

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_inicjały_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: 17.05.2025. 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: 24.05.2025. Należy przygotować diagram związków encji (ERD) i diagram hierarchii funkcji (FHD).**

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>

Charakterystyka FHD znajduje się pod adresem <https://dawwar.v.prz.edu.pl/materialy-do-pobrania/inzynieria-oprogramowania-ef-zi-3-6.html> (plik „DIAGRAM_HIERARCHII_FUNKCJI.PDF”, opracowanie dr. inż. Jacka Jakięły). Przykładowy diagram FHD można podejrzeć pod adresem: <https://slideplayer.pl/slide/830061/2/images/12/Diagram+hierarchii+funkcji.jpg>

III. Termin: 31.05.2025. Należy przygotować macierz CRUD (Create, Retrieve, Update, Delete) oraz diagram przepływu danych (DFD).

Diagram DFD powinien składać się z diagramu ogólnego i diagramu (lub 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_PRZEPŁYWU_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.

IV. Termin: 14.06.2025. Należy przygotować dwa diagramy wchodzące w skład języka UML: diagram przypadków użycia (UCD) oraz diagram klas.

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

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: asocjacja, agregacja częściowa, agregację całkowitą (silną, kompozycja), uogólnienie (dziedziczenie, generalizacja). Przy relacjach agregacji powinny być wpisane krotności.

Dokładna charakterystyka UCD i 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 klas i UCD są również dobrze scharakteryzowane na stronach:

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

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

- <https://www.youtube.com/watch?v=4emxjxonNRI>
- <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 student 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), Violet UML Editor, StarUML, UMLet, ArgoUML, Oracle Designer.

W razie problemów ze znalezieniem lub uruchomieniem wygodnych narzędzi do tworzenia diagramów można je wykonać w programie MS Word (kształty lub zwykłe tabele dla macierzy CRUD).