Projekt końcowy — specyfikacja

Czas: 2 × 45 min | praca zespołowa (2–3 osoby) | Wykład 2.30

Cel: W zespole przygotujecie specyfikację aplikacji MVVM z bazą danych: wymagania, makiety, architekturę warstw, plan testów i podział pracy w Git.

Po zajęciach potrafisz:

- formułuje sprawdzalne wymagania zespołowego projektu
- projektuje architekturę: widoki, ViewModele, usługi, repozytorium, model danych
- planuje pracę zespołu w zgłoszeniach i gałęziach Git
- planuje testy jednostkowe i ręczne

Sprzęt i materiały

- komputer z Windows 11 i dostępem do internetu
- pendrive lub konto w repozytorium do zapisania projektu

Oprogramowanie

- **Visual Studio Community 2026 (obciążenie „Programowanie aplikacji klasycznych platformy .NET")** — darmowe dla uczniów i nauczycieli; w pracowni zainstalowane
  visualstudio.microsoft.com/pl/downloads
- **.NET 10 SDK (LTS)** — instalowany razem z Visual Studio; sprawdzenie: `dotnet --version`
  dotnet.microsoft.com/pl-pl/download/dotnet/10.0
- **Git for Windows** — lub Git wbudowany w Visual Studio
  git-scm.com/downloads/win
- **Konto GitHub (bezpłatne)** — lub szkolne konto GitHub Education
  github.com/signup
- **Narzędzie do makiet (Penpot, Figma lub papier)**
  penpot.app

Bezpieczeństwo

- Zapisuj projekt w swoim folderze (np. `C:\Users\<login>\source\repos`), a na koniec zajęć skopiuj go na pendrive lub wyślij do repozytorium.
- Nie uruchamiaj programów i pakietów z nieznanych źródeł; pakiety NuGet instaluj tylko te, które wskazuje zadanie.

Przebieg zadania

1 Temat i wymagania

Zatwierdźcie temat z nauczycielem.

1. Wymagania obowiązkowe: WPF lub MAUI w MVVM (CommunityToolkit.Mvvm), baza SQLite (EF Core lub sqlite-net), min. 3 widoki/strony, walidacja, DI, testy ViewModeli, publikacja.
2. 10–15 ponumerowanych wymagań funkcjonalnych i 3 pozafunkcjonalne (wydajność, dostępność, motyw).
3. Przykładowe tematy: wypożyczalnia sprzętu szkolnego, dziennik treningów, rezerwacja sal, budżet domowy z kategoriami, quiz z pytaniami w bazie.

2 Architektura

Dokument w folderze `docs`.

1. Diagram encji bazy (encje, relacje 1:N/N:M, klucze).
2. Lista widoków i ViewModeli z komendami; interfejsy usług (repozytorium, dialogi, ustawienia).
3. Makiety każdego widoku z nazwami kontrolek i przejściami.

3 Organizacja

Repozytorium zespołu.

1. Repozytorium na GitHub z ochroną main, szablonem pull requesta i .gitignore.
2. Zgłoszenia (Issues) dla wymagań z przypisaniem osób; tablica projektu (Projects) z kolumnami Do zrobienia / W toku / Gotowe.
3. Harmonogram do zadań 33 i 34; plan co najmniej 10 testów jednostkowych.
4. Przedstawcie specyfikację nauczycielowi (5 minut) i wprowadźcie poprawki.

W sprawozdaniu umieść

- ☐ dokument specyfikacji (wymagania, diagram encji, architektura, makiety)
- ☐ adres repozytorium i zrzut tablicy projektu
- ☐ podział pracy

Pytania kontrolne

1. Które wymaganie jest największym ryzykiem i jak je ograniczycie?
2. Jak podzieliliście pracę, aby uniknąć konfliktów w Git?
3. Które klasy będą testowane jednostkowo?

Kryteria oceny

Kryterium	Punkty
Wymagania	5
Model danych i architektura	6
Makiety	4
Organizacja repozytorium i plan	5
Razem	**20**

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).