

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «Бакалавр»

«Розробка віртуальної платформи вивчення польської мови з застосуванням адаптивної моделі навчання»

(тема кваліфікаційної роботи українською мовою)

«Development of a Virtual Platform for Learning Polish Language Using an Adaptive Learning Model»

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання спеціальності 122 Комп'ютерні науки
(код, назва спеціальності)

Освітня програма Комп'ютерні науки
(назва)

КРИШТАЛЬ Максим Ігорович

(прізвище, ім'я, по-батькові здобувача)

Керівник д.т.н., проф. Казакова Н.Ф. _____
(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент к.т.н, доцент, НУ ОЮА, Щербина Ю.В. _____
(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри
Інформаційних технологій

№ _____ від _____ 2025 р.

Завідувачка кафедри

КАЗАКОВА Надія

(підпис)

(прізвище, ім'я)

Захищено на засіданні ЕК № _____
протокол № _____ від _____ 2025 р.

Оцінка _____ / _____ / _____
(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

КОПИЧЕНКО Іван

(підпис)

(прізвище, ім'я)

Одеса 2025

АНОТАЦІЯ

У кваліфікаційній роботі досліджується тема «Розробка віртуальної платформи вивчення польської мови з застосуванням адаптивної моделі навчання».

Мета роботи – розробка прототипу інтелектуальної навчальної системи, яка автоматично адаптується до рівня знань, потреб і індивідуальних особливостей кожного користувача, забезпечуючи персоналізований підхід до вивчення польської мови.

Об'єкт розробки – процес адаптивного навчання польської мови на основі штучного інтелекту, який підлаштовується під рівень знань студентів або дорослих користувачів.

У процесі роботи було виконано такі основні завдання:

а) проведено аналіз сучасних методів адаптивного навчання та їх використання в системах штучного інтелекту;

б) спроектовано уніфікований діаграмний комплекс, що включає діаграми варіантів використання, станів, компонентів та розгортання;

в) розроблено структуру бази даних та бази знань, що забезпечує ефективне зберігання та управління навчальними матеріалами та даними користувачів;

г) створено графічний інтерфейс користувача у середовищі Visual Studio з урахуванням принципів адаптивного навчання;

д) реалізовано базову логіку роботи системи.

Розроблена платформа закладає основу для подальшого розвитку інноваційних методів адаптивного навчання, що можуть бути використані не лише для вивчення польської мови, а й для інших мовних дисциплін та завдань.

ABSTRACT

The qualification work explores the topic «Development of a virtual platform for learning polish language using an adaptive learning model».

The aim of the study is to develop a prototype of an intelligent learning system that automatically adapts to the knowledge level, needs, and individual characteristics of each user, ensuring a personalized approach to learning Polish.

The object of development is the process of adaptive learning of the Polish language based on artificial intelligence, which adjusts to the knowledge level of students or adult users.

During the work, the following key tasks were completed:

- a) an analysis of modern adaptive learning methods and their use in artificial intelligence systems was conducted;
- b) a unified diagram complex was designed, including use case diagrams, state diagrams, component diagrams, and deployment diagrams;
- c) the structure of the database and knowledge base was developed to ensure efficient storage and management of learning materials and user data;
- d) a graphical user interface was created in the Visual Studio environment, considering the principles of adaptive learning;
- e) the basic logic of the system's operation was implemented.

The developed platform lays the foundation for further development of innovative adaptive learning methods that can be applied not only to learning Polish but also to other language disciplines and tasks.

ЗМІСТ

	Стор.
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ	7
ВСТУП.....	8
1 ПОСТАНОВКА ЗАДАЧІ ВИВЧЕННЯ ПОЛЬСЬКОЇ МОВИ	10
1.1 Діаграма варіантів використання для наочності постановки завдання вивчення польської мови	10
2 ПРОЄКТУВАННЯ УНІФІКОВАНОГО ДІАГРАМНОГО КОМПЛЕКСУ	14
2.1 Побудова діаграми станів	14
2.2 Проєктування діаграми компонентів	17
2.3 Проєктування діаграми розгортання	21
3. ФОРМУВАННЯ БАЗИ ДАНИХ ТА БАЗИ ЗНАНЬ	25
3.1 Бази даних і бази знань	25
3.2 Структурування бази даних	26
3.2.1 Опис сутностей БД	26
3.2.2 Взаємозв'язки між сутностями	27
3.3 Формування ER-діаграми	28
4. СТВОРЕННЯ ПРОТОТИПУ ІНТЕРФЕЙСУ СИСТЕМИ.....	32
4.1 Визначення основних компонентів інтерфейсу.....	32
4.2 Розробка графічного інтерфейсу у Visual Studio	33
4.2.1 Вибір платформи та технології.....	33
4.2.2 Створення нового проєкту у Visual Studio	34
4.2.3 Реалізація логіки на C# у .NET	34
4.2.4 Оптимізація стилю та логіки.....	35
4.2.5 Стартовий код проєкту у Visual Studio.....	35
4.3 Результати розробки прототипу GUI	36
4.3.1 Створення десктопного додатку	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	41

ДОДАТОК А Стартовий код для GUI платформи вивчення польської мови із застосуванням адаптивної моделі навчання.....	43
ДОДАТОК Б Програмний код для дизайну форми польськомовного інтерфейсу навчальної платформи із застосуванням адаптивної моделі навчання.....	45

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

БД – бази даних

БЗ – база знань

ВВ – варіанти використання

ДВВ – діаграма варіантів використання

ДКМ – діаграма компонентів

ДР – діаграма розгортання

ДС – діаграма станів

ШІ – штучний інтелект

AI – Artificial Intelligence

ER-діаграма – Entity-Relationship Diagram

GUI – Graphic User Interface

UI – User Interface

UML – Unified Modeling Language

ВСТУП

У сучасному світі знання польської мови є важливою складовою успішної професійної та особистої діяльності. Польща – одна з ключових європейських країн із динамічною економікою, високим рівнем освіти та широкими можливостями для міжнародної співпраці. Володіння польською мовою відкриває доступ до європейських освітніх програм, сприяє професійному зростанню та розширює можливості міжкультурної комунікації.

Зростання економічних і культурних зв'язків між Україною та Польщею, а також інтеграція в європейський простір обумовлюють необхідність постійного вдосконалення мовних навичок. Водночас традиційні методи навчання часто не забезпечують достатньої гнучкості та індивідуалізації процесу. У зв'язку з цим адаптивне навчання з використанням штучного інтелекту (ШІ) набуває все більшої популярності, оскільки дозволяє враховувати особливості кожного студента та оптимізувати навчальний процес.

Адаптивне навчання ґрунтується на принципах персоналізації, коли навчальні матеріали та завдання автоматично підлаштовуються під рівень знань, темп засвоєння та інтереси студента. Це дозволяє значно підвищити ефективність навчального процесу, зменшуючи кількість помилок та посилюючи мотивацію до навчання. Використання ШІ відкриває нові можливості для створення інтерактивних, динамічних і персоналізованих платформ навчання.

Адаптивне навчання польської мови із застосуванням штучного інтелекту є перспективним напрямом у сфері освіти. Воно сприяє створенню більш гнучкого, ефективного та орієнтованого на потреби кожного студента освітнього процесу.

Система навчання польської мови, що базується на використанні ШІ, враховує:

- різні рівні підготовки;
- темпи засвоєння матеріалу;
- специфічні потреби кожного користувача.

Такий підхід допомагає знизити рівень стресу під час навчання та підвищити мотивацію, оскільки студент отримує завдання відповідно до своїх можливостей.

Додатково, застосування ІІІ-технологій сприяє автоматизації рутинних процесів, таких як перевірка завдань, відстеження прогресу та генерація рекомендацій. Це дозволяє викладачам зосередитися на ключових аспектах навчального процесу, а студентам – отримувати більш якісний зворотний зв'язок.

Завдяки інтерактивним функціям, таким як чат-боти та підсистеми рекомендацій, студенти мають змогу постійно практикувати польську мову та отримувати підтримку в реальному часі, що сприяє кращому її засвоєнню та вдосконаленню мовних навичок.

1 ПОСТАНОВКА ЗАДАЧІ ВИВЧЕННЯ ПОЛЬСЬКОЇ МОВИ

Вивчення польської мови – це як отримання ключа, що відкриває двері до нових можливостей у спілкуванні, кар’єрі, освіті та розумінні культури Польщі й інших країн Центральної Європи. Польська мова не лише сприяє професійному розвитку, а й допомагає глибше зануритися в історичні та культурні традиції польського народу.

Щоб пояснити користь вивчення польської мови простими аналогіями, варто розглянути це з кількох точок зору: як міст до якісної освіти, як інструмент для побудови успішної кар’єри та як спосіб розширення світогляду через знайомство з культурою та літературою Польщі.

1.1 Діаграма варіантів використання для наочності постановки завдання вивчення польської мови

Діаграма варіантів використання (Use Case Diagram) або ДВВ є одним із важливих інструментів для моделювання функціональності системи з точки зору її користувачів або інших зацікавлених осіб (акторів). Вона допомагає відобразити взаємодію між користувачами (акторами) та системою, а також визначити функціональні можливості, які повинна підтримувати система [1].

На діаграмі варіантів використання (ДВВ) актори, такі як користувачі, системи або інші зовнішні компоненти, представлені як зовнішні елементи. Їх взаємодія із системою відображається через варіанти використання (ВВ), які демонструють ключові функції або сценарії роботи. Діаграма дає змогу зрозуміти:

- які завдання виконує система;
- як користувачі взаємодіють із її компонентами;
- які основні функції вона реалізує;
- які існують зв’язки між акторами та системою.

У системах вивчення іноземних мов із використанням ШІ, ДВВ може бути використана для моделювання таких функціональних можливостей:

- «автоматичне виправлення вимови» [2];
- «виправлення граматичних помилок» [3];
- «персоналізовані уроки» [4];
- «створення тестів» [5];
- «аналіз тестів» [6].

Побудуємо ДВВ (Use Case Diagram), щоб проілюструвати причини вивчення польської мови. У центрі діаграми розміщується головний персонаж – актор «Студент», який виступає бенефіціаром, тобто отримувачем знань». Це може бути як учень чи студент, так і будь-яка інша людина, яка прагне опанувати іноземну мову.

Сформуємо ДВВ у вигляді програмного коду, включно із зв'язками:

```
@startuml
actor Студент as Student

usecase "Подорожі та комунікація з поляками" as Travel
usecase "Навчання за кордоном" as StudyAbroad
usecase "Кар'єрні перспективи" as Career
usecase "Складання мовних іспитів" as Exams
usecase "Культурний розвиток" as Culture
usecase "Споживання контенту польською мовою" as Content
usecase "Участь у програмах обміну" as Exchange
usecase "Особистий розвиток та впевненість" as PersonalGrowth

Student --> Travel
Student --> StudyAbroad
Student --> Career
Student --> Exams
Student --> Culture
Student --> Content
Student --> Exchange
Student --> PersonalGrowth

Travel --> Culture : "Підвищує культурний розвиток"
```

```

StudyAbroad --> Exchange : "Пов'язано з міжнародними
програмами"
Career --> Exams : "Потрібно для роботи"
Exams --> PersonalGrowth : "Підвищує впевненість"
Culture --> Content : "Мотивація до споживання контенту"
Exchange --> PersonalGrowth : "Розширює кругозір"
Content --> Travel : "Може надихнути на подорожі"

@enduml
---
```

Виходячи з наведеного коду, побудуємо відповідну діаграму (рис. 1.1). Ця діаграма ілюструє ключові мотиви, що спонукають студентів до вивчення польської мови. Головним актором виступає «Студент», який взаємодіє з різними варіантами використання, що відображають основні причини опанування польської мови.

Об'єкт розробки – процес адаптивного навчання польської мови на основі штучного інтелекту, який підлаштовується під рівень знань студентів або дорослих користувачів.

З цього можна сформулювати мету роботи.

Метою роботи є розробка прототипу інтелектуальної навчальної системи, яка автоматично адаптується до рівня знань, потреб і індивідуальних особливостей кожного користувача, забезпечуючи персоналізований підхід до вивчення польської мови.

Основні завдання для досягнення мети:

- визначення основних компонентів адаптивної системи навчання польської мови;
- аналіз алгоритмів персоналізованого підбору навчальних матеріалів та завдань на основі даних користувача;
- інтеграція технологій обробки мови для забезпечення інтерактивної взаємодії з користувачем;
- розробка прототипу графічного інтерфейсу користувача (GUI), що відповідає сучасним вимогам зручності та функціональності.



Рисунок 1.1 – ДВВ (Use Case Diagram), що ілюструє причини вивчення польської мови

2 ПРОЄКТУВАННЯ УНІФІКОВАНОГО ДІАГРАМНОГО КОМПЛЕКСУ

2.1 Побудова діаграми станів

Діаграма станів – ДС (State Machine Diagram) – це діаграма у Unified Modeling Language (UML), яка використовується для моделювання поведінки системи або об'єкта, що змінюється залежно від подій та умов. Вона описує всі можливі стани об'єкта протягом його життєвого циклу, а також переходи між цими станами під впливом певних тригерів. Кожен стан може містити внутрішні дії або підстани, що уточнюють поведінку системи в конкретний момент [7].

ДС складається з початкового стану, кінцевого стану, проміжних станів та переходів між станами. Переходи визначаються умовами або подіями, які спричиняють зміну стану. Зазвичай ДС використовуються для моделювання об'єктів із складною поведінкою, наприклад, програмного забезпечення, систем керування, бізнес-процесів (підходить для нашого випадку вивчення польської мови) або взаємодії користувачів із системою.

Основне призначення ДС – показати, як об'єкт змінює свій стан залежно від зовнішніх подій і як ці зміни впливають на його подальшу поведінку. Це дозволяє зрозуміти логіку роботи системи, врахувати можливі сценарії поведінки та підвищити ефективність розробки та тестування системи.

Сформуємо програмний код для побудови State Machine Diagram.

```
@startuml
[*] --> ПочатокНавчання

state ПочатокНавчання {
    [*] --> ВизначенняРівняЗнань
    ВизначенняРівняЗнань --> АдаптаціяМатеріалу : Результати
тестування
}

state АдаптаціяМатеріалу {
    [*] --> Навчання
    Навчання --> ПеревіркаПрогресу : Завдання виконано
```

```

    ПеревіркаПрогресу --> Навчання : Помилки в тестах
    ПеревіркаПрогресу --> ЗавершенняКурсу : Досягнуто рівень
}

ЗавершенняКурсу --> [*]

state Навчання {
    [*] --> ПереглядВідеоуроків
    ПереглядВідеоуроків --> ВиконанняЗавдань
    ВиконанняЗавдань --> ТренуванняВимови
    ТренуванняВимови --> ПереглядВідеоуроків : Повторення
матеріалу
}

@enduml

```

Побудуємо на підставі цього коду ДС (рис. 2.1).

Опишемо ДС для процесу вивчення польської мови за допомогою ШІ. Ця ДС відображає процес вивчення польської мови засобами ШІ. Вона включає основні етапи та переходи між ними.

Основні стани:

1. «Початок навчання» – процес розпочинається з цього стану;
2. «Визначення рівня знань» – ШІ проводить тестування студента для визначення його поточного рівня володіння мовою;
3. «Адаптація матеріалу» – на основі результатів тестування ШІ підбирає індивідуальну програму навчання;
4. «Навчання» – основний стан, який включає кілька підстанів:
 - «Перегляд відеоуроків» – студент переглядає навчальні відео;
 - «Виконання завдань» – після перегляду матеріалу студент виконує завдання для закріплення знань;
 - «Тренування вимови» – ШІ допомагає студенту вдосконалити вимову;
5. «Перевірка прогресу» – регулярне оцінювання прогресу студента шляхом тестів та завдань;
6. «Завершення курсу» – після досягнення визначеного рівня студент завершує курс.

Опис зв'язку між станами:

- після початку навчання студент проходить тестування для визначення рівня знань;

- на основі результатів тестування ІІІ адаптує програму навчання;
- під час навчання студент проходить різні етапи, такі як перегляд відеоуроків, виконання завдань та тренування вимови;
- якщо під час перевірки прогресу виявляються помилки, студент повертається до навчання;
- курс завершується, коли студент досягає поставлених цілей.

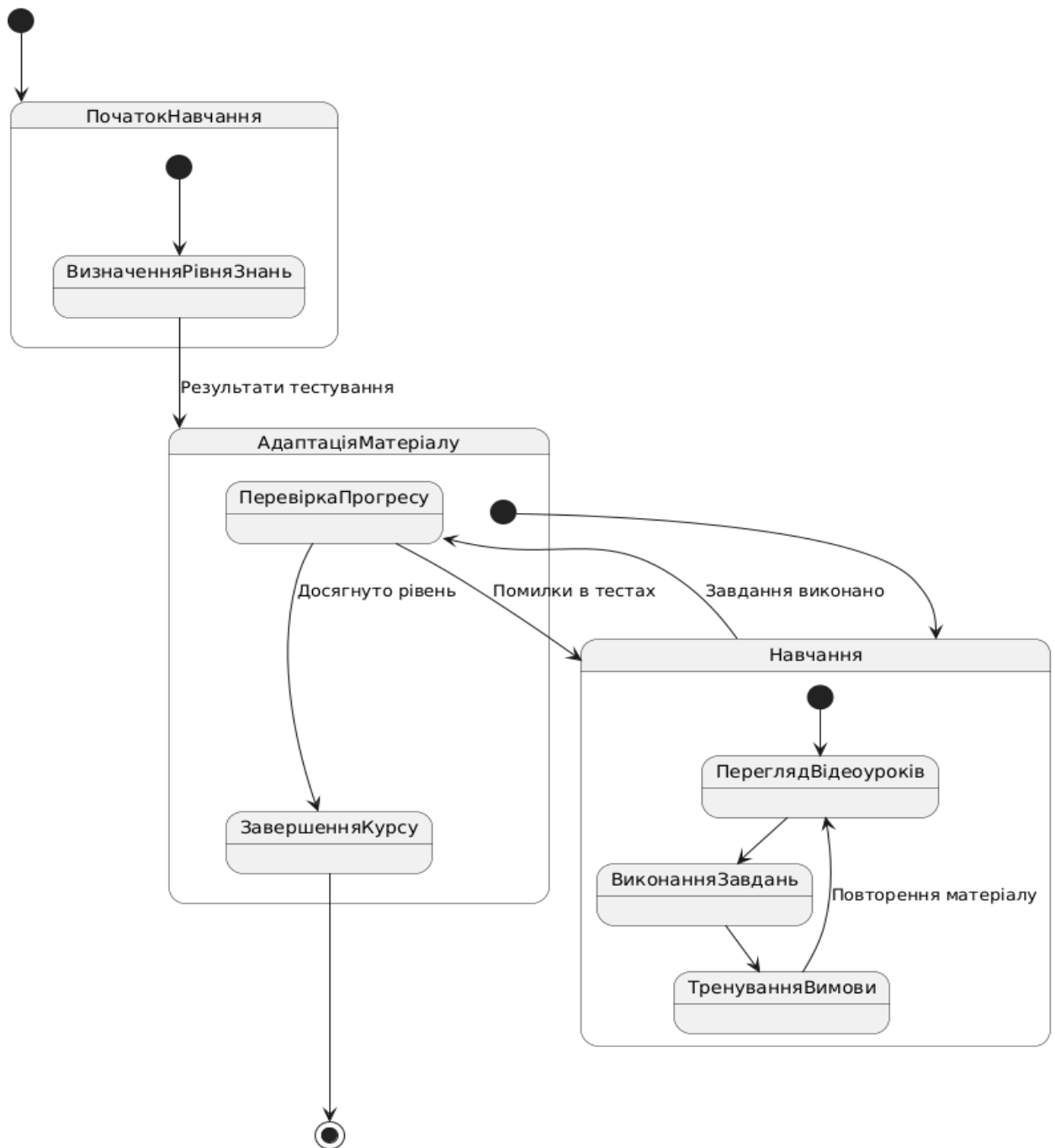


Рисунок 2.1 – ДС (State Machine Diagram) для розробки віртуальної платформи адаптивного навчання польської мови засобами ІІІ

2.2 Прокєтування діаграми компонентів

Діаграма компонентів (ДКМ) є ключовим інструментом в об'єктно-орієнтованому аналізі та проєктуванні програмного забезпечення. Вона використовується для моделювання фізичних компонентів системи та їх взаємодії на високому рівні абстракції. Завдяки цій діаграмі можна візуалізувати архітектуру програмної системи, визначити функціональне призначення її складових частин і проаналізувати їхню взаємодію. Це допомагає зрозуміти, як розподіляються програмні модулі та яким чином вони обмінюються даними [8].

Основні елементи діаграми компонентів включають компоненти (наприклад, модулі або сервіси), їхні інтерфейси та зв'язки між ними. До компонентів можуть належати програмні бібліотеки, бази даних, вебсервіси тощо. Зв'язки між ними описують залежності, що існують у системі. Діаграми компонентів відіграють важливу роль у відображенні структурної організації системи та плануванні її розробки, включаючи тестування, впровадження й оновлення [9].

Сформуємо ДКМ у вигляді програмного коду, включно із зв'язками та наведемо опис та пояснення для цієї діаграми.

1. «User Interface (Інтерфейс користувача)»

Цей компонент відповідає за взаємодію з користувачем. Він представлений двома підкомпонентами:

- «Web App (Веб-додаток)»: дозволяє студентам отримати доступ до системи через веб-браузер;
- «Mobile App (Мобільний додаток)»: доступ до системи через мобільні пристрої.

```
@startuml
package "User Interface" {
    [Web App]
    [Mobile App]
}
```

2. «AI Engine (ШІ-двигун)»

Цей компонент здійснює основну роботу з персоналізації навчального процесу. Він включає:

- «Recommendation Module (Модуль рекомендацій)»: аналізує дані студента та рекомендує відповідні курси;
- «Progress Analysis (Аналіз прогресу)»: відстежує успішність студента під час навчання;
- «Feedback Generator (Генератор зворотного зв'язку)»: надає студенту персоналізовані рекомендації та коментарі.

```
package "AI Engine" {
  [Recommendation Module]
  [Progress Analysis]
  [Feedback Generator]
}
```

3. «Course Management (Керування курсами)»

Цей компонент відповідає за збереження та управління курсами та навчальними матеріалами:

- «Course Database (База даних курсів)»: зберігає інформацію про доступні курси;
- «Learning Material Repository (Сховище навчальних матеріалів)»: містить навчальні матеріали для кожного курсу;
- «User Data Storage (Сховище даних користувачів)»: зберігає інформацію про студентів і їхній прогрес.

```
package "Course Management" {
  [Course Database]
  [Learning Material Repository]
  [User Data Storage]
}
```

4. «Progress Tracking (Відстеження прогресу)»

Цей компонент забезпечує відстеження прогресу студента та надання аналітики:

- «Progress Report Generator (Генератор звітів про прогрес)»: створює

звіти про успішність студента;

– «Analytics Dashboard (Аналітична панель)»: відображає дані про прогрес студентів у вигляді графіків і діаграм;

```
package "Progress Tracking" {
  [Progress Report Generator]
  [Analytics Dashboard]
}
```

Пояснимо зв'язки між компонентами:

– «Web App / Mobile App → Recommendation Module»: інтерфейси користувача взаємодіють із модулем рекомендацій для персоналізації контенту;

– «Recommendation Module → Course Database / User Data Storage»: Модуль рекомендацій отримує інформацію про курси та студентів;

– «Progress Analysis → Progress Report Generator»: аналіз прогресу передає дані для створення звітів;

– «Feedback Generator → Analytics Dashboard»: генератор зворотного зв'язку забезпечує оновлення аналітичної панелі;

– «Course Database → Learning Material Repository»: база даних курсів отримує відповідні навчальні матеріали;

– «User Data Storage → Progress Report Generator»: сховище даних користувачів передає дані для створення звітів.

```
[Web App] --> [Recommendation Module]
[Mobile App] --> [Recommendation Module]
[Recommendation Module] --> [Course Database]
[Recommendation Module] --> [User Data Storage]

[Progress Analysis] --> [Progress Report Generator]
[Feedback Generator] --> [Analytics Dashboard]

[Course Database] --> [Learning Material Repository]
[User Data Storage] --> [Progress Report Generator]

@enduml
```

Побудуємо цю діаграму (рис. 2.2).

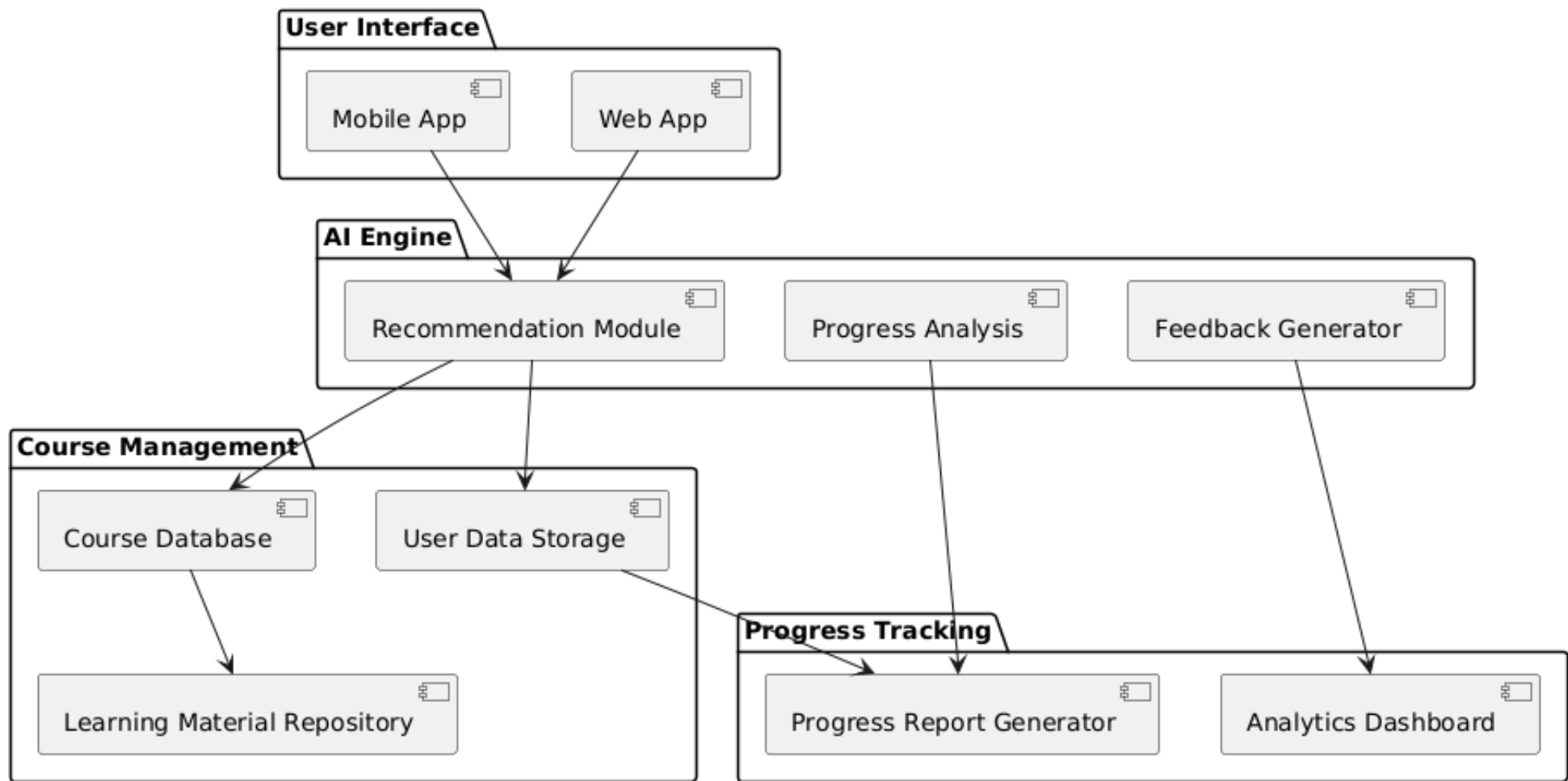


Рисунок 2.2 – ДКМ (Component Diagram) для розробки віртуальної платформи адаптивного навчання польської мови засобами штучного інтелекту

2.3 Проектування діаграми розгортання

Діаграма розгортання (Deployment Diagram, ДР) є одним із ключових типів діаграм у UML. Вона використовується для візуалізації фізичного розміщення програмних компонентів на апаратних засобах. Діаграма демонструє, як програмні модулі, такі як класи або сервіси, розподіляються між фізичними пристроями, зокрема серверами чи комп'ютерами, а також відображає зв'язки між ними [10].

Одним із прикладів застосування ДР у системах вивчення іноземних мов із використанням ШІ є створення платформи для автоматизованого мовного навчання. Така система може містити різні вузли, зокрема:

- сервери для обробки запитів користувачів;
- бази даних (БД) для збереження інформації про уроки та користувачів;
- користувацький інтерфейс на пристроях студентів.

На сервері може бути розміщене програмне забезпечення для обробки природної мови, яке використовує машинне навчання для аналізу вимови та граматики студентів.

Для підвищення продуктивності та зменшення навантаження на сервери окремі компоненти системи можуть бути розподілені між різними пристроями, наприклад, для навчання:

- словникового запасу;
- вимови;
- граматики.

ДР у цьому випадку дозволяє візуалізувати процес обробки запитів користувачів, а також передачу даних між пристроями, забезпечуючи ефективний навчальний процес.

Далі сформуємо код для опису ДР системи вивчення польської мови із використанням ШІ, який міститиме основні сервери, компоненти та їх взаємозв'язки.

```

@startuml

' Deployment Diagram for AI-Powered Language Learning System

node "Client Device" {
  [Web Browser]
  [Mobile App]
}
node "Application Server" {
  component "Web Application" as WebApp
  component "AI Module" as AIModule
  component "Database Access Layer" as DBAccess
}
node "Database Server" {
  database "Knowledge Database" as DB
}
node "AI Processing Server" {
  component "Natural Language Processing (NLP) Engine" as
NLPEngine
  component "Recommendation System" as RecSys
}
' Connections between nodes and components
[Web Browser] --> WebApp : "HTTPS"
[Mobile App] --> WebApp : "API"
WebApp --> DBAccess : "Database Queries"
DBAccess --> DB : "SQL Queries"
AIModule --> NLPEngine : "Process User Input"
AIModule --> RecSys : "Generate Recommendations"
RecSys --> DB : "Fetch Learning Materials"
NLPEngine --> DB : "Store Processed Data"

@enduml

```

Пояснимо наведений фрагмент коду:

– «Client Device» – клієнтський пристрій є основним засобом взаємодії користувачів із системою, доступ до неї здійснюється через веб-браузери або мобільні додатки, що забезпечують зручний інтерфейс для навчання;

– «Application Server» – сервер додатків відповідає за обробку запитів від клієнтів, забезпечує доступ до модулів штучного інтелекту та взаємодіє з базою даних, надаючи користувачам необхідні матеріали та функціонал;

– «Database Server» – сервер БД містить ключову інформацію для підтримки навчального процесу, зокрема дані про студентів, навчальні курси, матеріали та звіти про їхній прогрес;

– «AI Processing Server» – сервер обробки штучного інтелекту аналізує введений студентами текст за допомогою NLP-двигуна (Natural Language Processing), генерує персоналізовані рекомендації та адаптує навчальний процес на основі досягнень користувача;

– безпечний обмін даними між усіма компонентами системи здійснюється через HTTPS, що забезпечує захист інформації від несанкціонованого доступу або перехоплення.

На підставі наведеного коду, побудуємо ДР (рис. 2.3).

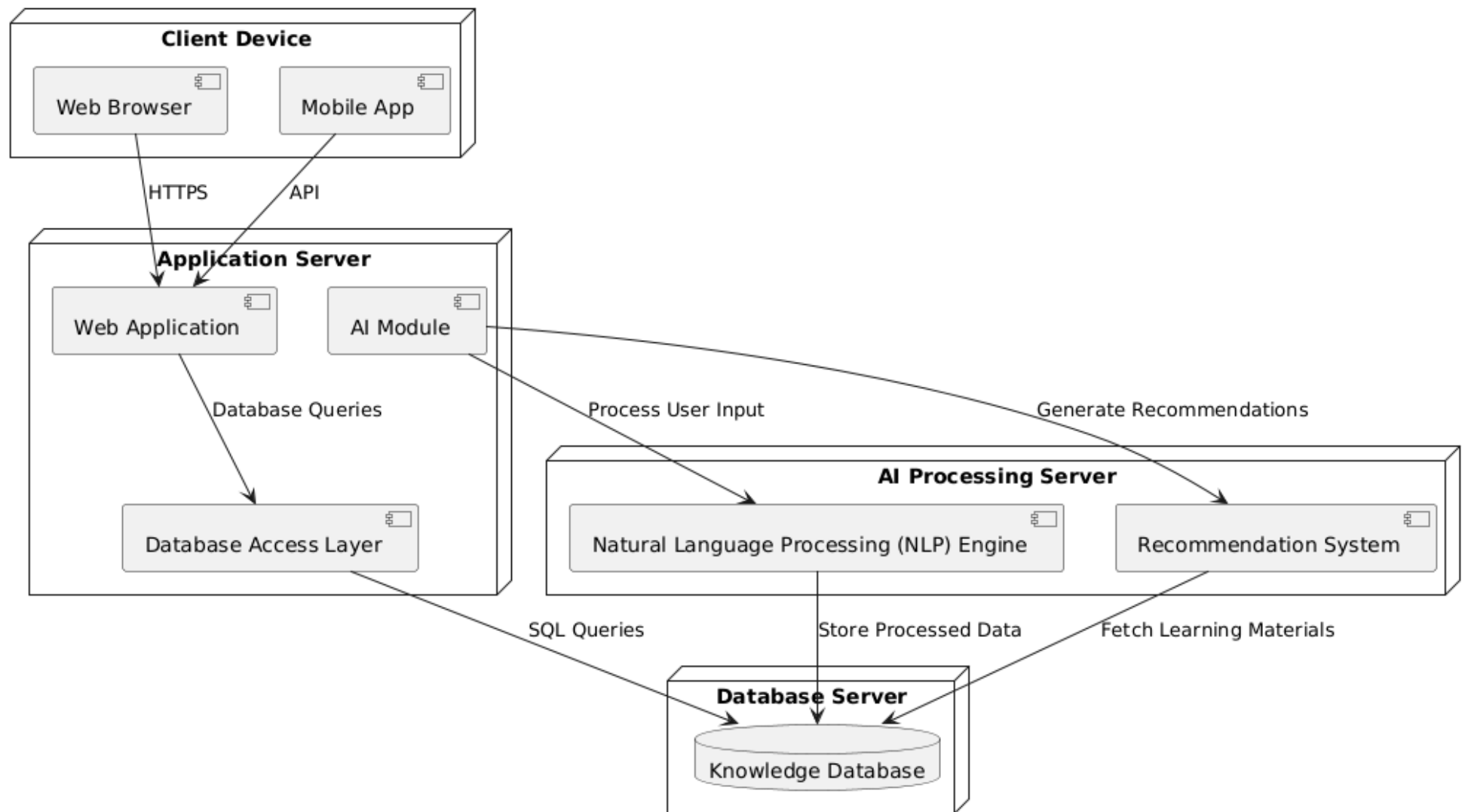


Рисунок 2.3 – ДР (Deployment Diagram) для віртуальної платформи адаптивного навчання польської мови засобами штучного інтелекту

3. ФОРМУВАННЯ БАЗИ ДАНИХ ТА БАЗИ ЗНАНЬ

3.1 Бази даних і бази знань

База даних (БД) – це структурована система зберігання інформації, організована для ефективного:

- пошуку;
- збереження;
- обробки;
- управління даними.

Основне призначення БД – упорядковане зберігання значних обсягів інформації, що забезпечує швидке виконання запитів та можливість аналізу.

Дані у базі можуть бути представлені у вигляді:

- таблиць;
- записів;
- полів;
- індексів;
- інших структур, які забезпечують логічний та фізичний доступ до інформації.

База знань (БЗ) – це система, що містить не лише дані, а й правила, факти та методи, які дозволяють отримувати нову інформацію на основі наявних знань. На відміну від БД, БЗ зосереджена на збереженні та використанні експертних знань у певній предметній області, що сприяє прийняттю рішень та автоматизації процесів розв’язання завдань.

БЗ широко застосовуються в системах ШІ для підтримки:

- експертних систем;
- чат-ботів;
- діагностичних платформ;
- рекомендаційних сервісів.

3.2 Структурування бази даних

Для створення системи вивчення іноземної мови із застосуванням штучного інтелекту необхідно розробити БД, що містить основні сутності та взаємозв'язки між ними.

3.2.1 Опис сутностей БД

Таблиця 1: Students (Студенти).

Містить інформацію про студентів, які користуються системою. Вона є центральною сутністю БД, оскільки кожен студент має доступ до курсів, навчальних матеріалів та отримує звіти про свій прогрес.

Основні атрибути:

- student_id (Primary Key) – унікальний ідентифікатор студента;
- name – ім'я студента для персоналізації інтерфейсу;
- email – електронна пошта для входу в систему та отримання повідомлень;
- level – поточний рівень володіння мовою (Beginner, Intermediate, Advanced), що допомагає адаптувати навчальні матеріали.

Таблиця 2: Courses (Курси).

Зберігає інформацію про мовні курси, що доступні студентам. Курси адаптуються до різних рівнів складності та охоплюють різні навички:

- course_id (Primary Key) – унікальний ідентифікатор курсу;
- title – назва курсу (наприклад, «Польська для подорожей»);
- description – короткий опис курсу;
- difficulty_level – рівень складності (A1, B1, C2).

Таблиця 3: LearningMaterials (Навчальні матеріали).

Містить ресурси, що використовуються в курсах.

Основні атрибути:

- material_id (Primary Key) – унікальний ідентифікатор матеріалу;

- type – тип ресурсу (відео, текст, аудіо);
- content – зміст навчального матеріалу;
- language – мова викладу.

Таблиця 4: ProgressReports (Звіти про прогрес).

Дозволяє відстежувати успішність студентів у навчанні.

Основні атрибути:

- report_id (Primary Key) – унікальний ідентифікатор звіту;
- student_id (Foreign Key) – посилання на студента;
- course_id (Foreign Key) – посилання на курс;
- progress_percentage – відсоток виконання курсу.

Таблиця 5: Assignments (Завдання)

Зберігає завдання, які студенти мають виконати.

Основні атрибути:

- assignment_id (Primary Key) – унікальний ідентифікатор завдання;
- course_id (Foreign Key) – посилання на курс;
- student_id (Foreign Key) – посилання на студента;
- status – статус виконання (виконано/не виконано).

Таблиця 6: AI_Tutors (ШІ-тьютори)

Містить інформацію про ШІ-моделі, що підтримують студентів у навчанні. Основні функції ШІ-тьюторів:

- генерація персоналізованих матеріалів;
- автоматизоване оцінювання завдань;
- надання рекомендацій щодо подальшого навчання.

3.2.2 Взаємозв'язки між сутностями

Розробимо таку структуру БД, завдяки якій забезпечується ефективне збереження, обробка та аналіз інформації, що необхідна для персоналізованого навчання.

Пояснимо зв'язки між сутностями:

- Students → ProgressReports: один студент може мати кілька звітів про прогрес;
- Students → Assignments: кожен студент виконує певні завдання в межах курсу;
- Courses → LearningMaterials: кожен курс містить свої навчальні матеріали;
- Courses → Assignments: завдання прив'язані до конкретних курсів;
- ProgressReports → Courses: звіт про успішність пов'язаний із певним курсом;
- AI_Tutors → ProgressReports: ШІ-тьютори аналізують прогрес студентів та надають рекомендації.

3.3 Формування ER-діаграми

ER-діаграма (Entity-Relationship Diagram, діаграма сутностей-зв'язків) є візуальним інструментом для відображення структури даних і зв'язків між ними в базі даних. Вона широко застосовується для моделювання інформаційних систем, допомагаючи зрозуміти, як різні сутності взаємодіють одна з одною. ER-діаграма дозволяє логічно організувати дані, що є ключовим аспектом їхнього ефективного зберігання, обробки та аналізу.

Використання ШІ в контексті ER-діаграм можна проілюструвати на прикладі створення бази даних для аналізу великих обсягів інформації. Наприклад, у проєктах машинного навчання, де зберігаються дані про користувачів, їхню активність на платформі та результати взаємодії з навчальним контентом.

ER-діаграма допомагає структурувати такі дані та показати взаємозв'язки між ключовими об'єктами, зокрема:

- користувачами;
- навчальними матеріалами;

- завданнями;
- прогресом студентів тощо.

Це сприяє ефективному управлінню даними, що використовується для персоналізації навчального процесу, прогнозування результатів і вдосконалення адаптивної системи навчання.

Далі сформуємо код для представлення ER-діаграми:

```
@startuml
entity "Students" {
    * student_id : INT
    --
    name : VARCHAR
    email : VARCHAR
    level : VARCHAR
}

entity "Courses" {
    * course_id : INT
    --
    title : VARCHAR
    description : TEXT
    difficulty_level : VARCHAR
}

entity "LearningMaterials" {
    * material_id : INT
    --
    type : VARCHAR
    content : TEXT
    language : VARCHAR
}

entity "ProgressReports" {
    * report_id : INT
    --
    student_id : INT
    course_id : INT
    progress_percentage : FLOAT
}

entity "Assignments" {
    * assignment_id : INT
```

```

--
course_id : INT
student_id : INT
status : VARCHAR
}

entity "AI_Tutors" {
* tutor_id : INT
--
name : VARCHAR
specialization : VARCHAR
}

Students ||--o{ ProgressReports : "has"
Students ||--o{ Assignments : "completes"
Courses ||--o{ LearningMaterials : "contains"
Courses ||--o{ Assignments : "has"
ProgressReports }--|| Courses : "relates to"
AI_Tutors ||--o{ ProgressReports : "generates"
@enduml

```

На підставі наведеного коду, побудуємо графічну модель структури БД (рис. 3.1).

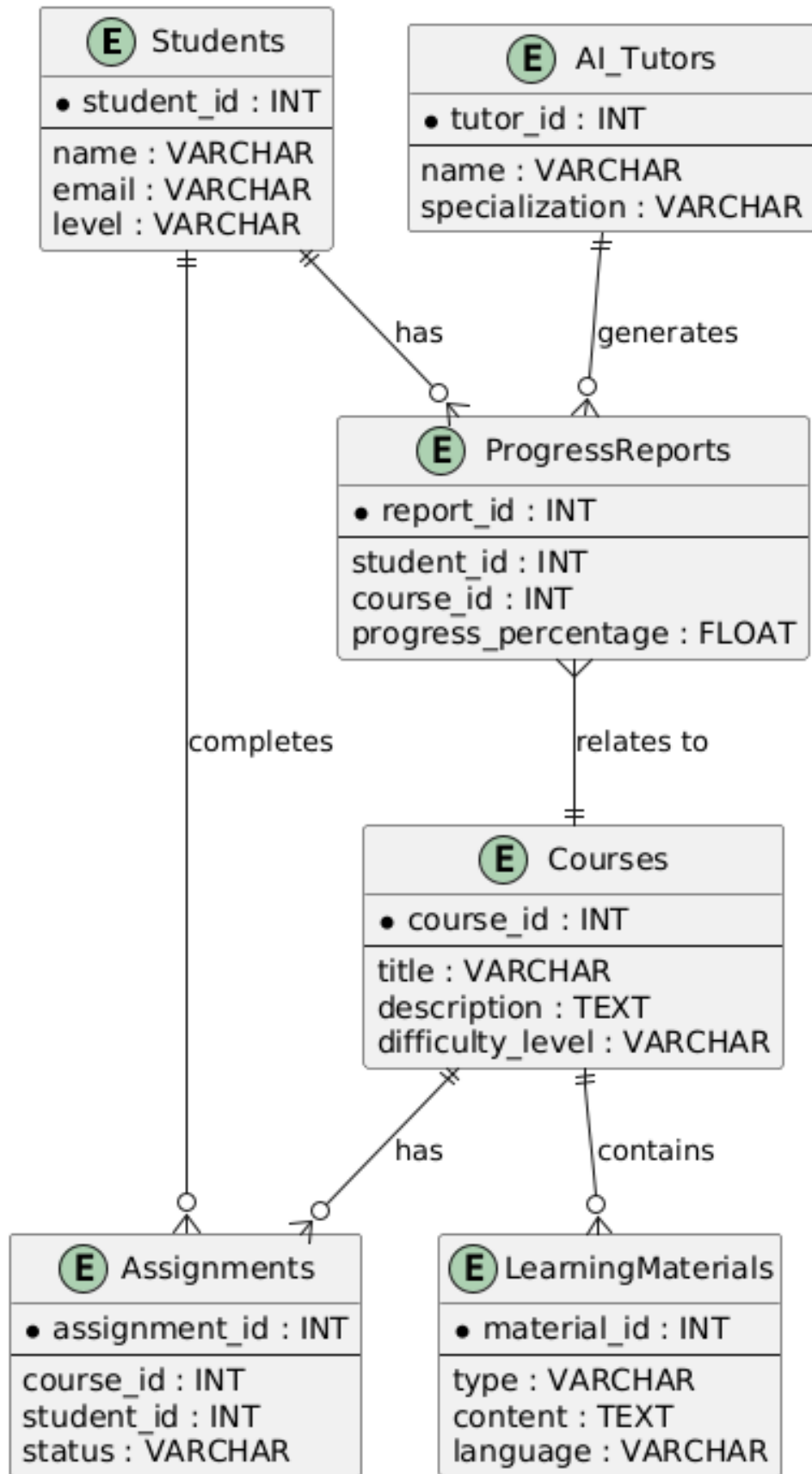


Рисунок 3.1 – ER-діаграма структури БД та БЗ для платформи вивчення польської мови з застосуванням адаптивної моделі навчання

4. СТВОРЕННЯ ПРОТОТИПУ ІНТЕРФЕЙСУ СИСТЕМИ

4.1 Визначення основних компонентів інтерфейсу

Графічний інтерфейс користувача (GUI – Graphic User Interface) для платформи вивчення польської мови на основі ШІ повинен бути інтуїтивно зрозумілим, адаптивним та привабливим для користувачів. Основні необхідні компоненти інтерфейсу:

а) ключові функції:

1) реєстрація та вхід у систему (за допомогою email, соціальних мереж або Google-акаунта);

2) відновлення пароля.

б) профіль користувача:

1) фото або аватар користувача;

2) вибір основної мови навчання;

3) вказання рівня володіння мовою (A1, A2, B1, B2, C1, C2);

4) редагування персональних даних.

в) дашборд:

1) щоденні завдання;

2) досягнення та нагороди;

3) відображення загального прогресу користувача;

4) час, проведений у навчанні.

г) чат із ШІ-репетитором:

1) інтерактивний чат із підтримкою текстового та голосового введення;

2) автоматичний аналіз помилок у тексті;

3) персоналізовані рекомендації від ШІ.

д) вправи та уроки:

1) аудіо- та відеоуроки;

2) граматичні вправи;

- 3) переклади текстів та диктанти;
- 4) інтерактивні завдання на розуміння мови.

е) аналіз вимови:

- 1) розпізнавання мовлення;
- 2) візуалізація помилок у вимові;
- 3) рекомендації щодо покращення вимови;
- 4) порівняння вимови з еталонними зразками.

ж) гейміфікація:

- 1) система рівнів та досягнень;
- 2) таблиці лідерів;
- 3) винагороди за активність (бали, значки, сертифікати);
- 4) можливість змагатися з друзями.

и) налаштування:

- 1) вибір теми інтерфейсу (світла/темна);
- 2) вибір мови інтерфейсу;
- 3) увімкнення / вимкнення сповіщень;
- 4) керування звуком і швидкістю відтворення медіаконтенту.

Отже, ці компоненти дозволять створити ефективну та зручну платформу для адаптивного навчання, що забезпечить персоналізований підхід для кожного користувача.

4.2 Розробка графічного інтерфейсу у Visual Studio

4.2.1 Вибір платформи та технології

Visual Studio надає широкий вибір інструментів для створення GUI, серед яких:

– Windows Forms (WinForms) – використовується для розробки настільних застосунків мовою C#;

- WPF (Windows Presentation Foundation) – забезпечує створення сучасних настільних додатків із гнучким дизайном;
- ASP.NET – підходить для розробки веб-додатків;
- MAUI – кросплатформне рішення для створення мобільних та настільних застосунків.

4.2.2 Створення нового проєкту у Visual Studio

Щоб розпочати новий проєкт, слід виконати такі дії:

- а) обрати опцію «Create a new project»;
- б) визначитися з відповідним типом проєкту залежно від вимог:
 - 1) Windows Forms App (.NET) – для настільних застосунків;
 - 2) WPF App (.NET) – для сучасних настільних рішень;
 - 3) ASP.NET Core Web App – для веб-додатків;
- в) задати назву проєкту та вибрати директорію для збереження файлів;
- г) натиснути «Create» для створення проєкту.

4.2.3 Реалізація логіки на C# у .NET

Щоб додати програмну логіку до інтерфейсу, необхідно виконати такі кроки:

- а) відкрити відповідний файл коду:
 - 1) MainWindow.xaml.cs – для WPF;
 - 2) Form1.cs – для WinForms;
- б) додати обробку подій для кнопок і текстових полів, що забезпечить інтерактивність інтерфейсу.

Нижче наведено приклад коду для кнопки «Почати урок» у WinForms:

```
e) private void btnStartLesson_Click(object sender, EventArgs
    {
```

```

        MessageBox.Show("Урок розпочато!", "Інформація",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

```

Після реалізації основної логіки інтерфейсу необхідно запустити проєкт, щоб перевірити його роботу. Для цього: натиснути F5 або кнопку «Start» у Visual Studio, щоб запустити застосунок і переглянути результати.

4.2.4 Оптимізація стилю та логіки

Для покращення функціональності та візуального вигляду програми варто реалізувати:

- підключення до БД – для збереження та обробки даних користувачів, їх прогресу та навчальних матеріалів;
- інтеграцію з AI-моделлю – для персоналізованих рекомендацій, аналізу тексту та голосових взаємодій;
- сучасний дизайн – використання стилів і тем (наприклад, у WPF – XAML-стилізація, у WinForms – бібліотеки для UI-елементів).

Ці покращення зроблять платформу більш ефективною, зручною та привабливою для користувачів.

4.2.5 Стартовий код проєкту у Visual Studio

У додатку А наводиться стартовий код для GUI у Visual Studio на основі Windows Forms. Цей код створено для реалізації платформи вивчення польської мови із застосуванням адаптивної моделі навчання.

У додатку Б наводиться код для дизайну форми платформи вивчення польської мови із застосуванням адаптивної моделі навчання. Це версія для польськомовного інтерфейсу навчальної платформи.

4.3 Результати розробки прототипу GUI

4.3.1 Створення десктопного додатку

Беручи до уваги наведені вимоги, було розроблено прототип графічного інтерфейсу користувача (GUI) для платформи вивчення польської мови із застосуванням адаптивної моделі навчання. Фронтендний дизайн представлений на рис. 4.1.

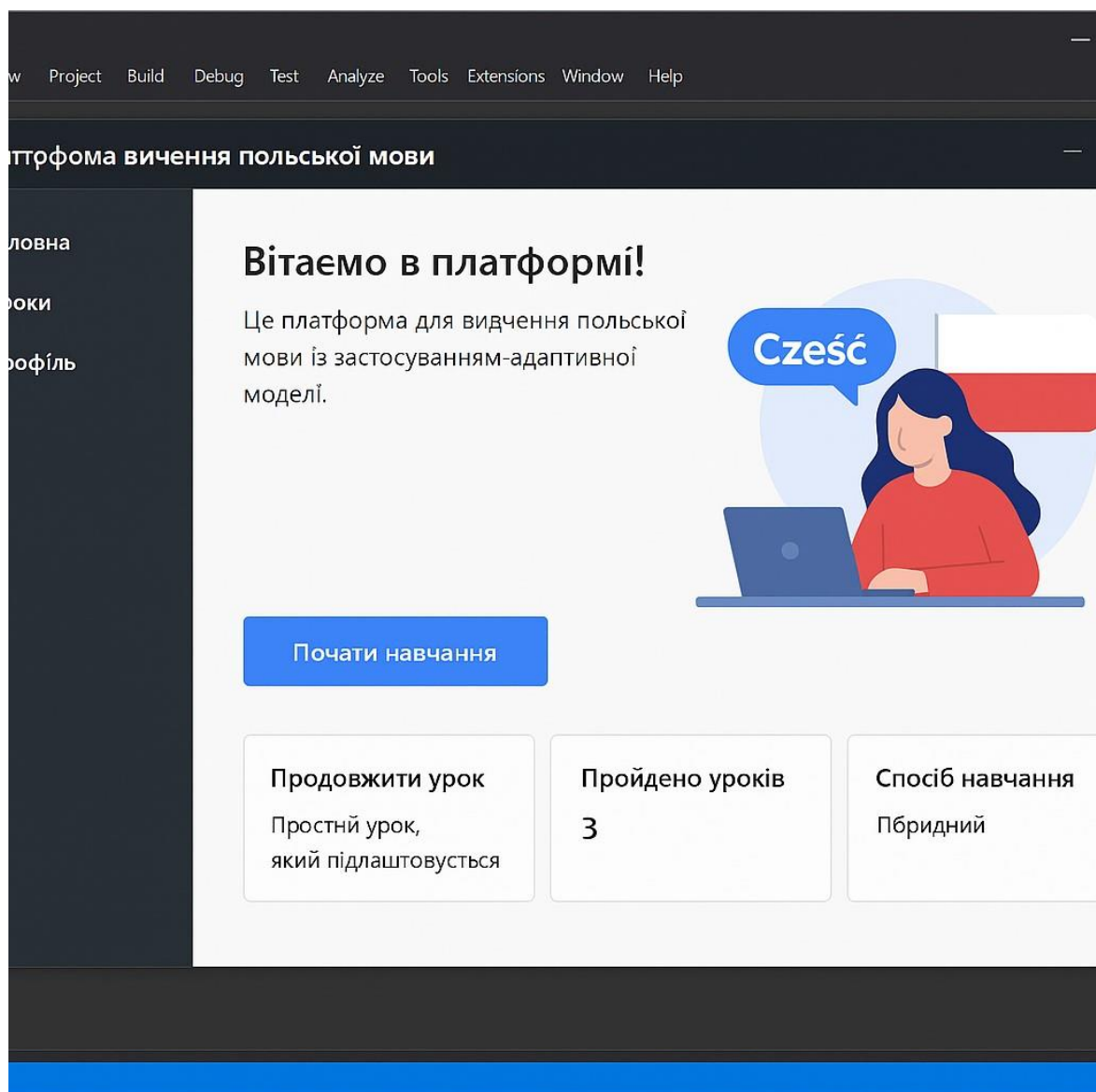


Рисунок 4.1 – Фронтендний прототип графічного інтерфейсу користувача для платформи вивчення польської мови із застосуванням адаптивної моделі

Розроблений прототип інтерфейсу містить (рис. 4.2):

- головний інформаційний екран із ключовими компонентами;
- відображення прогресу учня та рекомендованих матеріалів;
- інтерактивний чат-бот для практики;
- навігаційне меню.

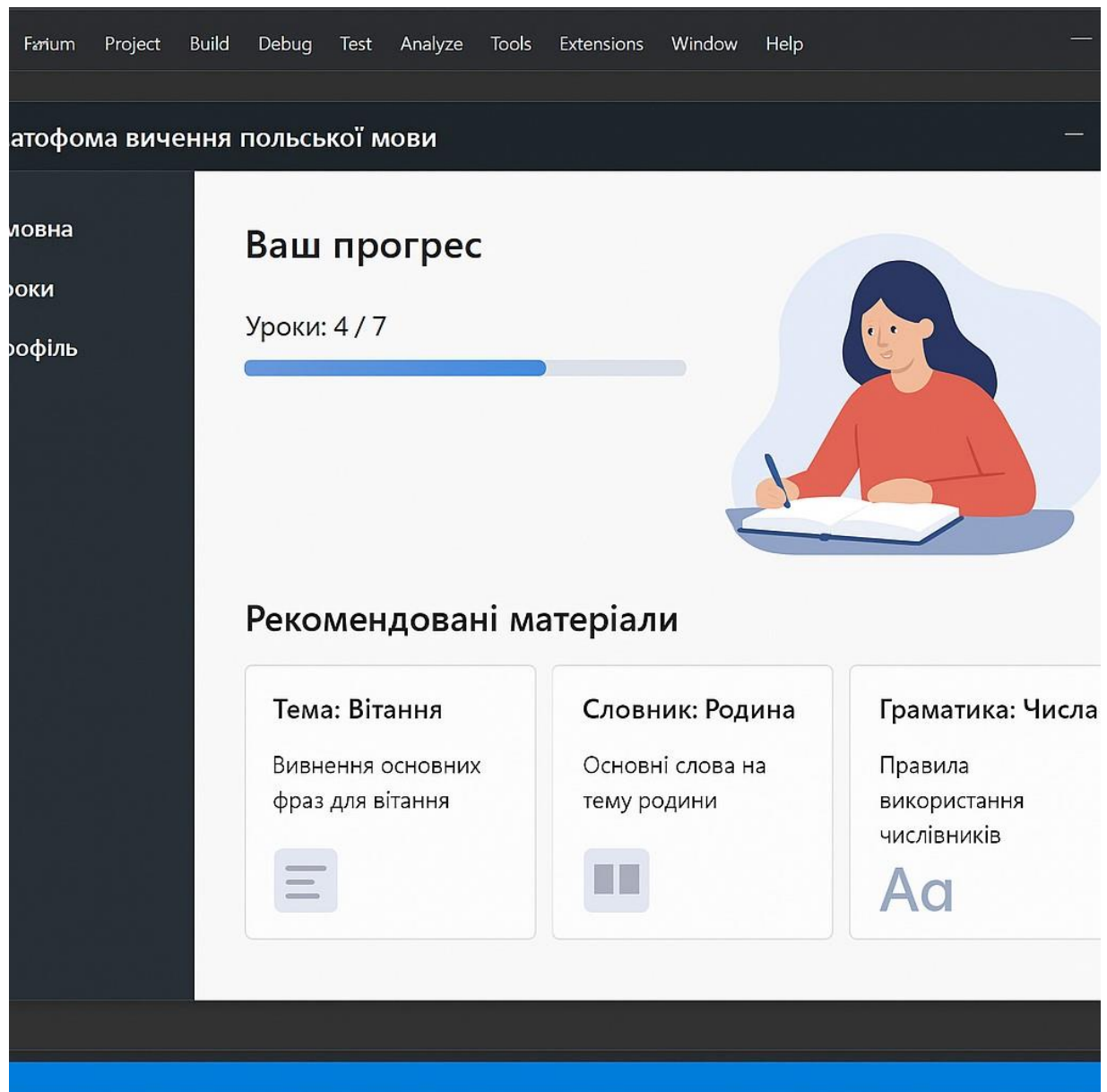


Рисунок 4.2 – Фронтендний прототип графічного інтерфейсу користувача платформи вивчення польської мови із застосуванням адаптивної моделі навчання із відображенням прогресу студента та рекомендованих дидактичних матеріалів

Наприклад, інтерфейс, зображений на рис. 4.2, має такі особливості:

- темну тему та впорядковану панель навігації;
- чітко структуровані секції для відображення навчального прогресу;
- рекомендації від ШІ;
- інтерактивну зону для практичних занять.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи на тему «Розробка віртуальної платформи вивчення польської мови з застосуванням адаптивної моделі навчання» було досягнуто основну мету роботи – створено прототип адаптивної системи, яка підвищує мотивацію користувачів, покращує результати навчання та забезпечує ефективний зворотний зв'язок під час вивчення польської мови.

У процесі роботи було виконано такі основні завдання:

а) проведено аналіз сучасних методів адаптивного навчання та їх використання в системах ІІІ;

б) спроектовано уніфікований діаграмний комплекс, що включає діаграми ВВ, станів, компонентів та розгортання;

в) розроблено структуру БД та БЗ, що забезпечує ефективне зберігання та управління навчальними матеріалами та даними користувачів;

г) створено графічний інтерфейс користувача (GUI) у середовищі Visual Studio з урахуванням принципів адаптивного навчання;

д) реалізовано базову логіку роботи системи, що включає:

1) персоналізацію навчального процесу;

2) інтерактивний чат-бот;

3) аналіз вимови користувачів.

Основним отриманим результатом роботи стала розробка прототипу адаптивної платформи для вивчення польської мови, що включає:

а) головний інформаційний екран із ключовими елементами;

б) систему моніторингу прогресу користувача та персоналізовані рекомендації;

в) інтерактивний чат-бот для практики та корекції вимови;

г) структуроване меню навігації, яке дозволяє легко орієнтуватися в навчальному процесі.

Прототип дозволяє автоматично адаптувати навчальні матеріали до рівня знань студента та покращувати якість навчання завдяки використанню технологій ШІ.

Подальші перспективи та розвиток проєкту може включати:

- а) інтеграцію нейромережевих моделей для покращення розпізнавання мови та генерації персоналізованих навчальних планів;
- б) розширення функціоналу платформи, включаючи мобільну версію додатку;
- в) додавання гейміфікаційних елементів для підвищення мотивації користувачів;
- г) створення механізмів зворотного зв'язку від користувачів для покращення адаптивності системи.

Таким чином, розроблена платформа закладає основу для подальшого розвитку інноваційних методів адаптивного навчання, що можуть бути використані не лише для вивчення польської мови, а й для інших мовних дисциплін та завдань.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Великодний С. С. Моделі та методи проактивного управління проєктами із розвитку програмних систем і продуктів. Монографія. Одеса: Одеський державний екологічний університет, 2021. 322 с. ISBN 978-966-186-182-3.
2. Systematic Review for AI-based Language Learning Tools. URL: https://www.researchgate.net/publication/356026044_Systematic_review_for_AI-based_language_learning_tools (дата звернення: 16.04.2025).
3. The use of AR elements in the study of foreign languages at the university. URL: <https://arxiv.org/pdf/2202.09161> (дата звернення: 18.04.2025).
4. Enhancing Second Language Acquisition through Artificial Intelligence (AI): Current Insights and Future Directions. URL: <https://www.jrspelt.com/wp-content/uploads/2023/09/jrspelt739003.pdf> (дата звернення: 19.04.2025).
5. Artificial Intelligence (AI): A Review of its Uses in Language Teaching and Learning. URL: https://www.researchgate.net/publication/342050108_Artificial_Intelligence_AI_A_Review_of_its_Uses_in_Language_Teaching_and_Learning (дата звернення: 21.04.2025).
6. Artificial Intelligence in Second Language Learning. URL: https://www.academia.edu/97906999/Artificial_Intelligence_in_Second_Language_Learning (дата звернення: 25.04.2025).
7. Velykodniy S. S. Analysis and synthesis of the results of complex experimental research on reengineering of open CAD systems. Applied Aspects of Information Technology. 2019. Vol. 2. No 3. P. 186–205. DOI: 10.15276/aait.03.2019.2.
8. Великодний С. С. Моделювання складних процесів та систем (Частина 1): конспект лекцій. Одеса: Одеський державний екологічний університет, 2021. 92 с. ISBN 978-966-186-181-6.
9. Великодний С. С., Тимофеева О. С., Зайцева-Великодна С. С. Метод розрахунку показників оцінки проєкту при виконанні реінжинірингу

програмних систем. *Радіоелектроніка, інформатика, управління*. 2018. № 4. С. 135–142. DOI: 10.15588/1607-3274-2018-4-13.

10. Velykodniy S., Burlachenko Zh., Zaitseva-Velykodna S. Modelling the behavioural component of the emergent parallel processes of working with graph databases using Petri net-tools. *International Journal of Parallel, Emergent and Distributed Systems*. (Scopus) 2021. Vol. 36. Iss. 6. P. 498-515. (Scopus) DOI: <https://doi.org/10.1080/17445760.2021.1934836>. Taylor & Francis Group, England & Wales. London.

ДОДАТОК А

Стартовий код для GUI платформи вивчення польської мови із застосуванням адаптивної моделі навчання

```

using System;
using System.Windows.Forms;

namespace PolishLanguageLearningApp
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void MainForm_Load(object sender, EventArgs
e)
        {
            // Inicjalizacja ekranu głównego
            lblWelcome.Text = "Witamy w aplikacji AI do
nauki języka polskiego!";
            progressDailyTasks.Value = 50; // Przykładowy
postęp użytkownika

            // Załadowanie danych użytkownika z bazy danych
lub API
            LoadUserData();
        }

        private void LoadUserData()
        {
            // Przykładowe dane użytkownika
            // TODO: Podłączyć do rzeczywistej bazy danych
lub API
            string userName = "Jan Kowalski"; // Pobranie z
bazy danych
            int userProgress = 75; // Przykładowy postęp

            lblWelcome.Text = $"Witaj ponownie,
{userName}!";
            progressDailyTasks.Value = userProgress;
        }
    }
}

```

```
private void btnStartLesson_Click(object sender,
EventArgs e)
{
    // Logika otwierania nowego okna lekcji
    LessonForm lessonForm = new LessonForm();
    lessonForm.Show();

    // TODO: Załadowanie lekcji z serwera lub bazy
danych
    MessageBox.Show("Dane lekcji zostały załadowane
z serwera.", "Rozpocznij lekcję", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}

private void btnChatWithAI_Click(object sender,
EventArgs e)
{
    MessageBox.Show("Otwieranie czatu AI...", "Czat
z AI", MessageBoxButtons.OK, MessageBoxIcon.Information);
}

private void btnCheckPronunciation_Click(object
sender, EventArgs e)
{
    MessageBox.Show("Analiza wymowy...", "Sprawdź
wymowę", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}
```

ДОДАТОК Б

Програмний код для дизайну форми польськомовного інтерфейсу навчальної платформи із застосуванням адаптивної моделі навчання

```

Form1.Designer.cs (Dizajn formularza)

namespace LanguageLearningApp
{
    partial class Form1
    {
        private System.ComponentModel.IContainer components
= null;
        private Button btnStartLesson;

        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        private void InitializeComponent()
        {
            this.btnStartLesson = new Sys-
tem.Windows.Forms.Button();
            this.SuspendLayout();

            //
            // btnStartLesson
            //
            this.btnStartLesson.Font = new Sys-
tem.Drawing.Font("Arial", 12F, System.Drawing.FontStyle.Bold);
            this.btnStartLesson.Location = new Sys-
tem.Drawing.Point(100, 50);
            this.btnStartLesson.Name = "btnStartLesson";
            this.btnStartLesson.Size = new Sys-
tem.Drawing.Size(200, 50);
            this.btnStartLesson.Text = "Rozpocznij lekcję";
            this.btnStartLesson.UseVisualStyleBackColor =
true;

            this.btnStartLesson.Click += new Sys-
tem.EventHandler(this.btnStartLesson_Click);

```

```
        //
        // Form1
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
        this.ClientSize = new System.Drawing.Size(400,
200);

        this.Controls.Add(this.btnStartLesson);
        this.Name = "Form1";
        this.Text = "Trener językowy";
        this.ResumeLayout(false);
    }
}
}
```