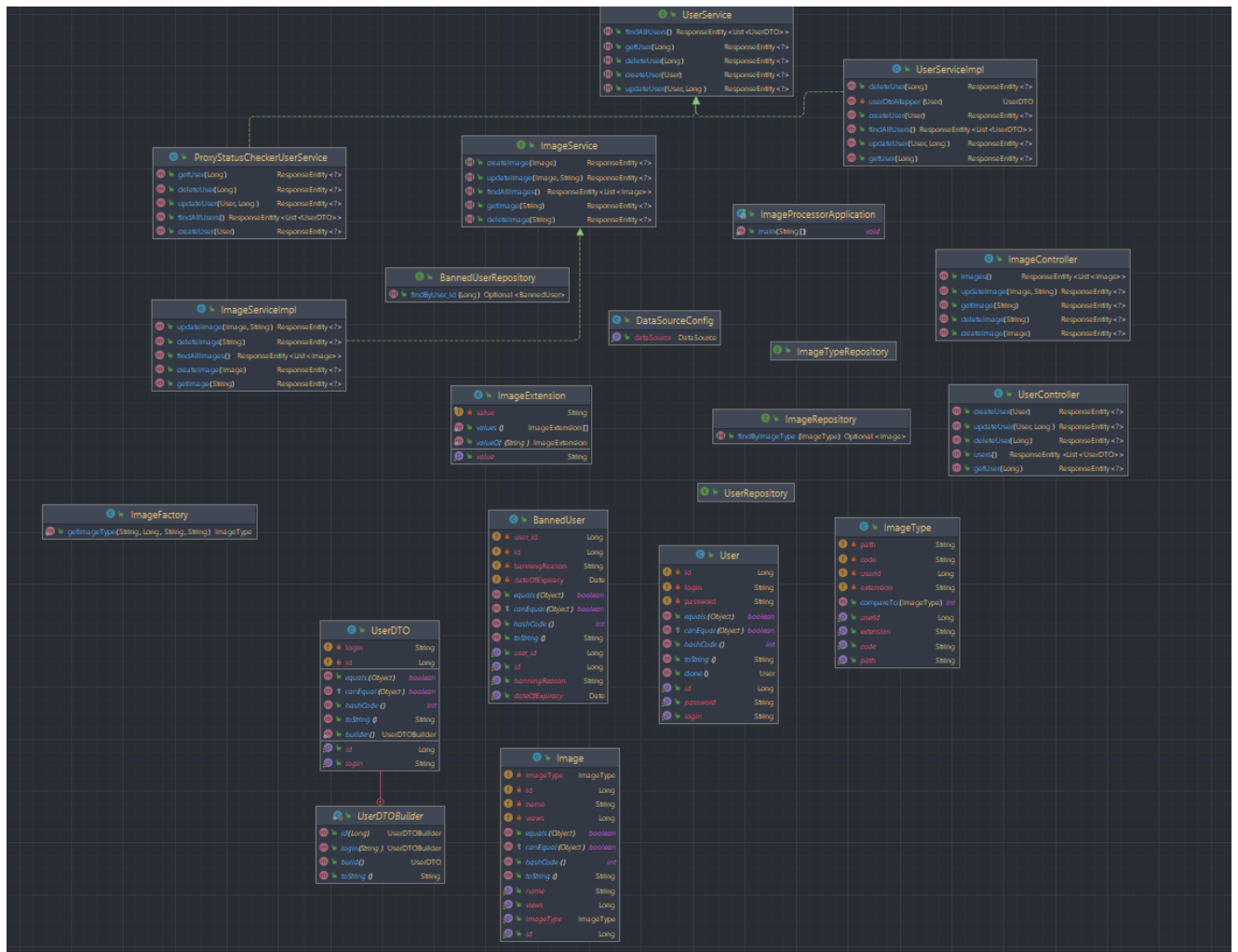
В моєму випадку SOA інтерпритується та використовується як пов'язання двох незалежних мікросервісів, які використовують одну базу даних. (Конкретно це можна побачити в діаграмах компонентів тощо)

Два сервіса є незалежними спрінг додатками та мають окрему структуру та хоста на сервері. З'єднання додатків проходить за допомогою http викликів між серверами.

Додаток-обробчик зображень:



Основний додаток по роботі з юзерами та фронтендом:



Таким чином, обидва додатки можуть працювати окремо, в чому і є ідея SOA, але Додаток використовує обидва сервіси для найкращого експерієнсу.