

Вимоги до виконання проектів з Програмної інженерії

1. Формулювання проектів

- Утворення команди з 3-5 людей
- Вибір теми проекту (власна тема або пов'язана з університетом)
- Створення репозиторію на github
- Короткий опис загальних особливостей проекту усно (за бажанням з використанням презентації)
- Ознайомлення з паттерном MVC та його реалізацією в рамках ASP .NET

2. Формування переліку вимог до проекту. Специфікація та валідація вимог. Use-case модель.

- Створення початкової документації проекту вибраною мовою (українською, англійською)
- Наявність назви та опису проекту
- Наявність таблиці або списку ролей системи із описом кожної з них
- Виділення підсистем системи
- Наявність таблиці із use-case та їх коротким описом для кожної підсистеми та ролі.
- Наявність use-case діаграм, що відповідають вищезазначеній таблиці із використанням UML
- Коректність зазначених ролей, підсистем та відповідних use-case
- Повнота системи в сенсі наявності усіх основних use-case
- Коректне форматування документу
- Наявність звіту про виконану роботу

3. Завершення формулювання вимог. Розробка моделі предметної області.

- Виправлення помилкових use-case
- Доповнення системи новими use-case
- Побудова діаграми послідовностей
- Побудова діаграми сутностей
- Побудова мокапів системи (зображення основних сторінок системи без деталізації)
- Наявність звіту про виконану роботу

4. Розробка та затвердження архітектури проекту.

- Знання та вибір підходу до побудови архітектури проекту
 - Layered architecture
 - Hexagonal architecture
 - Onion architecture
 - Clean architecture
- Побудова базових архітектурних структур в рамках вибраного архітектурного підходу та платформи ASP .NET Core MVC.
- Створення бази даних
 - MS SQL Server
 - MySQL
 - PostgreSQL
 - MongoDB

- Підключення Entity Framework або інших ORM для роботи з базою даних.
- Налаштування одного чи декількох аналізаторів коду
 - Sonar
 - StyleCop
 - Roslynator
- Налаштування структурного логування з використанням бібліотеки SeriLog та інструменту Seq (або інших).
- Наявність звіту про виконану роботу

5. Реалізація use-case в рамках MVC.

- Написання контролерів (controller) для обробки запиту користувача, що відповідає use-case
- Написання класів сутностей та сервісів (model) для виконання бізнес-логіки
 - Дотримання правил написання “чистого” коду
 - Логування дій користувача та помилок
- Написання класів (репозиторіїв) для отримання чи зміни даних в базі даних
- Створення шаблонів відображень (view) для виконання use-case користувачем з використанням Razor view engine у ASP .NET Core MVC
- Написання юніт-тестів з використанням xUnit або NUnit
 - Покриття тестами програмного методу, що відповідає use-case
 - Підбір параметрів методу для врахування різних сценаріїв його виконання (як позитивних, так і негативних)
 - Відповідність найкращим практикам написання юніт тестів
- Написання інтеграційних тестів (за бажанням)
- Наявність звіту про виконану роботу

6. Аутентифікація та авторизація з використанням ASP .NET Identity

- Підключення ASP .NET Identity до проекту
- Заповнення бази даних ролями відповідно до документації
- Розширення моделі користувача із ASP .NET Identity для зберігання додаткової інформації
- Налаштування аутентифікації на основі кук
- Налаштування авторизації на основі ролей
- Реалізація реєстрації користувача
 - Можливість введення необхідних даних та їх валідація
 - Коректне збереження даних
 - Підтвердження реєстрації (відправка email або підтвердження Адміністратором)
- Реалізація логіну
 - Введення та валідація імені користувача (email) та паролю
- Реалізація відновлення паролю
 - Можливість отримати посилання для відновлення паролю
 - Можливість введення нового паролю та оновлення його в базі.
- Визначення доступу до реалізованих use-case на основі ролі користувача
- Наявність звіту про виконану роботу

7. Вдосконалення UX/UI

- Виділення проблемних місць при взаємодії користувача із системою
- Використання можливостей Razor для налаштування UI
- Використання можливостей UI бібліотек для задання додаткових css стилів html елементів сторінки.
- Використання javascript та jQuery для додаткових дій над елементами сторінки
- Розбиття та об'єднання css стилів та javascript коду з використанням bundle в ASP .NET Core MVC
- Можливість зручного використання системи в цілому з точки зору UX
- Наявність звіту про виконану роботу

8. Підсумкове заняття

- Демонстрація скомпільованого та опублікованого проекту (у файловій системі або хмарному сховищі) з використанням
 - Production налаштувань
 - Режиму release
- Наявність коректних реалізацій основних use-case
 - Відповідність побудованої системи вимогам, зазначеним у документації
 - Відсутність необроблених помилок, які призводять до неможливості виконання use-case