**Міністерство освіти і науки України**

**Національний авіаційний університет**

**Навчально-науковий інститут комп’ютерних**

**інформаційних технологій**



Лабораторна робота № 4-5

з дисципліни «Архітектура та проектування програмного забезпечення»

«ПРОЕКТУВАННЯ БАГАТОРІВНЕВОЇ АРХІТЕКТУРИ ЗАСТОСУВАННЯ»

Виконав студент:

групи ПІ-224Б

Ляшенко Б.М.

Перевірив викладач:

Київ 2023

Історія контролю

|  |  |  |
| --- | --- | --- |
| **Дата** | **Опис** | **Автор** |
| 24.04.2023 | ПРОЕКТУВАННЯ БАГАТОРІВНЕВОЇ АРХІТЕКТУРИ ЗАСТОСУВАННЯ | Ляшенко Богдан Михайлович |

**Мета**: вивчити проектування багаторівневої архітектури застосування.

**Завдання**:

1. Спроектувати застосування, розділене на рівні, а також представити у вигляді UML діаграми(-

ам) функціонал згідно з варіантом.

1.1. Представити загальну архітектуру застосування, зв’язки між рівнями.

1.2. Представити діаграми класів кожного рівня.

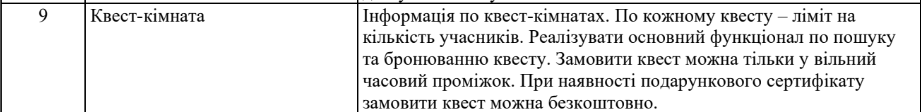
1. Розробити застосування на мові C#, яка відповідає вимогам у варіанті. Відокремити рівні доступу до даних, бізнес логіки та представлення. Для демонстрації роботи можна використати будь-який UI: консольний ввід-вивід, WinForms, WPF чи Web.

2.1. Верхній рівень – представлення (UI), для взаємодії з користувачем. Реалізація – як окремий проект в рішенні (solution). UI може бути будь-якого типу (консольне застосування, WinForms, WebForms, WPF, ASP.NET MVC, ASP.NET WebAPI). При цьому на рівні UI повинні бути тільки операції взаємодії з користувачем. UI повинен бути максимально простим, неперевантаженим великою кількістю операцій.

2.2. Проміжний рівень – бізнес логіка, реалізована як бібліотека в рішенні. На цьому рівні

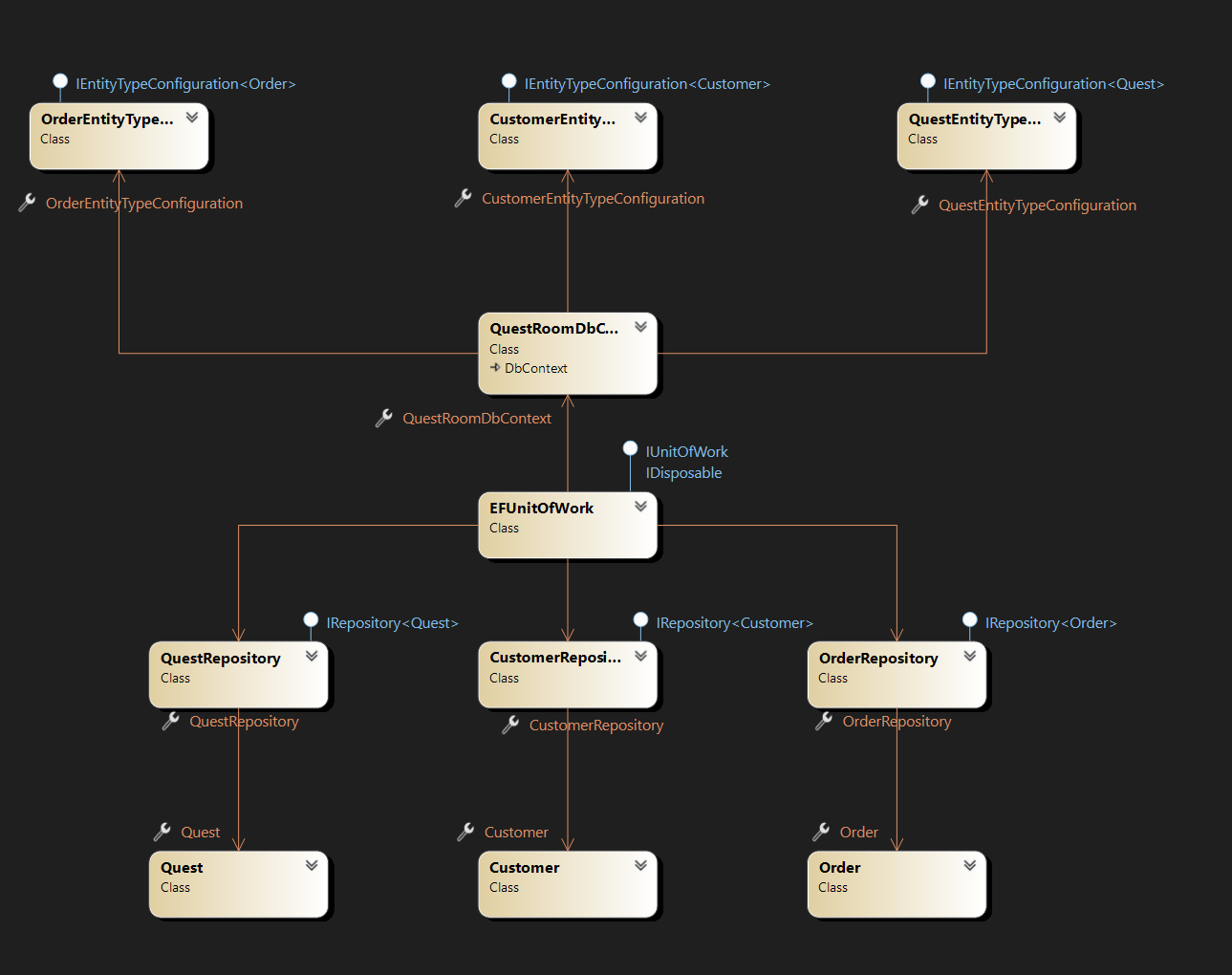
реалізований основний функціонал застосування.

2.3. Нижній рівень – бібліотека для доступу до даних. Використати Entity Framework codefirst. Доступ до даних організувати з використанням шаблонів проектування Repository та Unit of Work (UoF). Репозіторії надають доступ до набору сутностей (entities) певного типу. UoW є точкою єдиного доступу до репозіторіїв та контексту Entity Framework.

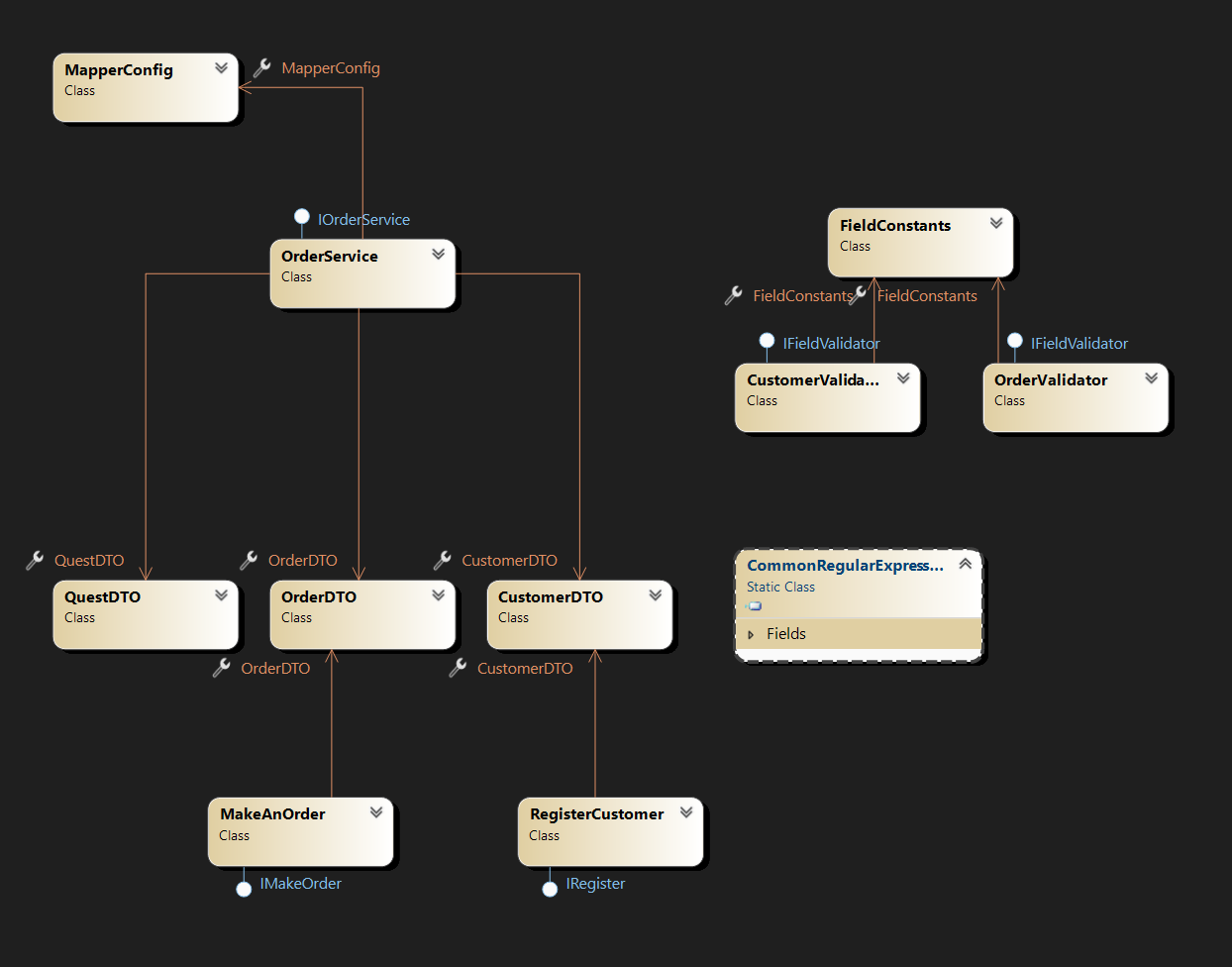
1. Шари взаємодіють між собою за наступним принципом: представлення використовує бізнес-логіку, бізнес-логіка – рівень доступу до даних. Для передачі даних крізь шари використовується технологія відображення (mapping).
2. При необхідності для більшої ізоляції основних рівнів можуть вводитися додаткові рівні (наприклад, винесення Repository та UoW).
3. Діаграма(-и) та вихідний код повинні відповідати основним принципам проектування: OOP,SOLID, Law of Demeter (LoD), DRY, YAGNI, KISS, cohesion – coupling, inheritance with caution.
4. Для отримання балів, що відповідають «задовільно» можна не використовувати mapping, реалізація шаблонів Repository та UoW може бути не ізольованою від інших рівнів, без інтерфейсів та не generic (явна імплементація репозиторію для кожної сутності), без використання зв’язків між сутностями. А також без використання репозиторію Git. Обов’язково повинні бути реалізовані CRUD операції.
5. Для отримання балів, що відповідають «добре» можна не використовувати mapping, але повинен бути механізм передачі даних з рівня на рівень (щоб сутності з DAL були тільки на BLL); реалізація шаблонів Repository та UoW повинна бути достатньо ізольованою від інших рівнів, з використанням інтерфейсів та з використанням generic репозиторію. При цьому допускається імплементація не всіх операцій саме в generic репозиторії. А також необхідно використати репозиторій Git.
6. Для отримання балів, що відповідають «відмінно» обов’язково потрібно використати mapping (будь-які підходи та бібліотеки). Реалізувати один generic репозиторій для всіх операцій з усіма сутностями. Імплементація операцій UoW та використання репозиторіїв у UoW повинна бути відокремленою. Реалізація шаблонів Repository та UoW повинна бути достатньо ізольованою від інших рівнів, щоб була можливість в рішення (solution) додати інші джерела даних таким чином, щоб не було потреби змінювати рівень DAL з EF. Також залишаються вимоги з попереднього пункту, що стосуються репозиторію Git.
7. 

**Хід** **роботи**

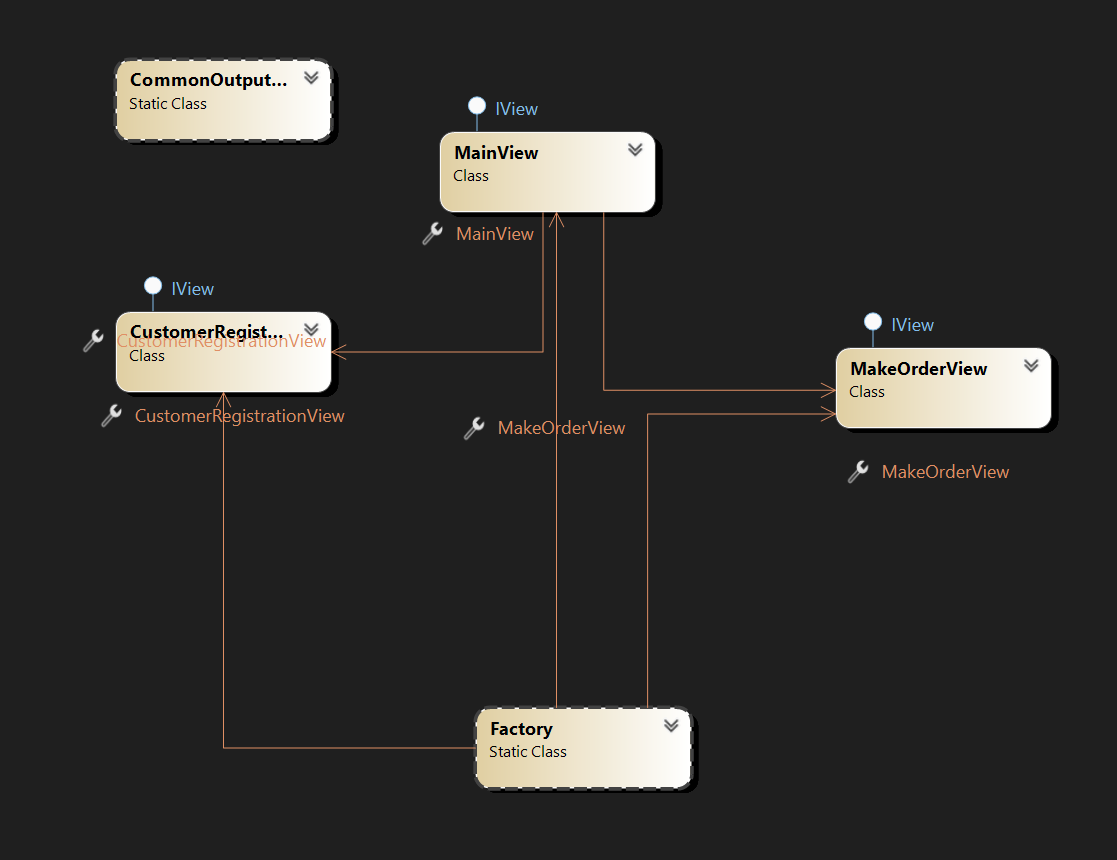
1. **Діаграма класів рівня доступу до даних.**

****

1. **Діаграма класів рівня бізнес логіки.**

****

1. **Діаграма класів рівня представлення.**

****

**Висновок**: під час виконная лабораторної роботи я вивчив проектування багаторівневої архітектури застосування.