Зміст

[Вступ 4](#_Toc10456343)

[1 Постановка задачі 5](#_Toc10456344)

[1.1 Загальні вимоги до програмного забезпечення 5](#_Toc10456345)

[1.2 Функціональні вимоги до програмного забезпечення за варіантом. 5](#_Toc10456346)

[2 Теоретичні відомості 6](#_Toc10456347)

[2.1 Трирівнева архітектура 6](#_Toc10456348)

[2.2 MVC 6](#_Toc10456349)

[3 ЗАГАЛЬНИЙ ОПИС СКЛАДОВИХ ПРОЕКТУ 8](#_Toc10456350)

[3.1 Загальна структура DAL 8](#_Toc10456351)

[3.2 Загальна структура BLL 8](#_Toc10456352)

[3.3 Загальна структура PL 9](#_Toc10456353)

[3.4 Dependency Resolver 9](#_Toc10456354)

[4 Опис DAL рівня 11](#_Toc10456355)

[4.1 Загальна структура класів DAL рівня 11](#_Toc10456356)

[4.2 Cтруктура класів, що описують сутності 11](#_Toc10456357)

[4.3 Опис засобів Entity Framework 12](#_Toc10456358)

[4.3.1 Cтруктура класів, що описують засоби Entity Framework 12](#_Toc10456359)

[4.3.2 Опис класів та їх методів 12](#_Toc10456360)

[4.4 Cтруктура інтерфейсів 13](#_Toc10456362)

[4.5 Опис репозиторіїв 13](#_Toc10456363)

[4.5.1 Cтруктура репозиторіїв 13](#_Toc10456364)

[4.5.2 Опис класів та їх методів 14](#_Toc10456365)

[5 Опис BLL рівня 16](#_Toc10456367)

[5.1 Загальна структура класів BLL рівня 16](#_Toc10456368)

[5.2 Cтруктура інтерфейсів 17](#_Toc10456369)

[5.3 Опис сервісів 17](#_Toc10456370)

[5.3.1 Cтруктура класів, що описують сервіси 17](#_Toc10456371)

[5.3.2 Опис класів та їх методів 17](#_Toc10456372)

[6 ОПИС PL рівня 21](#_Toc10456373)

[Висновки 24](#_Toc10456375)

[Перелік посилань 25](#_Toc10456376)

Вступ

Комп'ютери проникли в усі сфери діяльності людини, починаючи з початкової освіти і закінчуючи вивченням новітніх технологій, вивчення нових видів матерії, невідомих поки людству. Застосування комп'ютерних технологій полегшує процес освіти в середніх та вищих навчальних закладах як самих учнів, студентів, так і робочого персоналу. Завдяки різноманітності програмного і апаратного забезпечення сьогодні можливе використання всіх потенційних можливостей комп'ютерних технологій. Це дозволяє зберігати величезну кількість інформації, займаючи при цьому мінімальну місце.

Також комп'ютерні технології дозволяють швидко цю інформацію обробляти і тримати її в захищеному вигляді. Широке поширення ПК зіграло величезну роль у розвитку ринку праці. Автоматизація обробки інформації дозволяє в лічені секунди проробити роботу, на яку раніше губилися тижня, інформування керівників про стан підприємств і робочих місць відбувається миттєво. Збільшується економічний потенціал у галузі страхових і фінансових послуг завдяки збільшеному обміну послуг. Впровадження комп'ютерних технологій для введення нових форм зайнятості та організації праці. На розробку нових проектів витрачається значно менше часу, бо не треба витрачати масу часу на обчислювальні процеси і можна повністю присвятити час самого процесу.

# Постановка задачі

## Загальні вимоги до програмного забезпечення

Розробити програмну частину та базовий інтерфейс для демонстрації коректної роботи веб-застосунку за темою «Студентська бібліотека: облік літератури та читацьких формулярів»:

* В якості типу застосування обрати застосування з web-інтерфейсом на основі технологій asp.net MVC чи web API.
* Спроектувати та реалізувати систему класів, в основу якої покладено логічну структуру даних, для накопичення та обробки даних домену.
* Код програмної системи має складатись не менш як з трьох частин (окремих проектів) відповідно до багатошарової архітектури системи, де шарами архітектури є: шар (рівень) доступу до даних (DAL), шар бізнес логіки (BLL), шар представлення (інтерфейс програмної системи) (PL).

1. В якості типу застосування обрати застосування з web-інтерфейсом на основі технологій

asp.net MVC чи web API.

2. Спроектувати та реалізувати систему класів, в основу якої покладено логічну структуру

даних, наведену у варіанті, для накопичення та обробки даних домену відповідно

варіанту курсової роботи.

3. Структура програмної системи курсового проекту:

Код програмної системи має складатись не менш як з трьох частин (окремих проектів)

відповідно до багатошарової архітектури системи, де шарами архітектури є: шар

(рівень) доступу до даних (DAL), шар бізнес-логіки (BLL), шар представлення (інтерфейс

програмної системи) (PL).

4. Реалізувати обов’язковий контроль коректності введених даних в шарі представлення.

5. Реалізувати перевірку виняткових ситуацій в шарі бізнес-логіки та доступу до даних; в

тому числі – при роботі з даними. При необхідності створити власні класи виключень

(наприклад, виключення для ситуації перевищення ліміту кількості книжок на

абонементі в бібліотеці).

6. При написанні коду застосувати правила «хорошого» стилю програмування. А саме:

створювати мнемонічні ідентифікатори, чітко форматувати код, застосувати при

необхідності коментарі. Не використовувати public полів у класах. За необхідності

доступу до них використати властивості, індексатори чи окремі методи-аксесори. Опис

класів наводити в окремих файлах. За неохайне оформлення коду можливе зниження

оцінки за курсову роботу!

7. Пояснювальна записка має містити наступні складові:

a. Титульний лист

b. Зміст

c. Опис завдання

d. Опис шарів проекту

e. Опис інших компонентів проекту (при наявності)

f. Лістінг коду проекту (не друкувати, а зберегти на диску!)

g. Скріншоти (результати введення\виведення) проекту (не друкувати, а зберегти

на диску!)

8. До розробленого застосування та оформленої пояснювальної записки розробити

презентацію та доповідь (приблизно 5 хв.).

9. Приклад виконання курсової розглядається на лекції.

«Технології розробки ПЗ», завдання курсових робіт. Кафедра

2.4. Можливість переглянути дані конкретного документу

2.5. Можливість переглянути список всіх документів

2.5.1. Можливість відсортувати список по назві

2.5.2. Можливість відсортувати список по автору

3. Управління видачами документів

3.1. На рахунок користувача можна видавати n документів n<5

3.2. Можливість переглядати яку які документи взяв конкретний користувач

3.3. Можливість по заданому документу визначити, чи він знаходиться у бібліотеці. Якщо документ виданий, то котрому користувачеві.

3.4. Можливість повернути книжку в бібліотеку

4. Пошук

4.1. Можливість пошуку по ключовому слову серед документів

4.2. Можливість пошуку по ключовому слову серед користувачів

# Теоретичні відомості

## Трирівнева архітектура

Трирівнева n-layer архітектура – схема логічної організації певної програмної системи, яка складається з трьох частин:

* Presentation layer (рівень представлення) – це той рівень, з яким безпосередньо взаємодіє користувач. Містить компоненти користувацького інтерфейсу та механізм отримання введення від користувача.
* Business layer (рівень бізнес-логіки) – містить набір компонентів, що відповідають за обробку отриманих з рівня представлення даних, реалізує всю логіку застосунку, розрахунки, взаємодіє з базою даних і передає рівню представлення результати обробки.
* Data Access Layer або рівень доступу до даних зазвичай містить моделі усіх даних, що зберігаються в базі, а також класи, з допомогою яких здійснюється взаємодія з бд.

## MVC

Платформа ASP.NET MVC є фреймворком для створення сайтів та веб-застосунків за допомогою реалізації патерна MVC.

Концепція патерна MVC (model - view - controller) передбачає поділ застосунку на три компоненти:

* Контроллер (controller) є класом, що забезпечує зв’язок між користувачем та системою, представленням і базою даних. Він отримує введені користувачем дані та обробляє їх. Залежно від результатів обробки контролер надсилає користувачу певний висновок, наприклад, у вигляді представлення.
* Представлення (view) є візуальною частиною або користувацьким інтерфейсом. Зазвичай це html-сторінка.
* Модель (model) є класом, що описує логіку використовуваних даних. Схему патерна відтворює Рисунок 2.1.

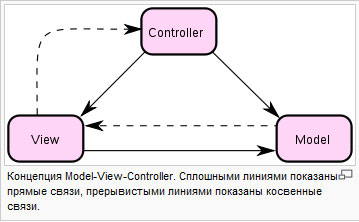
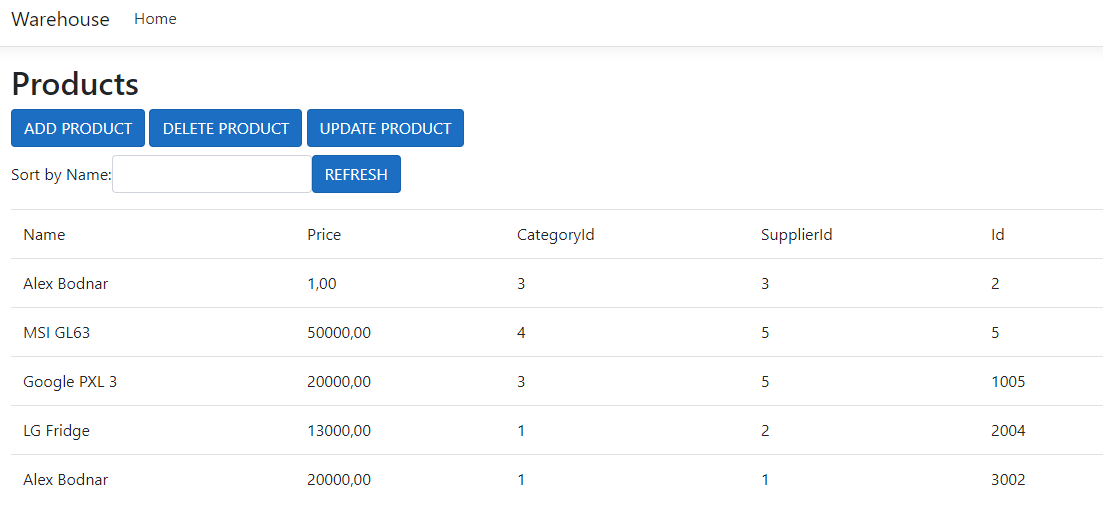
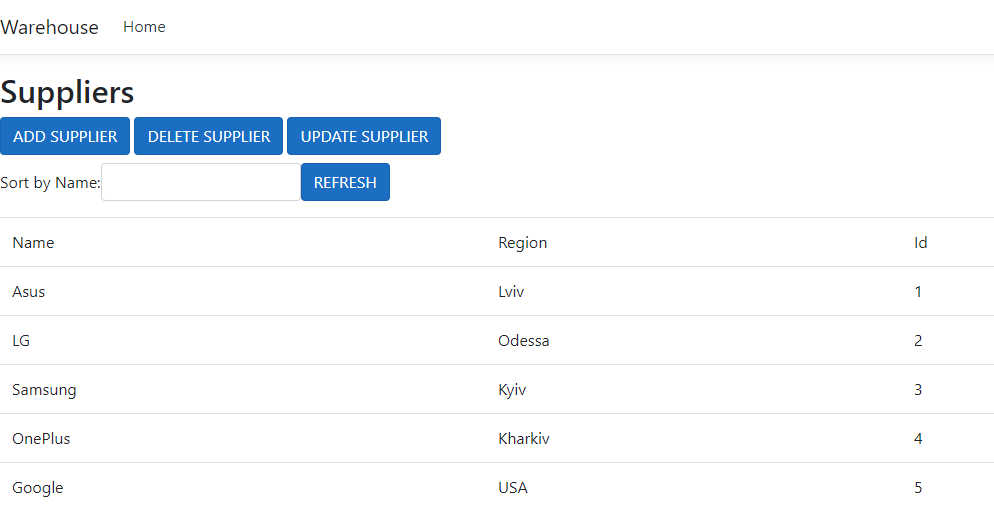
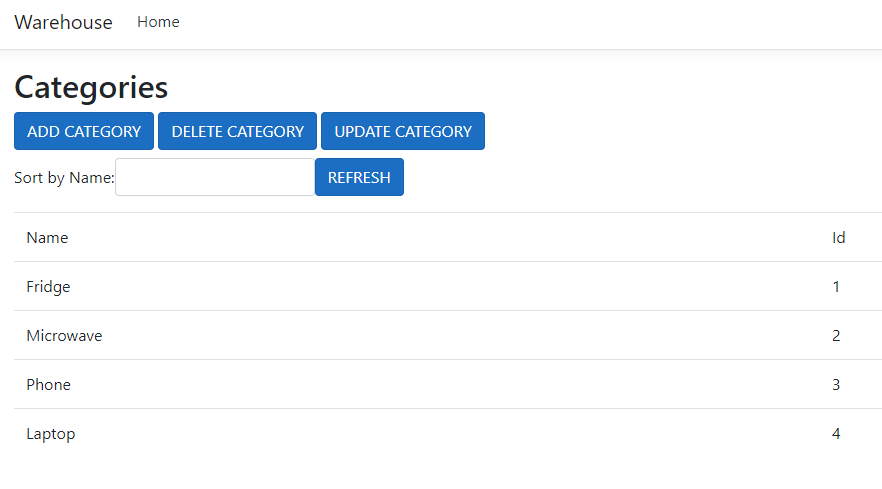


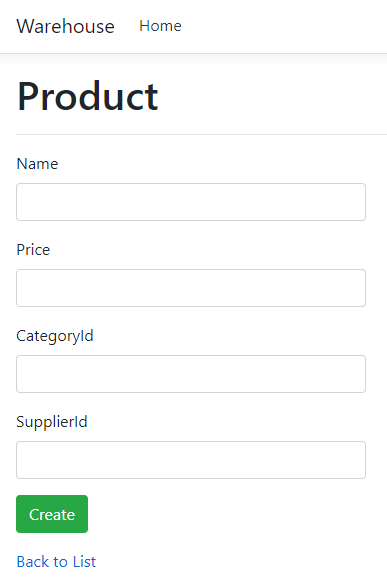
Рисунок 2.1 - Схема патерна MVC

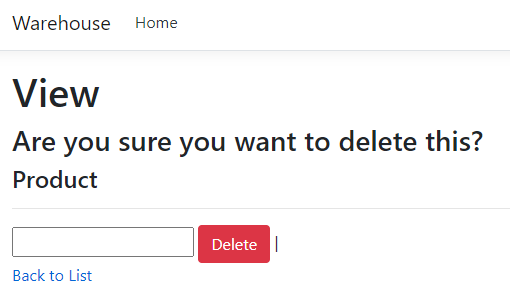
# ЗАГАЛЬНИЙ вигляд ПРОЕКТУ

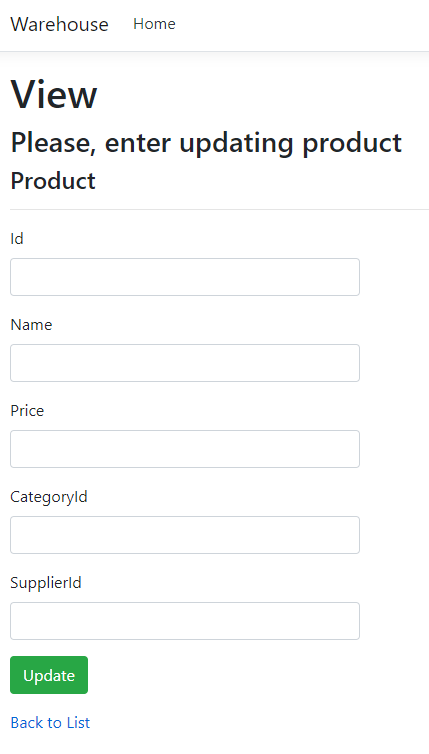












Висновки

Під час розглядання теоретичного матеріалу з теми даної курсової роботи було досліджено принципи роботи з трирівневою n-layer архітектурою та шаблоном MVC.

Протягом розробки ПЗ були отримані навички з застосування архітектури та патерну на практиці, а також був повторений і узагальнений весь навчальний матеріал, що викладався протягом семестру.

Окрім того, було розроблено структуру БД, що відповідала б вимогам поставленої задачі. Було реалізовано взаємодію веб-застосунку з базою даних засобами Entity Framework. Аби частини проекту були максимально гнучкими, використовувався Dependency Injection на базі NuGet пакету Unity.

Під час розробки застосунку було використано трирівневу архітектуру: рівень доступу до даних, що відповідає за взаємодію з БД, рівень бізнес-логіки, що реалізує усі підрахунки та обробку інформації, необхідні для функціонування застосунку, а також рівень представлень, що забезпечує взаємодію з користувачем.

Результатом даного курсового проекту є веб-застосунок для користування студентською бібліотекою. Увесь застосунок або його частини можуть бути використані у сфері освіти для створення сайту реєстрації студентів університету та полегшення збору інформації про певні документи бібліотеки.

Перелік посилань

1. Повне керівництво ASP.NET MVC 5 [Електронний ресурс]: Metanit – Адреса інтернет-сторінки: <https://metanit.com/sharp/mvc5/>
2. Багаторівнева архітектура [Електронний ресурс]: Metanit – Адреса інтернет-сторінки: <https://uk.wikipedia.org/wiki/Класифікація>\_комп’ютерних\_ігор .