ATD

Software Requirements Specification

1.0

03.05.2025

Андрійченко А.В.

**Історія версій**

**Дата     Опис          Автор**  **Коментарі**

|  |  |  |  |
| --- | --- | --- | --- |
| 04.05.25 | <Версія 1> | Андрійченко А.В. | <Перша редакція> |
|  |  |  |  |
|  |  |  |  |

**Затвердження документів**

Наступну Специфікацію вимог до програмного забезпечення було прийнято та схвалено:

**Підпис        Надруковане ім’я  Назва**  **Дата**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**ЗМІСТ**

[1 Вступ 3](#_Toc199934529)

[1.1 Призначення 3](#_Toc199934530)

[1.2 Область застосування 4](#_Toc199934531)

[1.3 Визначення, акроніми та абревіатури 4](#_Toc199934532)

[1.4 Огляд 5](#_Toc199934533)

[2 Загальний опис 6](#_Toc199934534)

[2.1 Перспектива продукту 6](#_Toc199934535)

[2.2 Функції продукту 6](#_Toc199934536)

[2.3 Характеристики користувача 7](#_Toc199934537)

[2.4 Загальні обмеження 8](#_Toc199934538)

[2.5 Припущення та залежності 9](#_Toc199934539)

[3 Особливі вимоги 10](#_Toc199934540)

[3.1 Вимоги до зовнішнього інтерфейсу 10](#_Toc199934541)

[3.1.1 Інтерфейси користувача 10](#_Toc199934542)

[3.1.2 Програмні інтерфейси 13](#_Toc199934543)

[3.1.3 Апаратні інтерфейси 13](#_Toc199934544)

[3.1.4 Комунікаційні інтерфейси 13](#_Toc199934545)

[3.2 Функціональні вимоги 13](#_Toc199934546)

[3.3 Use Case 17](#_Toc199934547)

[3.4 Класи та об’єкти 17](#_Toc199934548)

[3.5 Нефункціональні вимоги 18](#_Toc199934549)

[3.6 Проектні обмеження 19](#_Toc199934550)

[3.7 Вимоги до логічної бази даних 19](#_Toc199934551)

[4. Моделі аналізу 21](#_Toc199934552)

[4.1 Діаграми послідовності 21](#_Toc199934553)

[4.2 Діаграми діяльності 22](#_Toc199934554)

[4.3 Діаграми класів 23](#_Toc199934555)

[4.4 Діаграми компонентів 24](#_Toc199934556)

# **1 ВСТУП**

## 1.1 Призначення

Цей документ описує специфікацію вимог до програмного забезпечення (SRS) для системи автоматизованого перекладу текстів, яка забезпечує переклад документів за допомогою моделей обробки природної мови (NLP). Система призначена для користувачів, які потребують швидкого, точного та масштабованого перекладу текстів, зокрема для низькоресурсних мов, у сферах бізнесу, науки, освіти та міжнародної комунікації. Документ визначає функціональні та нефункціональні вимоги, архітектуру системи, а також моделі аналізу для розробки, тестування та впровадження.

## 1.2 Область застосування

Система автоматизованого перекладу текстів розроблена для виконання перекладу документів між різними мовами, за допомогою моделей NLP. Вона забезпечує генерацію альтернативних переформулювань для підвищення якості перекладу, оцінку якості, а також збереження історії перекладів і управління даними користувачів. Система підтримує обробку текстових датасетів для масового перекладу з подальшим збереженням результатів у форматі CSV. Вона надає зручний веб-інтерфейс для введення тексту, вибору мов і моделей, а також для відображення результатів і статистики.

Застосунок призначений для використання в комерційних проєктах, таких як локалізація програмного забезпечення та переклад технічної й юридичної документації, наукових дослідженнях для обробки багатомовних текстів, освітніх цілях для перекладу навчальних матеріалів, а також у міжнародних комунікаціях для швидкого перекладу офіційних документів.

Система підтримує автентифікацію користувачів через JWT-токени, локалізацію інтерфейсу для шести мов і можливість інтеграції через REST API для подальшого розширення функціоналу.

## 1.3 Визначення, акроніми та абревіатури

* NLP – Natural Language Processing (обробка природної мови);
* Transformer – Архітектура нейронної мережі з механізмом самоуваги;
* NLLB – No Language Left Behind, модель для багатомовного перекладу;
* OPUS-MT – Набір відкритих моделей для двомовного перекладу;
* M2M100 – Many-to-Many 100, модель для прямого перекладу між 100 мовами;
* SeamlessM4T – Мультимодальна модель для перекладу тексту та мовлення;
* T5 – Text-to-Text Transfer Transformer, універсальна модель для NLP;
* LaBSE – Language-agnostic BERT Sentence Embedding, модель для семантичних ембедінгів;
* BLEU – Bilingual Evaluation Understudy, метрика оцінки якості перекладу;
* FastAPI – Веб-фреймворк для асинхронної обробки запитів;
* JWT – JSON Web Token, стандарт для автентифікації;
* PostgreSQL – Система управління базами даних;

## 1.4 Огляд

Документ структуровано відповідно до стандарту SRS:

* розділ 2: загальний опис продукту, включаючи його перспективи, функції, характеристики користувачів, обмеження та припущення;
* розділ 3: детальні вимоги, включаючи зовнішні інтерфейси, функціональні вимоги, use case, класи, нефункціональні вимоги та вимоги до бази даних;
* розділ 4: моделі аналізу, включаючи діаграми послідовності, діяльності, класів і компонентів.

# **2 ЗАГАЛЬНИЙ ОПИС**

## 2.1 Перспектива продукту

Система автоматизованого перекладу текстів є сучасним веб-додатком із клієнт-серверною архітектурою, призначеним для виконання перекладу документів між різними мовами, за допомогою моделей обробки природної мови. Вона позиціонується як універсальний інструмент для широкого спектра користувачів, включаючи комерційні організації, які потребують локалізації програмного забезпечення, перекладу технічної, юридичної чи маркетингової документації, а також наукові установи, що працюють із багатомовними текстами в дослідженнях.

Система також знаходить застосування в освітніх платформах для перекладу навчальних матеріалів і в міжнародних організаціях для швидкої обробки офіційних документів, таких як договори чи декларації.

## 2.2 Функції продукту

Метою є розробка веб-додатку для автоматизованого перекладу текстів із використанням сучасних моделей обробки природної мови, що забезпечують переклад між різними мовами. Система підтримує оцінку якості перекладів, обробку великих текстових датасетів і локалізацію інтерфейсу для різних мов. Програмна система складається з двох основних частин: серверної і клієнтської.

Сервісна частина (back-end):

* забезпечує переклад текстів між різними мовами, включаючи прямий і транзитивний переклад через англійську мову як посередника;
* виконує сегментацію тексту на фрагменти для обробки великими NLP-моделями;
* генерує альтернативні переформулювання тексту за допомогою моделі T5 і обирає оптимальний варіант через оцінку семантичної схожості з використанням LaBSE;
* обчислює метрики якості перекладу, а також статистику продуктивності, таку як час виконання, використання пам’яті та CPU;
* обробляє текстові датасети із збереженням результатів у форматі CSV;
* зберігає та отримує дані про переклади та користувачів у базі даних PostgreSQL;
* генерує JWT-токени для автентифікації та авторизації користувачів;
* надає REST API для взаємодії з клієнтською частиною.

Клієнтська частина (front-end):

* надає зручний веб-інтерфейс для введення тексту, вибору вихідної та цільової мов, а також моделі перекладу;
* відображає результати перекладу, метрики якості і статистику продуктивності у вигляді таблиць і графіків;
* забезпечує доступ до історії перекладів користувача з можливістю перегляду та редагування;
* підтримує обробку текстових датасетів із вибором діапазону речень через слайдер;
* надає інтерфейс для авторизації та реєстрації користувачів;
* забезпечує локалізацію інтерфейсу для шести мов (англійська, німецька, фінська, французька, італійська, українська) за допомогою бібліотеки i18next;
* використовує адаптивний дизайн на основі Bootstrap для коректного відображення на десктопах і мобільних пристроях.
* забезпечує зручну навігацію по веб-додатку та взаємодію з сервером через HTTP-запити.

## 2.3 Характеристики користувача

Система автоматизованого перекладу текстів розроблена для різноманітної аудиторії з різними потребами та рівнями технічної підготовки. Перекладачі використовують її для автоматизації рутинних задач, таких як переклад документів чи локалізація контенту, що дозволяє значно скоротити час і підвищити якість завдяки автоматичній оцінці перекладів.

Бізнес-користувачі, зокрема компанії, що розробляють програмне забезпечення чи займаються маркетингом, застосовують систему для перекладу технічної документації, інтерфейсів користувача чи рекламних матеріалів, забезпечуючи швидке адаптування контенту до міжнародних ринків.

Науковці, які працюють із багатомовними текстами, використовують систему для обробки великих обсягів даних, таких як наукові статті чи міжнародні звіти, що полегшує аналіз і порівняння текстів різними мовами. Студенти та викладачі застосовують її для перекладу навчальних матеріалів, сприяючи доступу до освітнього контенту різними мовами.

Адміністратори системи відповідають за налаштування серверної частини, додавання нових моделей чи датасетів, а також моніторинг продуктивності системи.

Для кінцевих користувачів достатньо базових навичок роботи з веб-додатками, тоді як адміністратори та розробники потребують технічної експертизи для управління та розширення системи.

## 2.4 Загальні обмеження

* обчислювальні ресурси: моделі NLP потребують GPU/TPU для ефективної роботи;
* малоресурсні мови: обмежена якість перекладу через брак тренувальних даних;
* інтернет-з’єднання: система потребує стабільного з’єднання для клієнт-серверної взаємодії;
* локалізація: потенційна втрата культурного контексту при перекладі ідіом чи специфічних термінів.

## 2.5 Припущення та залежності

Припущення:

* користувачі мають доступ до сучасних браузерів;
* сервер має достатні обчислювальні ресурси;
* датасети для тестування доступні.

Залежності:

* бібліотеки: Hugging Face Transformers, PyTorch, NLTK, scikit-learn;
* зовнішні сервіси: PostgreSQL для зберігання даних, Hugging Face Model Hub для моделей;
* операційна система: Linux/Windows для серверної частини.

# **3 ОСОБЛИВІ ВИМОГИ**

## 3.1 Вимоги до зовнішнього інтерфейсу

### 3.1.1 Інтерфейси користувача

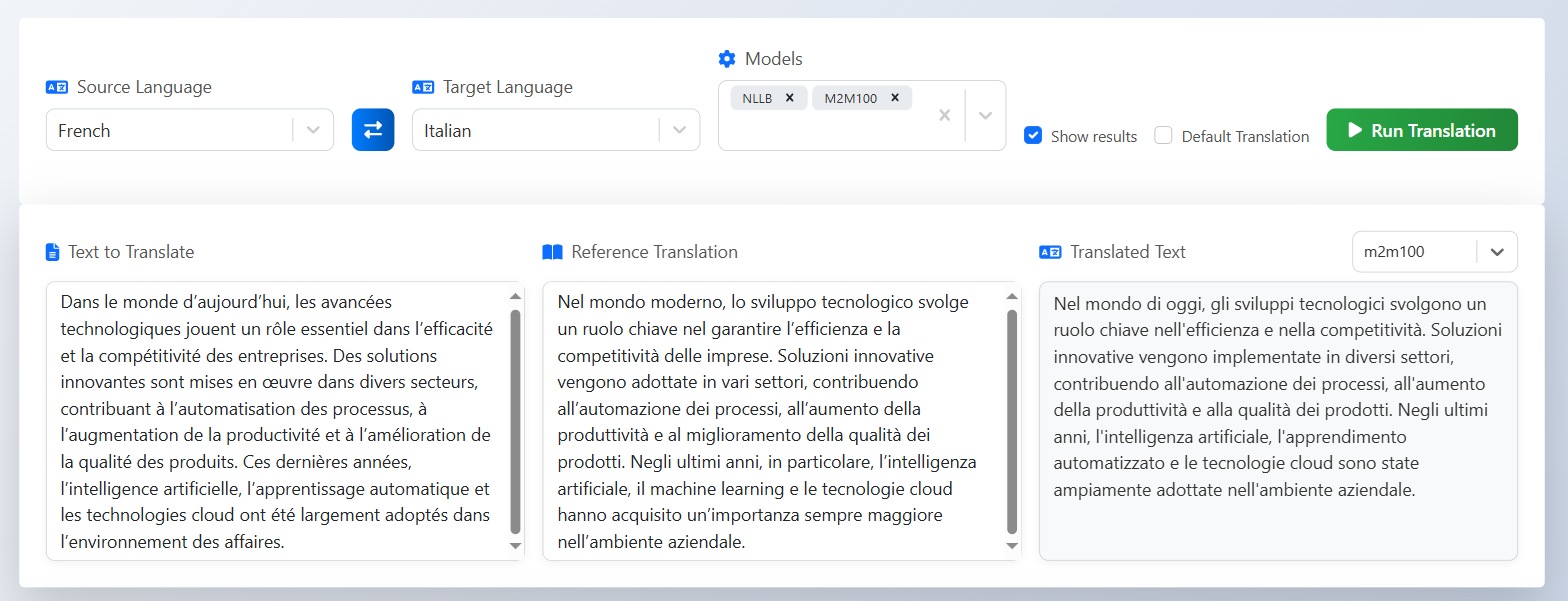


Рисунок 3.1 – Форма для введення тексту, вибору мов і моделей

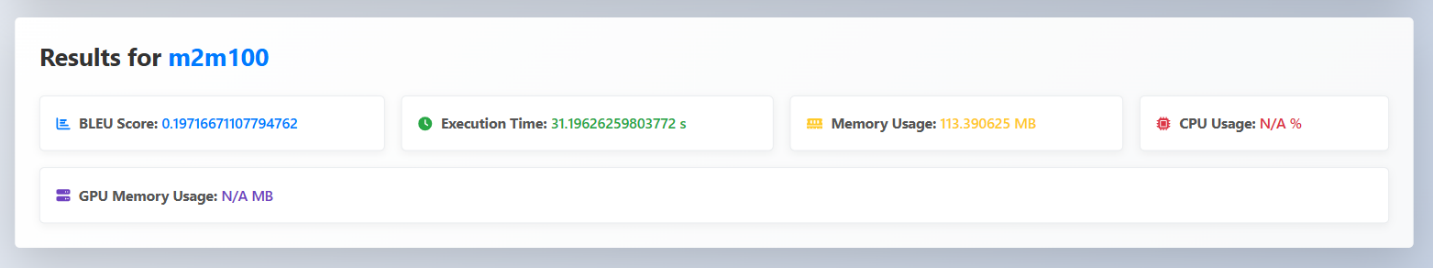


Рисунок 3.2 – Відображення результатів перекладу



Рисунок 3.3 – Табличне та графічне відображення результатів

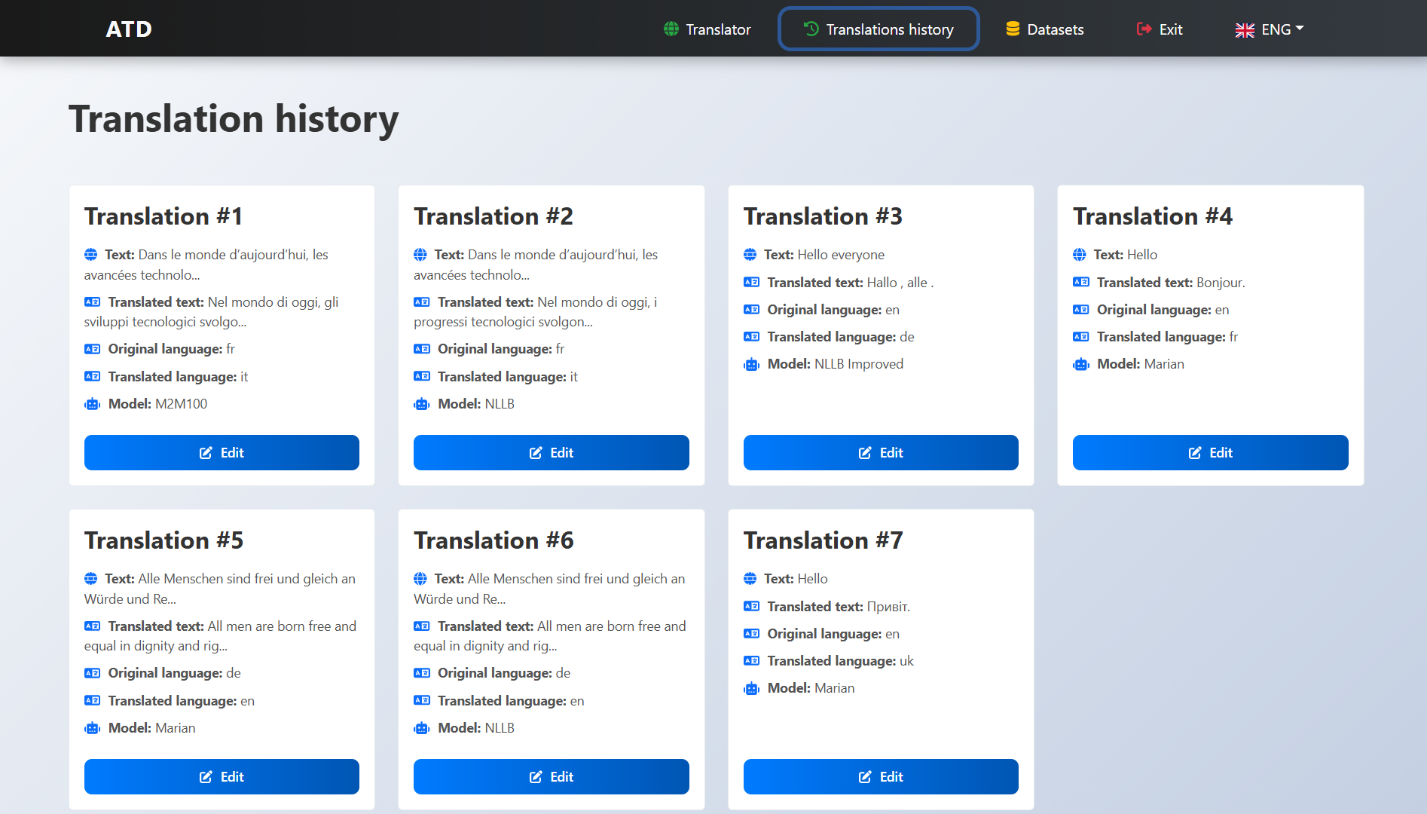


Рисунок 3.4 – Сторінка історії перекладів із можливістю редагування;

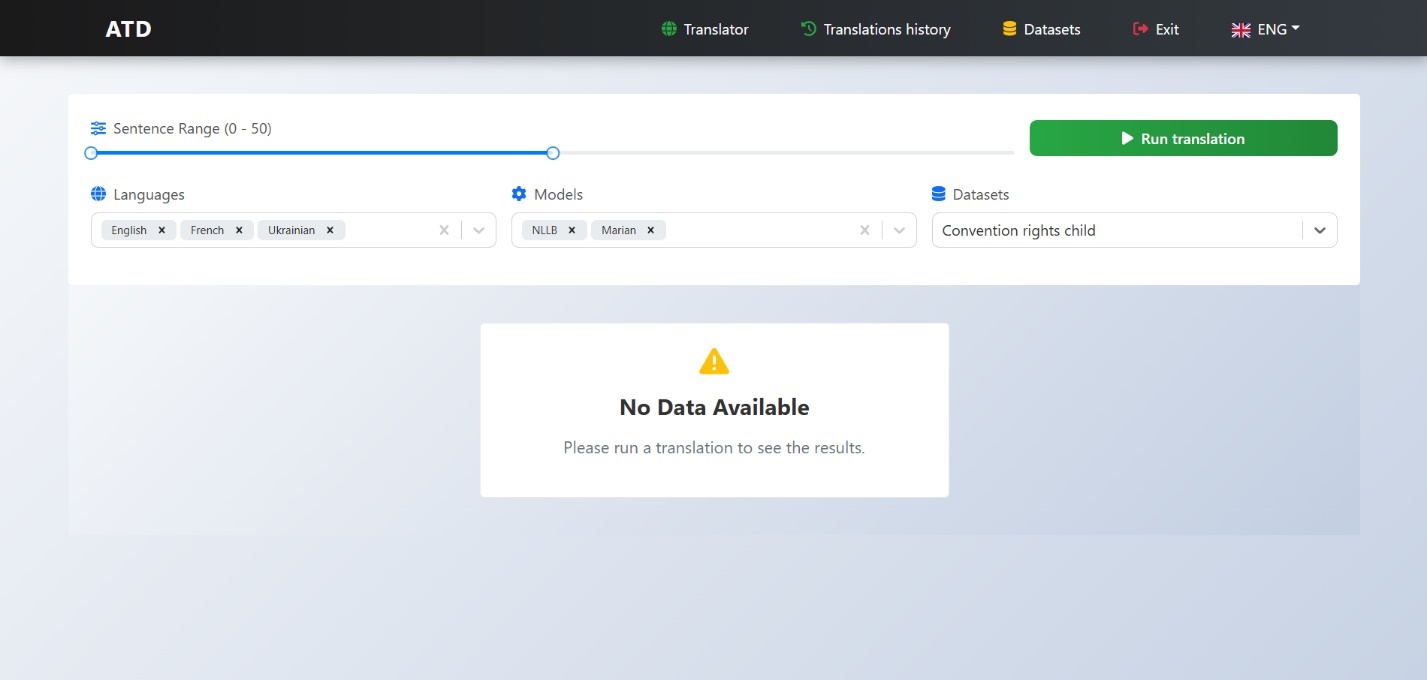


Рисунок 3.5 – сторінка для обробки датасетів із вибором діапазону речень;

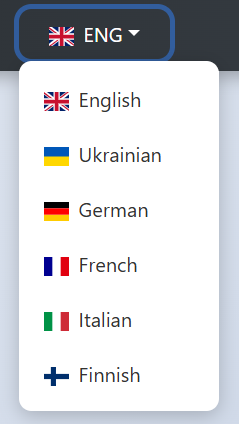


Рисунок 3.6 – локалізація інтерфейсу (6 мов через i18next);

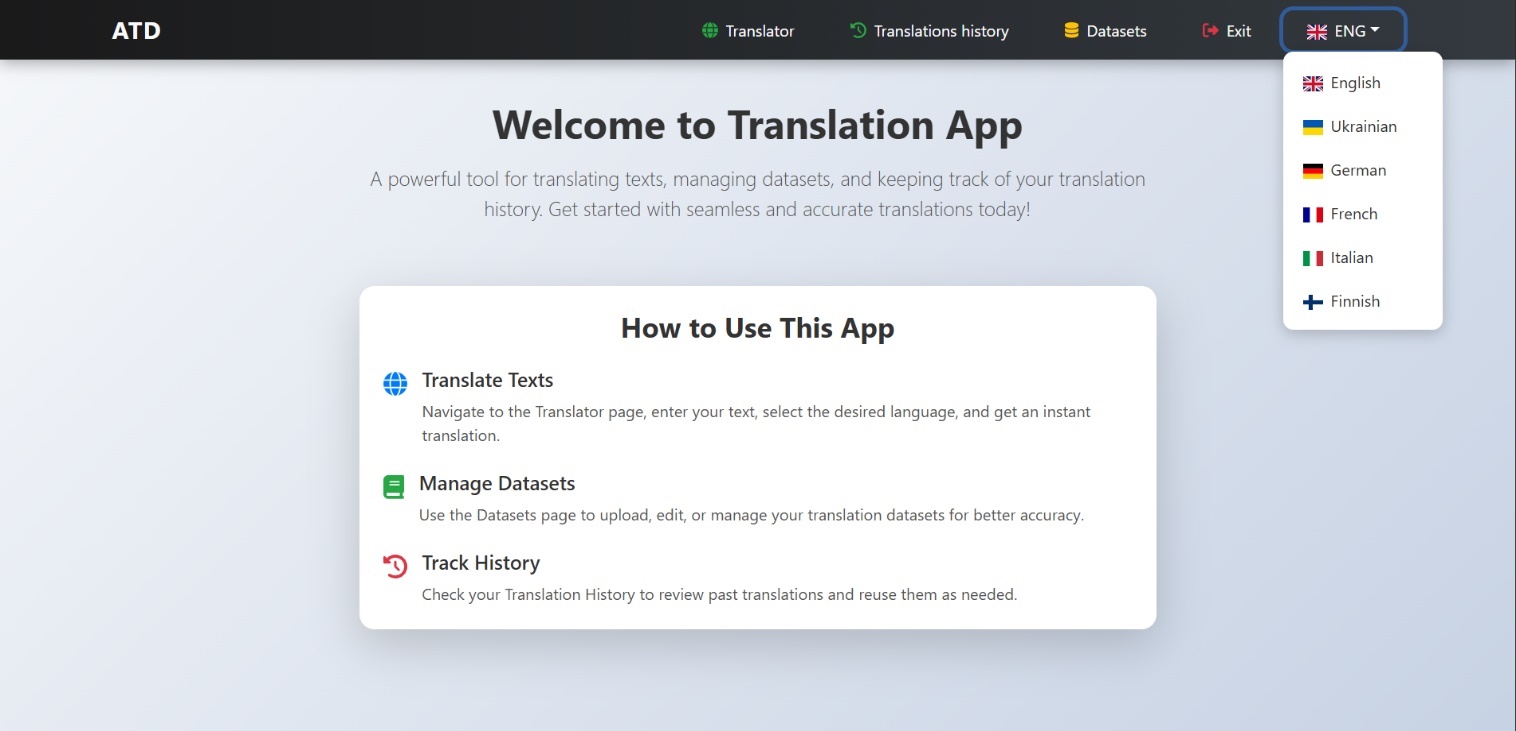


Рисунок 3.7 – Головна сторінка, демонструє Bootstrap для стилізації, адаптивний дизайн для десктопів і мобільних пристроїв.

### 3.1.2 Програмні інтерфейси

REST API (FastAPI) для взаємодії між клієнтом і сервером:

* POST /register: виконання реєстрації;
* POST /authenticate: авторизація користувача;
* POST /logout: вихід користувача;
* POST /datasets: виконання перекладу датасету;
* GET /datasets/names: отримання назв датасетів;
* GET /translations/{user\_id}: отримання перекладів користувача;
* POST /translations: виконання перекладу;
* GET /languages: отримання мов;
* GET /models: отримання назв моделей;

Інтеграція з Hugging Face Transformers для завантаження моделей.

### 3.1.3 Апаратні інтерфейси

* cервер із GPU/TPU для обробки NLP-моделей;
* підключення до мережі для REST API та бази даних.

### 3.1.4 Комунікаційні інтерфейси

* HTTPS для безпечної передачі даних;
* CORS для обмеження доступу до клієнтських адрес;
* JWT для автентифікації запитів.

## 3.2 Функціональні вимоги

3.2.1 Переклад тексту

а) Introduction – програма повинна надавати можливість користувачу виконувати переклад тексту між різними мовами з використанням моделей NLP;

б) Inputs – користувач вводить текст у відповідне поле, обирає вихідну та цільову мови, а також модель перекладу через веб-інтерфейс і натискає кнопку "Перекласти";

в) Processing – система сегментує текст на фрагменти, виконує переклад за допомогою обраної моделі. Результат зберігається в базі даних PostgreSQL;

г) Outputs – відображення перекладеного тексту у веб-інтерфейсі та повідомлення про успішне виконання перекладу;

г) Error Handling – повідомлення про неправильно введені дані (наприклад, порожній текст, некоректна мова) або недоступність обраної моделі.

3.2.2 Оцінка якості перекладу

а) Introduction – програма повинна надавати можливість оцінювати якість перекладу;

б) Inputs – користувач виконує переклад тексту, система автоматично ініціює оцінку якості;

в) Processing – система обчислює BLEU-метрику, порівнюючи переклад із референтним текстом (якщо доступний). Додатково збирається статистика продуктивності. Дані зберігаються в базі даних;

г) Outputs – відображення метрик якості і статистики у веб-інтерфейсі у вигляді таблиць або графіків;

д) Error Handling – повідомлення про недоступність сервера для обчислень.

3.2.3 Обробка текстових датасетів

а) Introduction – програма повинна надавати можливість користувачу обробляти текстові датасети для перекладу;

б) Inputs – користувач обирає датасет, вихідну та цільову мови, модель перекладу, діапазон речень через слайдер і натискає кнопку "Обробити";

в) Processing – система завантажує датасет, сегментує його на фрагменти, виконує переклад за допомогою обраної моделі, обчислює метрики якості (BLEU) і зберігає результати у форматі CSV на сервері;

г) Outputs – повідомлення про успішну обробку датасету, відображення результатів у веб-інтерфейсі у вигляді таблиць і графіків;

д) Error Handling – повідомлення про недоступність датасету, помилки формату файлу або відсутність з’єднання з базою даних.

3.2.4 Реєстрація користувача

а) Introduction – програма повинна дозволяти користувачу реєструватися в системі для доступу до персоналізованих функцій;

б) Inputs – користувач заповнює поля (username, email, password) у формі реєстрації та натискає кнопку "Реєстрація";

в) Processing – система перевіряє правильність введених даних, шукає в базі даних наявність email і, за його відсутності, додає нового користувача з хешованим паролем (passlib) до PostgreSQL;

г) Outputs – перенаправлення на головну сторінку;

д) Error Handling – повідомлення про вже існуючий email, неправильний формат даних або відсутність з’єднання з базою даних.

3.2.5 Авторизація користувача

а) Introduction – програма повинна дозволяти користувачу входити в систему для доступу до своїх даних і функцій;

б) Inputs – користувач вводить email і пароль у форму авторизації та натискає кнопку "Вхід";

в) Processing – система перевіряє введені дані, порівнюючи їх із записами в базі даних PostgreSQL, у разі збігу, генерує JWT-токен для авторизації;

г) Outputs – Перенаправлення до головної сторінки;

д) Error Handling – повідомлення про неправильний email чи пароль або відсутність з’єднання з базою даних.

3.2.6 Перегляд історії перекладів

а) Introduction – програма повинна надавати можливість користувачу переглядати історію своїх перекладів із деталями;

б) Inputs – користувач переходить до розділу "Історія перекладів" у веб-інтерфейсі;

в) Processing – система шукає в базі даних PostgreSQL записи про переклади, пов’язані з ID користувача, і повертає список із деталями (вхідний текст, переклад, мови, модель, метрики);

г) Outputs – сторінка з історією перекладів, відображена у вигляді карток;

д) Error Handling – повідомлення про відсутність перекладів для користувача або недоступність бази даних.

3.2.7 Вибір мов для перекладу

а) Introduction – програма повинна надавати можливість користувачу обирати вихідну та цільову мови для перекладу;

б) Inputs – користувач обирає мови зі списку (наприклад, англійська, українська, французька) через випадаюче меню у веб-інтерфейсі;

в) Processing – система перевіряє доступність обраних мов для вибраної моделі та зберігає вибір у базі даних як частину запиту на переклад;

г) Outputs – повідомлення про успішний вибір мов і відображення їх у формі перекладу;

д) Error Handling – повідомлення про непідтримувані мови для обраної моделі.

3.3 Use Case

Для відображення функціональності програмної системи була створена Use Case діаграма (див. рис. 3.8).

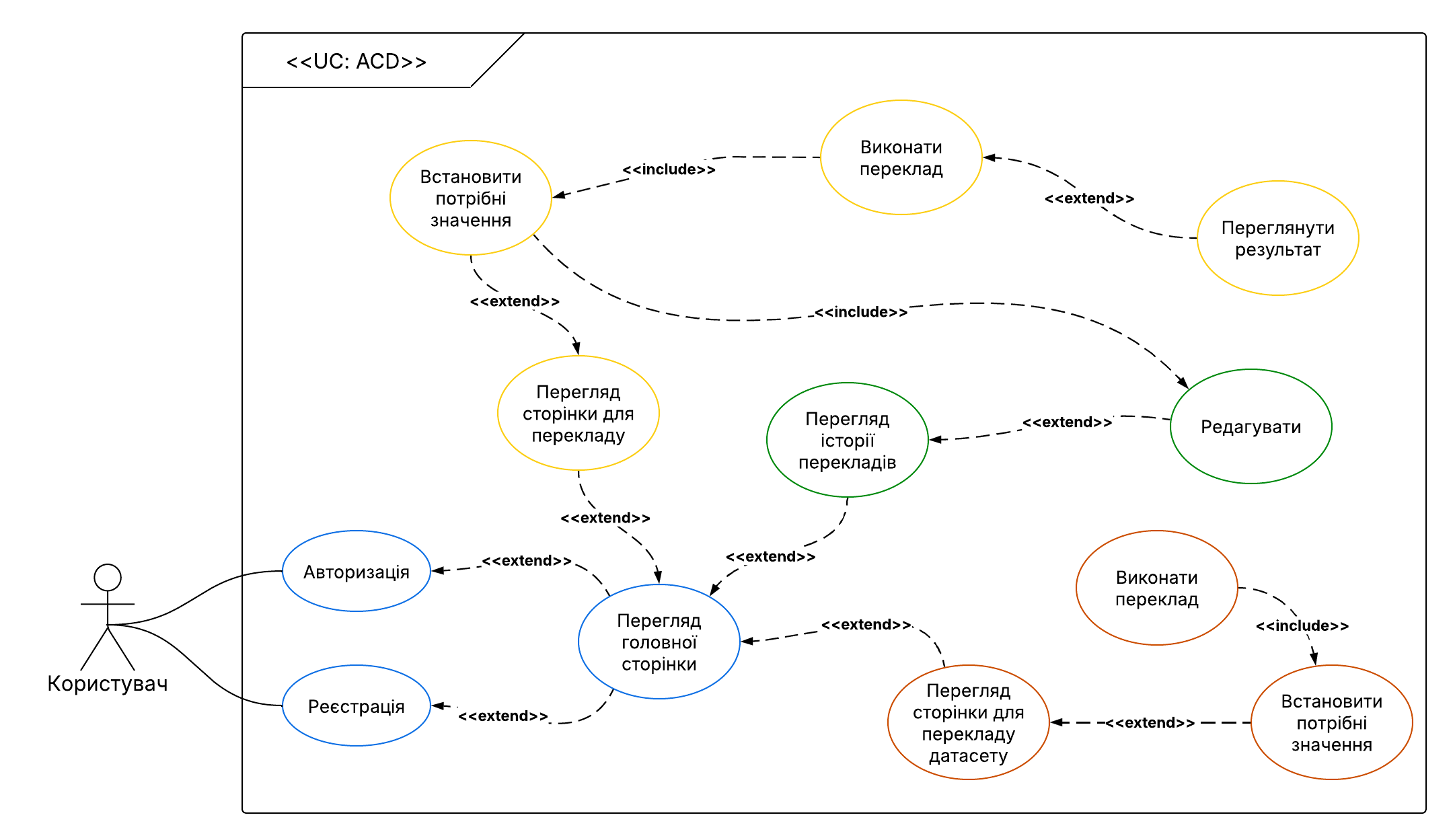


Рисунок 3.8 – Use Case діаграма проекту

3.4 Класи та об’єкти

3.4.1 User

Нижче представлені поля для класу User:

user\_id: Integer

username: String

email: String

password String

3.4.2 Model

Нижче представлені поля для класу Model:

model\_id: Integer

name: String

version: String

3.4.3 Translation

Нижче представлені поля для класу Translation:

translation\_id: Integer

user\_id: Integer

model\_id: Integer

file\_path: String

source\_lang: String

target\_lang: String

is\_default\_translation: Boolean

text\_size: Integer

blue\_score: Float

execution\_time: Float

memory\_usage: Float

cpu\_usage: Float

gpu\_memory\_usage: Float

3.5 Нефункціональні вимоги

* продуктивність: система повинна обробляти до 100 запитів на переклад за хвилину на сервері з GPU;
* масштабованість: підтримка до 1000 одночасних користувачів;
* безпека: використання HTTPS, JWT, CORS для захисту даних;
* доступність: час роботи системи – 99.9% (з урахуванням планового обслуговування);
* локалізація: інтерфейс підтримує 6 мов (en, de, fi, fr, it, uk);
* портативність: система працює на Linux/Windows серверах.

3.6 Проектні обмеження

* апаратні обмеження: потреба в GPU/TPU для NLP-моделей;
* обмеження даних: якість перекладу залежить від доступності тренувальних корпусів;
* технологічні обмеження: використання бібліотек Hugging Face, PyTorch, FastAPI.

3.7 Вимоги до логічної бази даних

На основі ER-діаграми (рис. 3.9):

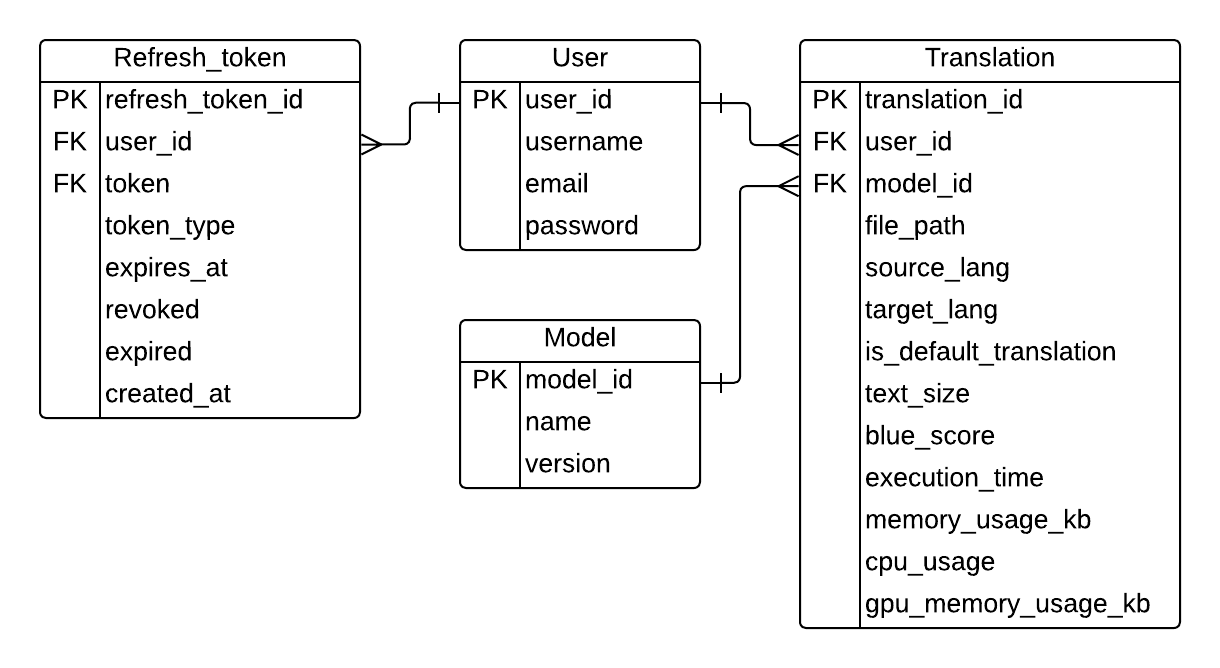


Рисунок 3.8 – ER-діаграма проекту

Сутність User містить ідентифікаційні дані користувачів, зокрема унікальний ідентифікатор, ім’я користувача, електронну пошту та пароль.

Сутність Translation зберігає інформацію про кожен переклад: шлях до збереженого файлу, мови оригіналу та перекладу, ознаку, чи є переклад основним, обсяг тексту, оцінку якості, час виконання та використання апаратних ресурсів. Вона пов’язана з конкретним користувачем і мовною моделлю.

Сутність Model зберігає метадані мовних моделей, включаючи унікальний ідентифікатор, назву моделі та її версію.

Сутність Refresh\_token реалізує механізм оновлення сесій автентифікації. Вона зберігає токени авторизації, тип токена, дату створення, термін дії, а також статуси відкликання та завершення дії. Токен пов’язаний із відповідним користувачем.

**4. МОДЕЛІ АНАЛІЗУ**

4.1 Діаграми послідовності

На діаграмі послідовності, що виконана для цього додатку, можна побачити взаємодію між різними компонентами системи (див. рис 4.1).

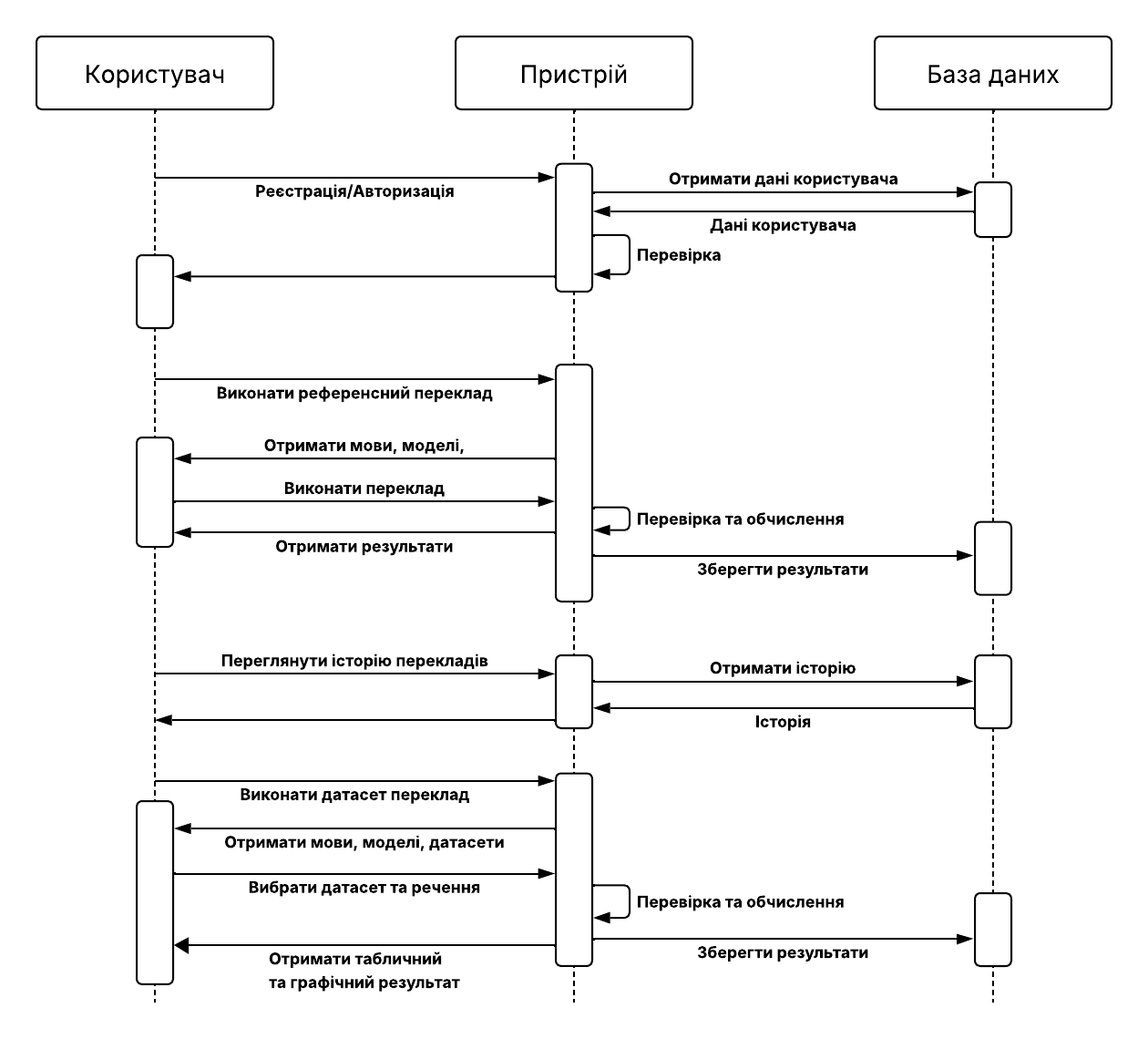


Рисунок 4.1 – Діаграма послідовності проекту

4.2 Діаграми діяльності

На діаграмі діяльності, що представлена для даного додатку, можна побачити послідовність дій, що відбуваються в процесі взаємодії з системою (див. рис 4.2).

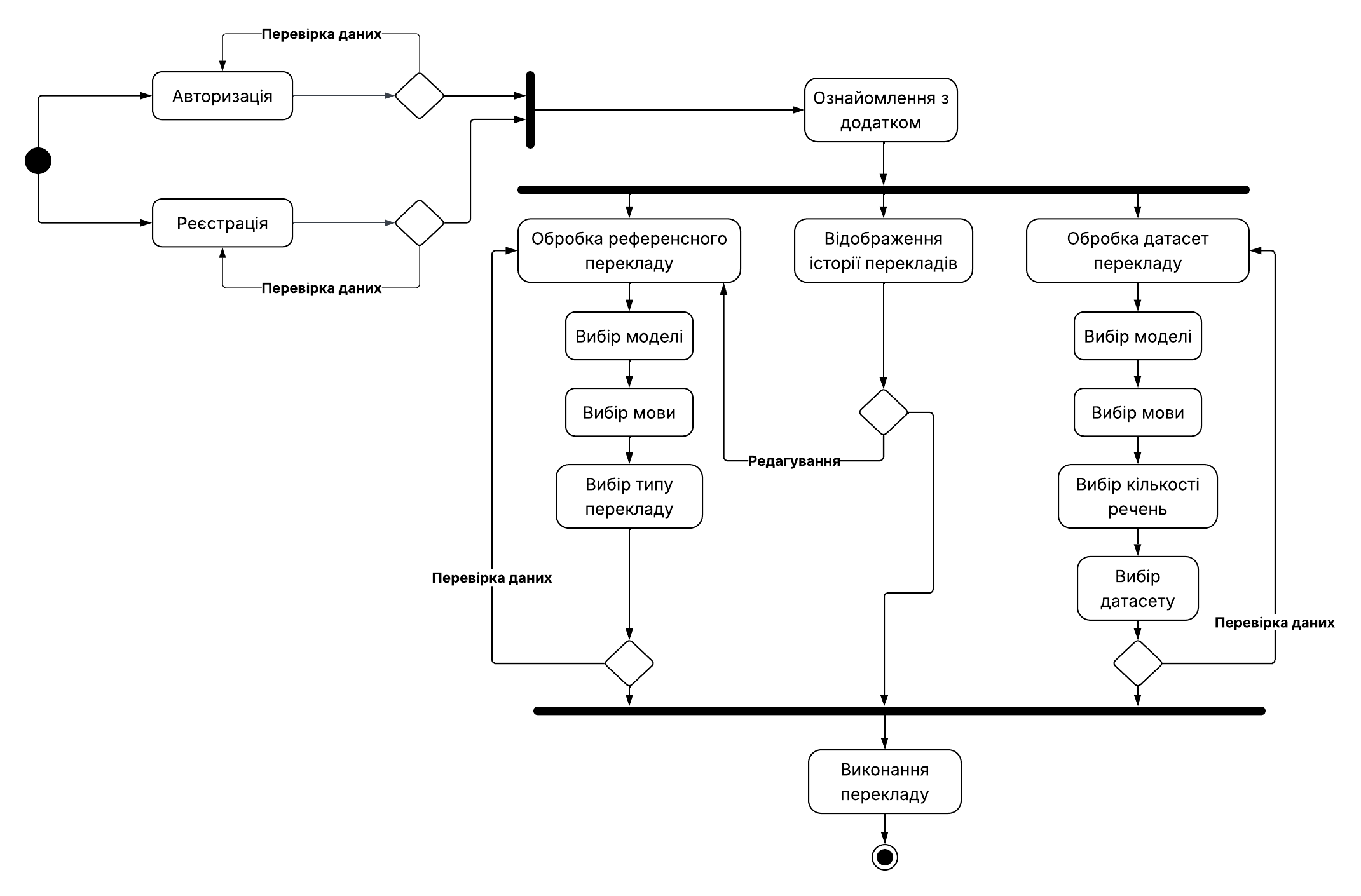


Рисунок 4.2 – Діаграма діяльності проекту

4.3 Діаграми класів

Для відображення взаємодії класів та інтерфейсів програмної системи була створена діаграма класів (див. рис. 4.3).

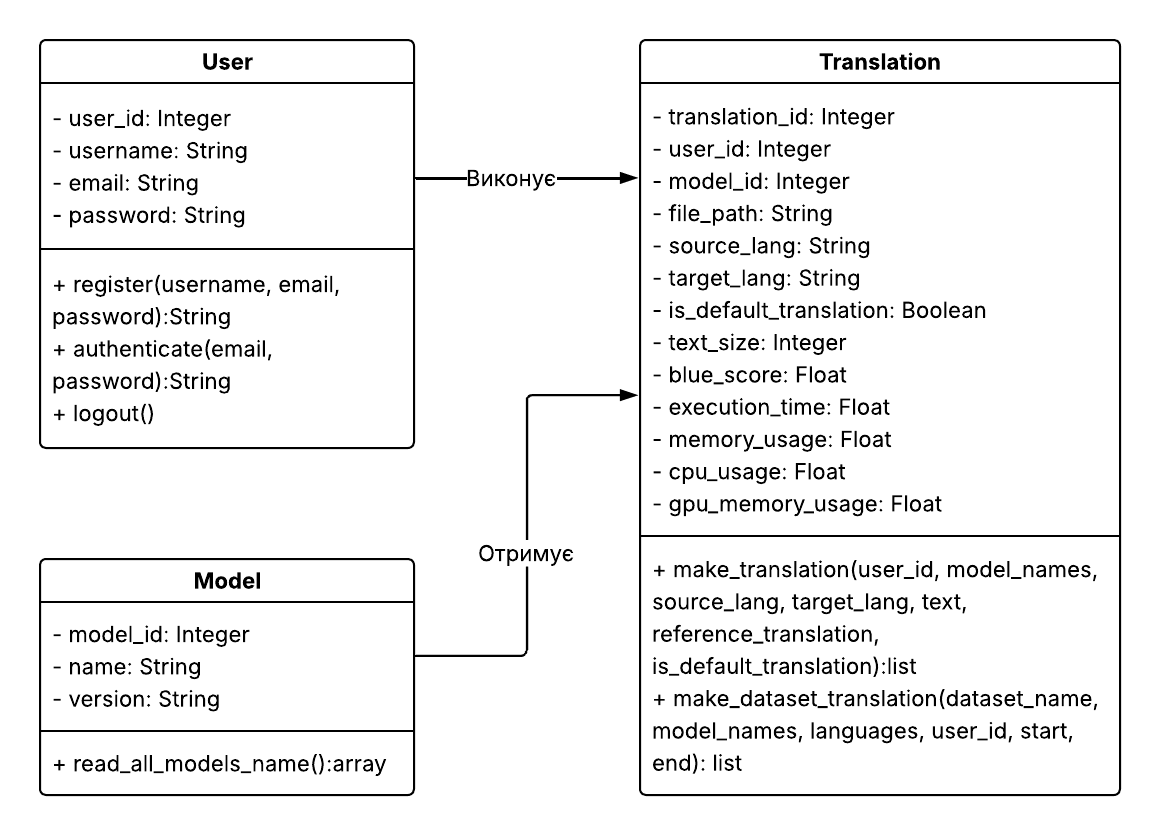


Рисунок 4.3 – Діаграма класів проекту

## 4.4 Діаграми компонентів

Для того щоб візуалізувати структуру та організацію системи у вигляді пакетів, що містять споріднені класи та інші елементи програми та зрозуміти, які частини системи залежать одна від одної та як вони пов'язані між собою була розроблена діаграма пакетів (див. рис. 4.4).

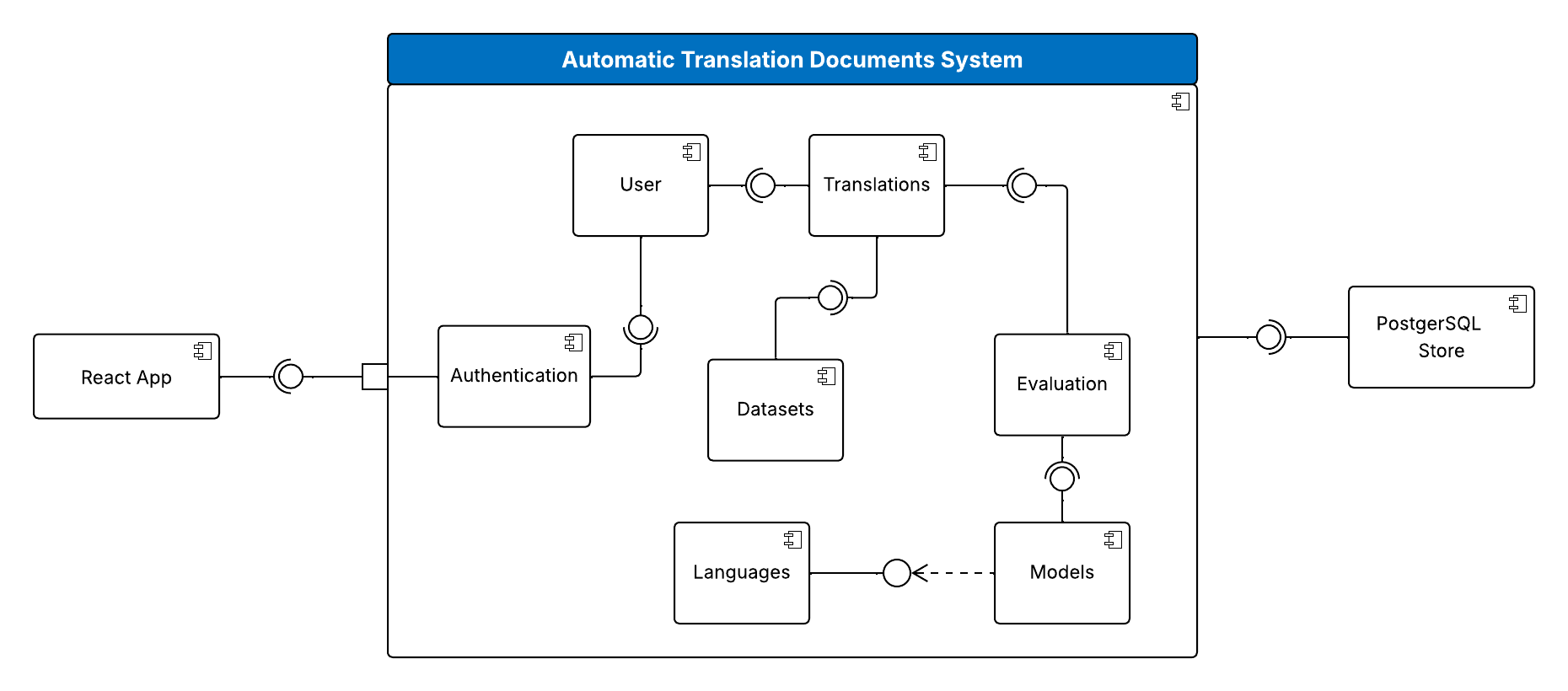


Рисунок 4.4 – UML діаграма компонентів серверної частини   
проекту