|  |  |  |  |
| --- | --- | --- | --- |
| МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  **Кафедра програмних систем і технологій**  Дисципліна  **«Архітектура та проектування програмного забезпечення»**  **Лабораторна робота № 7**  «Дослідження шаблонів проектування»  **на тему:**  **«**Онлайн клієнт для погодних даних**»** | | | |
| **Виконав:** | Гоша Давід | **Перевірив**: | Берестов Д.С |
| Група | ІПЗ-23 | Дата перевірки |  |
| Форма навчання | денна | Оцінка |  |
| Спеціальність | 121 |
| 2022 | | | |

# Вступ

Зміст

[Вступ 2](#_Toc101629728)

[Завдання 2](#_Toc101629729)

[Вимоги, висунуті до системи: 2](#_Toc101629730)

[Вибір компонентів 2](#_Toc101629731)

[Технології які використовувалися: 3](#_Toc101629732)

[Вибір архітектурного стилю 3](#_Toc101629733)

[Основні проекті рішення 3](#_Toc101629734)

[Вибір з’єднувачів 3](#_Toc101629735)

[Опис архітектури системи 3](#_Toc101629736)

[Проблеми, з якими зіткнувся в процесі реалізації 3](#_Toc101629737)

[Архітектурний аналіз 3](#_Toc101629738)

[Початок програмної реалізації 3](#_Toc101629739)

[Архітектурні рішення 3](#_Toc101629740)

[Модель розгортання 3](#_Toc101629741)

[Кінцевий етап розробки 3](#_Toc101629742)

[Висновок 3](#_Toc101629743)

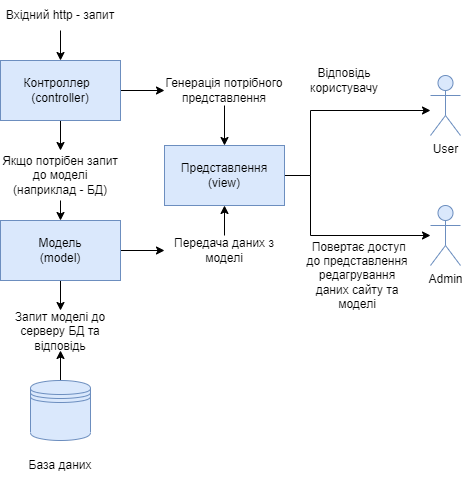
**Мета практикуму** – отримати практичні навички застосування шаблонів проектування для підвищення внутрішньої якості ПЗ.

**Тема проекту** – «Онлайн клієнт для погодних даних».

**Застосування проекту** - проект розробляється та проектується з метою публікації прогнозів погодних даних.

# Завдання

**Створення ПЗ,** яке буде приймати погодні дані , передавати їх до нашого серверу, який буде містити сайт (графічний інтерфейс) та базу даних, та виводитиме їх у спеціальних блоках сайту.

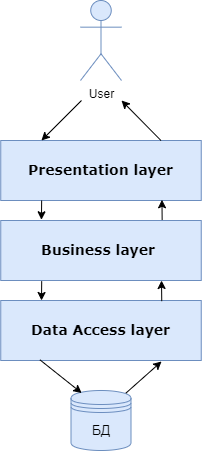


# Вимоги, висунуті до системи:

1. Опис застосування, що розробляється з точки зору користувача.
   1. Інтерфейс , що друкує інформацію про поточний стан погоди. А саме: 1.1.1.
      1. Температура (Погодинно).
      2. Швидкість вітру.
      3. Відносна вологість.
      4. Видимість.
      5. Стан опадів (Йде сніг, пасмурно, тощо).
2. **Опис основних функціональних вимог:** 
   1. Отримання даних з серверу метеорологічного центру. 2.2.
   2. Сортування та нормалізація даних. 2.3.
   3. Створення візуального зображення погодних умов. 3.
3. **Опис основних нефункціональних вимог:**  
   Не функціональні вимоги можна поділити на дві категорії: покращення (безпека, надійність, швидкодія, зручність у використанні ...) та вдосконалення (маштабування, відновлюваність ...) властивостей системи.
   1. Вимоги до Інтерфейсу (**Interface Requirements**).
      1. Апаратні Інтерфейси (**Hardware Interfaces**) Апаратні інтерфейси необхідні для підтримки системи, включаючи логічну структуру, фізичні адреси і очікувану поведінку.
      2. Інтерфейси ПЗ (**Software Interfaces**) Назви Інтерфейсів програмного забезпечення з якими аплікація повинна взаємодіяти.
      3. Звязки Інтерфейсів (**Communications Interfaces**) Звязки інтерфейсу з іншими системами або приладами.
   2. Апаратні та Програмні Вимоги (**Hardware/Software Requirements**):  
      Опис апаратної та програмної платформ, необхідних для підтримки системи.
   3. **Operational Requirements:**
      1. Безпека та Конфіденційність (**Security and Privacy**).
      2. Надійність (**Reliability**).
      3. Відновлювальність (**Recoverability**).
      4. Продуктивність (**Performance**).
      5. Потенціал (**Capacity**).
      6. Збереження даних (**Data Retention**).
      7. Керування помилками (**Error Handling**).
      8. Правила Перевірки (**Validation Rules**).

# Вибір компонентів

Будемо використовувати багаторівневу архітектуру (На діаграмі зображено саме логічні шари архітектури).

1. Presentation layer (рівень представлення): це той рівень, з яким безпосередньо взаємодіє користувач. Цей рівень включає компоненти інтерфейсу користувача, механізм отримання , введення від користувача. Стосовно asp.net mvc на даному рівні розташовані уявлення і всі ті компоненти, які складають інтерфейс користувача (стилі, статичні сторінки html, javascript), а також моделі уявлень, контролери, об'єкти контексту запиту.
2. Business layer (рівень бізнес-логіки): містить набір компонентів, що відповідають за обробку отриманих від рівня уявлень даних, реалізує всю необхідну логіку програми, всі обчислення, взаємодіє з базою даних та передає рівню представлення результат обробки.
3. Data Access layer (рівень доступу до даних): зберігає моделі, що описують використовувані сутності, також розміщуються специфічні класи для роботи з різними технологіями доступу до даних, наприклад, клас контексту даних Entity Framework. Тут також зберігаються репозиторії, якими рівень бізнес-логіки взаємодіє з базою даних.
4. ASP.NET identity: Вбудована технологія для аутинтифікації та авторизації користувачів. Потрібна для делегування обов’язків та редагування даних у системі.
5. Entity Framework: представляє спеціальну об'єктноорієнтовану технологію на базі фреймворку .NET для роботи з даними. Слугує сервісом для зв’язку та управління даними з БД. 5.

# Технології які використовувалися

1. Середовище розробки Visual Studio 2022
2. Сервер бд MS SQL Server 2019
3. Мова прогрумування C#
4. Entity Framework Core + Migrations + Identity
5. Тип застосунку ASP.NET Core MVC 3.1 5.6. HTML5 + SASS (CSS)
6. JavaScript + jQuery

# Вибір архітектурного стилю

# Основні проекті рішення

# Вибір з’єднувачів

# Опис архітектури системи

# Проблеми, з якими зіткнувся в процесі реалізації

# Архітектурний аналіз

# Початок програмної реалізації

# Архітектурні рішення

# Модель розгортання

# Кінцевий етап розробки

# Висновок

В ході даної лабораторної роботи був реалізований застосунок шляхом реалізації компонентів та проемно-орієнтованих компонентів. Побудовано архітектурну модель розгортання ПЗ, яка підтримує реалізацію необхідної для вашого проекту якості сервісу, виконано аналіз моделі .Реалізовано розгортання розробленого застосунку, підготовано його до демонстрації. У ході роботи, обов’язково продокументовано будь-які зміни архітектури застосування.