

Inżynieria oprogramowania - projekt - EF-ZI

Cel: zaprojektowanie aplikacji obsługującej relacyjną bazę danych firmy/institucji itp.

Projekt jest dwuosobowy. Jeśli w grupie projektowej jest nieparzysta liczba osób, jedna osoba powinna wykonać projekt sama bądź ze studentem z innej grupy.

Przed upływem określonych terminów należy wysłać maila na adres: dawwar@prz.edu.pl z wykonanym kolejnym etapem projektu w załączniku (format *doc*, *docx*, *pdf* lub *odt*; proszę nie wysyłać projektu w treści maila).

Format tematu e-maili: *io_inicjaly_etap_projektu*.

Format nazw plików z projektem: *nazwisko1, nazwisko2.format_pliku*.

Przykładowo, jeśli drugi etap projektu wysyłają Jan Kowalski i Tomasz Nowak, to temat maila powinien wyglądać następująco: *io_JKTN_etap_2*, natomiast plik powinien nazywać się: *Kowalski, Nowak.pdf* (zakładając, że jest on w formacie PDF).

Kolejne etapy projektu wysyłamy w osobnych plikach (nie dopisujemy do poprzednich etapów w tym samym pliku).

Kolejne etapy projektu:

I. **Termin: 16.05.2026. Należy określić skład osobowy, przygotować temat i krótki opis projektu.** W temacie należy określić rodzaj firmy/institucji (np. szkoła średnia/wyższa, hotel). Opis powinien zawierać listę grup użytkowników (co najmniej dwóch), którzy mogą korzystać z systemu oraz ich role/uprawnienia. Tematy projektów w jednej grupie nie mogą się powtarzać.

II. **Termin: 30.05.2026. Należy przygotować diagram związków encji (ERD) oraz diagram przypadków użycia (UCD) z języka UML.**

Diagram ERD powinien zawierać co najmniej sześć encji wraz z atrybutami, ich typami (z języka SQL, np. INT, DATE) i oznaczonymi kluczami głównymi i obcymi. Diagram powinien być znormalizowany, a więc nie mogą w nim występować relacje typu wiele-do-wielu (należy je zamienić na relacje jeden-do-wielu wykorzystując encje pomocnicze). Opis tworzenia ERD znajduje się np. na stronie z materiałami:

<http://materialy.prz-rzeszow.pl/materialy.php?przedmiot=13> (login: student, hasło: kaiwww) - wykład „Diagram związków encji (ERD)” i inne wykłady związane z ERD. Pomocne mogą również okazać się poniższe strony (oraz wiele innych, ponieważ ERD, to bardzo powszechnie stosowany diagram i jest o nim wiele informacji w Internecie):

- <https://technikinformatyk.pl/programowanie/bazy-danych-diagram-erd>
- <https://miro.com/pl/diagramming/what-is-an-er-diagram>

Diagram UCD powinien zawierać przypadki użycia z zależnościami rozszerzania i zawierania (nie wystarczy tylko jeden typ zależności).

Dokładna charakterystyka UCD znajduje się w książce: *Język UML 2.0 w modelowaniu systemów informatycznych*, Wrycza S., Marcinkowski B., Wyrzykowski K. (rozdziały 2 i 3). Diagram jest również dobrze scharakteryzowany na stronach:

- <https://wolski.pro/diagramy-uml/diagram-przypadkw-uzycia>
- <https://it-consulting.pl/2022/04/03/diagram-przypadkow-uzycia>

Dla osób, którym łatwiej uczyć się z kursów wideo (obraz + komentarz) polecam ten kurs do zrozumienia diagramu UCD (jest w języku angielskim, ale można włączyć napisy automatyczne tłumaczone na język polski):

- <https://www.youtube.com/watch?v=4emxjxonNRI>

III. Termin: 13.06.2026. Należy przygotować macierz CRUD (Create, Retrieve, Update, Delete) oraz diagram przepływu danych (DFD).

Macierz CRUD powinna określać jak operacje (funkcje, procesy) wpływają na określone tabele bazy danych (encje ERD). Diagram DFD powinien składać się z diagramu ogólnego i diagramów szczegółowych. Opis tworzenia CRUD i DFD znajduje się np. na stronie z materiałami: <http://materialy.prz-rzeszow.pl/materialy.php?przedmiot=13> - wykład/„Modelowanie procesów – DFD”. Diagramy DFD są również scharakteryzowane na stronie pod adresem:

<https://dawwar.v.prz.edu.pl/materialy-do-pobrania/inzynieria-oprogramowania-ef-zi-3-6.html> (plik „DIAGRAM_PRZEPLYWU_DANYCH.PDF”, opracowanie dr. inż. Jacka Jakięły). W przypadku wystąpienia trudności w rysowaniu DFD metodami z wyżej wymienionych źródeł można przyjąć inną konwencję graficzną. Ważne, żeby istota diagramu została zachowana i żeby różne bloki składowe miały własne symbole, tak aby dało się je odróżnić.

IV. Termin: 20.06.2026. Należy przygotować diagram klas z języka UML oraz jeden dowolny diagram wchodzący w skład języka UML, który autorzy uznają za odpowiedni w kontekście projektowanej aplikacji.

Diagram klas powinien być na poziomie implementacyjnym, tzn. każda klasa powinna być reprezentowana przez jej nazwę, zestaw pól i metod. Opracowując diagram klas należy zwrócić uwagę na rodzaje relacji pomiędzy klasami. Relacje, które trzeba rozumieć i odróżniać, to: asocjacja, agregacja częściowa, agregacja całkowita (silna, kompozycja), uogólnienie (dziedziczenie, generalizacja). Przy relacjach agregacji powinny być wpisane krotności.

Dokładna charakterystyka diagramu klas znajduje się w książce: *Język UML 2.0 w modelowaniu systemów informatycznych*, Wrycza S., Marcinkowski B., Wyrzykowski K. (rozdziały 2 i 3). Diagram jest również dobrze scharakteryzowany na stronach:

- <https://devmentor.pl/b/podstawy-uml-diagramy-klas>
- <https://wolski.pro/diagramy-uml/diagram-klas>

Dla osób, którym łatwiej uczyć się z kursów wideo (obraz + komentarz) polecam ten kurs do zrozumienia diagramu klas (są w języku angielskim, ale można włączyć napisy automatyczne tłumaczone na język polski):

- <https://www.youtube.com/watch?v=6XrL5jXmTwM>

W przypadku nie wysłania kolejnego etapu projektu w wyznaczonym terminie, maksymalna ocena za projekt zostaje obniżona o $x \cdot 0,5$ stopnia, gdzie x jest liczbą niewysłanych etapów.

Jeśli autor chce wykonać poprawki do wcześniej wysłanego etapu projektu (pomimo, że prowadzący nie zgłosił żadnych uwag), może je zrobić i wysłać razem z kolejnym etapem oznajmiając o tym prowadzącemu w mailu. Należy w takim wypadku napisać jakie zmiany zostały wprowadzone.

Przykładowe narzędzia, które umożliwiają tworzenie diagramów:

- Draw.io (<https://www.draw.io>)
- Lucidchart
- Dbdiagram.io (<https://dbdiagram.io>) (ERD)
- MySQL Workbench (ERD)
- Visual Paradigm Online
- Violet UML Editor
- StarUML
- UMLet