КУРСОВА РОБОТА

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ З WEB-ІНТЕРФЕЙСОМ НА ПЛАТФОРМІ .NET

Завдання

1. Спроектувати Web-застосування у відповідності з принципами багатошарової архітектури програмних систем та скласти проектну документацію.
   - 1.1. Описати загальну архітектуру застосування, призначення її шарів та зв'язки між шарами (рівнями).
   - 1.2. Вихідний код програмної системи представляє сукупність проектів різного типу, поєднаних одним рішенням. Описи типів розташовуються в окремих файлах. При необхідності бажано створити додаткові папки проекту, щоб код проекту був логічно структурований.
   - 1.3. Представити діаграми класів кожного рівня, використовуючи UML-нотації.

2. Розробити застосування на мові C#, яке відповідає вимогам у варіанті. Відокремити рівні доступу до даних, бізнес-логіки та представлення.
   - 2.1. Верхній рівень – представлення (UI), призначений для взаємодії з користувачем. Реалізувати окремим проектом. Для створення Web-інтерфейсу використати технологію ASP.NET WebAPI. На рівні UI повинні бути тільки операції взаємодії з користувачем. Код UI повинний бути максимально простим, неперевантаженим великою кількістю операцій. Ця частина системи (Front End) може бути реалізована будь-яким способом (для стилістичного оформлення допускається використання будь-яких фреймворків та бібліотек, наприклад, Bootstrap). Дані, з якими працює рівень представлення, повинні зберігатись в окремих моделях цього рівня (маються на увазі власні класи (типи) рівня, не запозичені з інших рівнів). Передбачити контроль\перевірку даних, введених користувачем (обов'язково заповнені текстові поля, довжина введених даних, тощо).
   - 2.2. Проміжний рівень – бізнес-логіка, реалізований як динамічна бібліотека. На цьому рівні реалізується саме функціонал застосування, описаний у варіанті. Доречним буде застосувати відомі принципи та шаблони проектування. Для виконання операцій бізнес-логіки передбачити перевірку виняткових ситуацій - виключень. При необхідності створити власні класи виключень.
   - 2.3. Нижній рівень – шар доступу до даних у вигляді бібліотеки. Збереження даних програмної системи виконується у реляційній БД під керуванням СУБД MS SQL. Для взаємодії зі сховищем даних використати ORM ADO .Net Entity Framework (code first). Доступ до даних для шару бізнес-логіки організувати через репозиторії, поєднані у одиницю роботи (їм відповідають шаблони проектування Repository та Unit of Work (UoF). Репозиторії надають доступ до набору сутностей (entities) певного типу. Одиниця роботи (UoW) є точкою єдиного доступу до репозиторіїв та контексту Entity Framework.

3. 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).

4. При необхідності для більшої ізоляції основних рівнів можуть вводитися додаткові рівні (наприклад, винесення Repository та UoW).

5. Для ізоляції рівнів використати DI (Ninject чи Autofac).

6. Створити модульні тести для перевірки працездатності коду
   - 6.1. За допомогою бібліотек NUnit чи XUnit написати модульні тести до бізнес логіки. Мінімальне покриття тестами – 50%. Покриття продемонструвати відповідними засобами, наприклад AxoCover, CodeCoverage та ін. Модульні тести повинні бути окремим проектом в рішенні.

6.2. Для оформлення коду модульних тестів овоб'язково застосовувати принцип Triple A.
6.3. За допомогою DI забезпечити, щоб тести для перевірки методів роботи з даними, ніяким чином не впливали на ці дані, використавши для даних об'єкти-емулятори чи замінювачі реальних даних (mock, stub).

7. Для визначення об'єму робіт та декомпозиції задач використати систему розподілу (відслідковування) задач. Це може бути Visual Studio Online/ Team Foundation Server online, Jira, Trello чи інша система. Для комплексної ітеративної розробки окремих компонентів застосування та спільного їх збору використовувати Visual Studio Online/ Team Foundation Server online або/ і GitHub.

8. Діаграма(-и) та вихідний код повинні відповідати основним принципам проектування: ООР, SOLID, Law of Demeter (LoD), DRY, YAGNI, KISS, cohesion – coupling, inheritance with caution. За невідповідність принципам оцінка за курсову може бути знижена.

9. При написанні коду притримуватися C#/.NET Code Conventions та кращих практик написання коду: іменування класів, об'єктів, властивостей, методів, інші назви повинні бути зрозумілими та відповідати їх задачам, форматувати код, не використовувати magic numbers, а також ставити мінімально необхідний модифікатор доступу (public повинен бути обґрунтований, а не використаний для всіх членів класу та класів проектів без виключення). За неохайне оформлення коду можливе зниження оцінки.

10. Зміст курсової роботи:
10.1. Титульний лист
10.2. Зміст
10.3. Загальне завдання та варіант
10.4. Загальна архітектура проекту з поясненням
10.5. Описання процесу розробки з використанням системи розподілу версій та відслідковування завдань
10.6. Діаграми кожного рівня з поясненнями
10.7. Важливі деталі реалізації проекту
10.8. Інструкції з використання застосування
10.9. Висновки
10.10. Використані джерела

11. **Вихідний код** додавати в пояснювальну записку курсової роботи **не потрібно**. Виключення можуть бути тільки для пояснення важливих деталей реалізації, але це можуть бути тільки невеликі фрагменти коду в кілька рядків.

12. Виконувати один варіант повинно не менше 2-х студентів з однієї підгрупи та не більше 3-х. В підгрупі варіанти не повинні повторюватися.

13. Під час захисту КР продемонструвати репозиторій системи контролю версій, пояснити стратегію створення гілок. А також показати систему розподілу задач та пояснити розподіл задач та процес їх виконання.

Варіанти

Номер варіанту	Предметна область	Вимоги та компоненти
1	Інтернет-аукціон	Складається з підсистем: 1) підсистема додавання/ редагування лотів, 2) підсистема перегляду лотів (з пошуком, фільтрами), 3) підсистема торгів. Кожний лот повинен входити до певної категорії чи підкатегорії. Ступінь вкладеності може бути довільний. Ролі користувачів: адміністратор, менеджер, зареєстрований, незареєстрований. Менеджер підтверджує лот та проводить торги. Незареєстрований користувач може тільки проглядати лоти, а зареєстрований – додавати та брати участь в аукціоні.
2	Туристична агенція	Складається з підсистем: 1) підсистема додавання/ редагування турів, 2) підсистема виводу (гарячі тури, екскурсійні тури, країни/ регіони, по типу і т.д. – з фільтрами, пошуком), 3) підсистема бронювання туру, 4) підсистема бронювання квитка на певний вид транспорту чи номер во готелі. Ролі користувачів: адміністратор, менеджер, зареєстрований, незареєстрований. Менеджер може змінювати інформацію про тури. Зареєстрований користувач – бронювати. Незареєстрований – тільки переглядати.
3	Інтернет-магазин	Складається з підсистем: 1) підсистема додавання/ редагування товарів, 2) підсистема виводу товарів (з сортуванням та фільтрами, пошуком), 3) підсистема замовлення, 4) підсистема статистики. Ролі користувачів: адміністратор, менеджер, зареєстрований, незареєстрований. Менеджер може змінювати інформацію про товари, обробляти замовлення. Зареєстрований користувач – замовляти. Незареєстрований – тільки переглядати.
4	Електронний розклад	Складається з підсистем: 1) підсистема створення розкладу (аудиторії, групи, викладачі, дисципліни), 2) підсистема виведення розкладу (аудиторії, групи, викладачі, підрозділи), 3) підсистема статистики. Ролі користувачів: адміністратор, редактор, управлінський склад, викладач, студент. Редактор може змінювати інформацію про розклад. Управлінський склад може переглядати увесь розклад по кожному зі фільтрів (аудиторії, групи, викладачі, дисципліни). Викладачі можуть переглядати свій розклад та розклад будь-яких груп. Студент може переглядати тільки розклад груп. Неавторизований користувач не повинен мати доступу до розкладу.
5	База резюме та вакансій	Складається з підсистем: 1) підсистема додавання/ редагування резюме та вакансій, 2) підсистема виводу (з сортуванням та фільтрами, пошуком), 3) підсистема подачі резюме на вакансію чи пропозиції вакансії по резюме/ перегляд «прив'язаних» вакансій резюме. Ролі користувачів: адміністратор, роботодавець/ рекрутер, працівник. Роботодавець/ рекрутер може додавати вакансії, шукати резюме за вакансією. Працівник – додавати резюме, шукати вакансію за резюме. Неавторизований користувач не повинен мати доступу до вакансій та резюме.

6	Рекламна агенція	Складається з підсистем: 1) підсистема додавання/ редагування послуг, 2) підсистема виводу (знижки на послуги, швидкі замовлення, різні типи послуг – з фільтрами, пошуком), 3) підсистема замовити послугу, 4) підсистема роботи з поліграфією. Ролі користувачів: адміністратор, менеджер, зареєстрований, незареєстрований. Менеджер може змінювати інформацію про послуги. Зареєстрований користувач – замовляти. Незареєстрований – тільки переглядати.