

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

«Система автоматизації перекладу»

Виконав студент групи ПЗ-20-1
ДАНЧЕНКО Владислав Валерійович

Керівник роботи:
ЛОБАНЧИКОВА Надія Миколаївна

Рецензент:
ГОДЛЕВСЬКИЙ Юрій Олександрович

Житомир – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«26» лютого 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: ДАНЧЕНКО Владислав Валерійович

Керівник роботи: ЛОБАНЧИКОВА Надія Миколаївна

Тема роботи: **«Система автоматизації перекладу»**,

затверджена Наказом закладу вищої освіти від **«23» лютого 2024 р., №74/с**

Вихідні дані для роботи: **об'єкт дослідження – процес побудови системи автоматизації перекладу, предмет дослідження – технології, методи та алгоритми розробки системи автоматизації перекладу текстів.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Надія ЛОБАНЧИКОВА	27.02.2024р.	27.02.2024р.
2	Надія ЛОБАНЧИКОВА	18.03.2024р.	18.03.2024р.
3	Надія ЛОБАНЧИКОВА	15.04.2024р.	15.04.2024р.

РЕФЕРАТ

Випускна робота бакалавра складається з комп'ютерної програми автоматизованого перекладу тексту (системи САТ) та пояснювальної записки. Пояснювальна записка складається з 109 сторінок, 47 рисунків, 1 таблиці, 4 додатків та 25 літературних джерел.

Основною метою цієї роботи є розробка комп'ютерної програми САТ Tool для автоматизації перекладів.

У роботі визначено ключові завдання, пов'язані з розробкою комп'ютерної програми для автоматизації перекладу, проаналізовано існуючі засоби автоматизації перекладу та розроблено програму для комп'ютерного перекладу текстів.

Проведено аналіз існуючих інструментів, для того, щоб виявити їхні переваги, недоліки та можливості для подальшого вдосконалення програми, що розробляється.

Під час виконання кваліфікаційної роботи нами визначено сценарії роботи програмного продукту, а також подано загальну структуру системи, описано алгоритми взаємодії між компонентами, розроблено відповідні діаграми.

Проведено аналіз та дослідження існуючих методів та рішень у сфері програмного перекладу. В результаті проведених досліджень було розроблено додаток для автоматизації та спрощення перекладу писемних текстів.

Результати дослідження можуть бути використані локалізаційними або перекладацькими компаніями для прискорення виконання поставлених завдань.

КЛЮЧОВІ СЛОВА: САТ, JAVA, АВТОМАТИЗАЦІЯ, ЕФЕКТИВНІСТЬ, ПЕРЕКЛАД, ПРОГРАМНИЙ ПРОДУКТ.

					ІПЗ.КР.Б – 121 – 24 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Система автоматизації перекладу	Літ.	Арк.	Аркушів
Розроб.		Данченко В.В.						
Керівник		Лобанчикова Н.М.					3	109
Рецензент		Годлевський Ю.О.				Житомирська політехніка, група ІПЗ-20-1		
Зав. каф.		Вакалюк Т.А.						

ABSTRACT

A bachelor's thesis consists of a computer-aided translation program (CAT system) and an explanatory note.

The explanatory note consists of 109 pages, 47 figures, 1 table, 4 appendices and 25 references.

The main purpose of this work is to develop a CAT Tool computer program for translation automation.

The paper identifies key tasks related to the development of a computer program for translation automation, analyzes existing translation automation tools, and develops a program for computer-aided translation of texts.

The existing tools were analyzed in order to identify their advantages, disadvantages and opportunities for further improvement of the program under development.

During the qualification work, we defined the scenarios of the software product, as well as presented the general structure of the system, described the algorithms of interaction between components, and developed the corresponding diagrams.

We analyzed and researched existing methods and solutions in the field of software translation. As a result of the research, an application was developed to automate and simplify the translation of written texts.

The results of the study can be used by localization or translation companies to speed up the execution of their tasks.

KEYWORDS: CAT, JAVA, SOFTWARE PRODUCT, TRANSLATION AUTOMATION, TRANSLATION EFFICIENCY.

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ПЕРЕКЛАДУ ТЕКСТІВ.....	9
1.1 Постановка задачі.....	9
1.2 Аналіз аналогів програмного продукту	10
1.3 Вибір архітектури та визначення вимог САТ системи	16
1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	17
Висновки до першого розділу.....	19
РОЗДІЛ 2. ПРОЕКТУВАННЯ САТ СИСТЕМИ.....	20
2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи	20
2.2 Розробка бази даних системи.....	25
2.3 Проектування та реалізація алгоритмів роботи системи	28
2.4 Реалізація додатку САТ системи перекладу текстів	33
Висновки до другого розділу	41
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З САТ СИСТЕМОЮ ПЕРЕКЛАДУ ТЕКСТУ	42
3.1 Порядок встановлення та налаштування системи	42
3.2 Структура інтерфейсу САТ системи перекладу тексту	45
3.3 Тестування роботи програмного продукту.....	48
Висновки до третього розділу.....	53
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДОДАТКИ	58

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Інтерфейс програмування додатків.

CAT – Система автоматизації перекладу.

CRUD – Create Read Delete Update.

GHz – Гігагерц.

GUI – Графічний інтерфейс користувача.

JDK – Набір розробника Java, необхідний для розробки та запуску Java програм.

Jar – Розширення executable файлу програми Java.

MT – Машинний переклад.

MVC – Архітектурний патерн програмування.

ORM – Об'єктно реляційний менеджмент.

БД – База даних.

МБ – Мегабайт.

ОС – Операційна система.

САП – Система автоматизації перекладу.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Системи автоматизованого перекладу (CAT) дуже актуальні в сучасному світі. Зростаюча потреба у швидкому, точному та ефективному перекладі текстів різними мовами стає важливим для компаній, які мають міжнародні партнерські відносини або здійснюють свою діяльність на міжнародному рівні. Системи CAT дозволяють підтримувати термінологічну послідовність, прискорювати процес перекладу за рахунок використання попередньо перекладених сегментів, а також дозволяють тісно співпрацювати перекладачам та редакторам у процесі своєї діяльності. Крім того, CAT-системи стають розумнішими та більш адаптованими до потреб користувачів завдяки використанню сучасних програмних технологій, що робить їх важливим інструментом сучасного мультимовного міжнародного світу.

Метою роботи є створення системи автоматизації перекладу.

Встановлена мета обумовлює наступні завдання:

- Аналіз існуючих систем автоматизації перекладу.
- Оцінка функціональності та ефективності існуючих CAT систем.
- Розробка архітектури нової CAT системи.
- Визначення ключових компонентів системи та їх функціональності.
- Написання коду програми для нової CAT системи.
- Проведення тестування нової CAT системи.
- Оцінка результатів тестування та ефективності роботи нової CAT системи.

Об'єктом дослідження є процес побудови системи автоматизації перекладу.

Предметом дослідження є технології, методи та алгоритми розробки системи автоматизації перекладу текстів.

Методи дослідження: спостереження та інтерв'ювання, розробка настільних клієнтських додатків для ПК для автоматизації процесу перекладу, об'єктно-орієнтоване проектування та моделювання.

Практичне значення розробки: CAT Tool (інструмент для автоматизованого перекладу) має практичне значення в області локалізації та

					ІІЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

перекладу текстів. Отриманий результат є важливим для компаній, які працюють з міжнародним контентом, та мають потреби у швидкій локалізації своїх текстових матеріалів. Впровадження цієї системи може значно скоротити час на виконання перекладу, підвищити персональну продуктивність перекладача та забезпечити високу якість поставленого перекладу, що є запорукою успішної конкуренції на міжнародному світовому ринку.

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ПЕРЕКЛАДУ ТЕКСТІВ

1.1 Постановка задачі

Система автоматизації перекладу (САП) – це програмне забезпечення для перекладу (локалізації) текстів на різних мовах.

Головною метою створення такої системи є автоматизація та спрощення процесів перекладу, зменшення часу, необхідного для виконання перекладу та підвищення точності перекладів завдяки використанню сучасних комп'ютерних технологій.

Для досягнення вказаної мети в рамках даної роботи необхідно вирішити наступні завдання:

- Проаналізувати предметну область та визначити потреби потенційних користувачів. Це допоможе зрозуміти необхідність у конкретних функціях для перекладу і сприятиме правильній постановці задачі кваліфікаційної роботи з автоматизації перекладу.
- Розробити функціональні вимоги до системи автоматизації перекладу.
- Проаналізувати існуючі аналоги САТ інструментів. Визначити їх основні функціональні можливості, переваги та недоліки. На основі цього аналізу сформулювати вимоги до власного програмного продукту.
- Визначити архітектуру системи. Оберемо оптимальний архітектурний підхід для розробки системи автоматизації перекладу та обґрунтуємо його, враховуючи вимоги до масштабованості, надійності та продуктивності системи.
- Вибрати інструментальні засоби та технології реалізації, тобто визначити програмні засоби та технології, які можливо використати для реалізації системи, що розробляється та встановити вимоги до ресурсів комп'ютера, на яких буде розгортатися система.
- Розробити прототип системи САТ Tool та провести тестування на основі попередньо зазначених вимог.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

— Провести тестування функціональних можливостей прототипу, ідентифікацію та виправлення виявлених недоліків та помилок.

— Впровадити систему та оцінити її ефективність у реальних умовах.

Вирішення зазначених пунктів дозволить створити ефективну САТ систему автоматизації перекладу, яка задовольнятиме потреби користувачів та підвищуватиме продуктивність їхньої роботи.

1.2 Аналіз аналогів програмного продукту

Проведемо детальний аналіз існуючих аналогів автоматизації перекладу текстів. У цьому аналізі розглянемо основні інструменти в даній області, їх функціональні можливості, переваги та недоліки.

Для аналізу було обрано такі інструменти як Transifex [1] (рис. 1.1), MateCat [2], BasicCat [3], (рис. 1.2) та OmegaT [4]. По черзі розглянемо функції кожного з них.

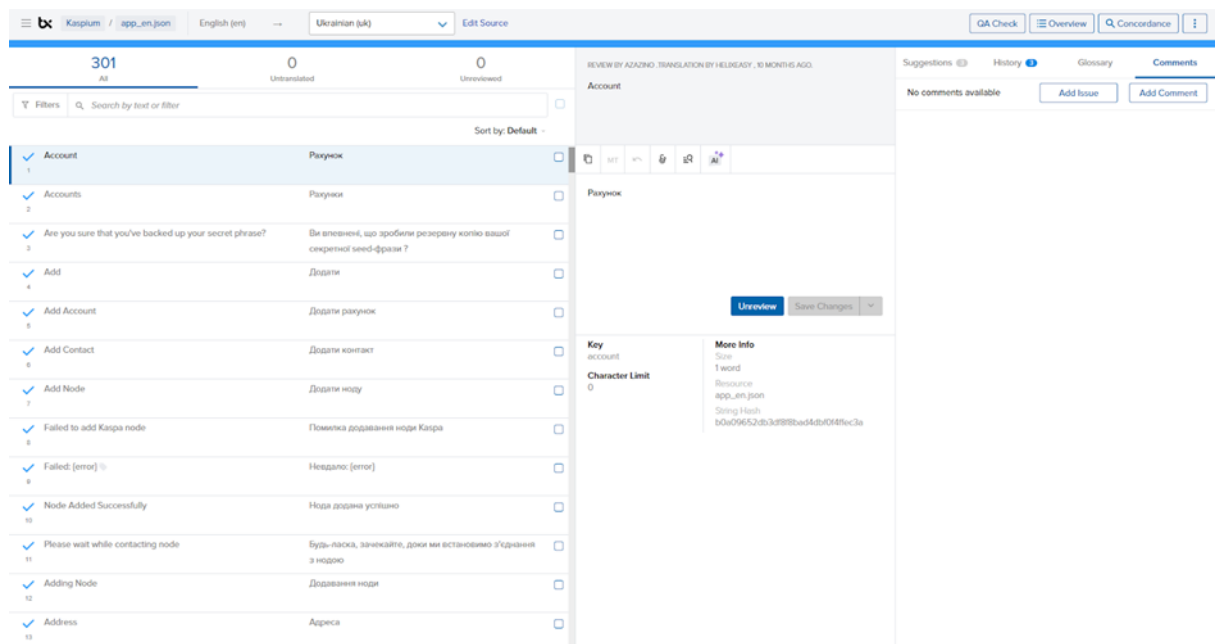


Рис. 1.1. Інтерфейс ПЗ Transifex

Transifex є провідним програмним інструментом у сфері автоперекладу, (рис. 1.1). Ця система надає різноманітні функціональні можливості для ефективного процесу локалізації додатків або контенту від початку до кінця.

Функціональні можливості:

- Автоматизація процесу перекладу.
- Інтеграція з різними середовищами.
- Можливість колаборації.
- Підвищення якості та швидкості локалізації матеріалів.

Переваги:

- Автоматизація процесу локалізації, що зберігає час користувачів.
- Інтеграція з GitHub, BitBucket та підтримка різних форматів файлів, що забезпечує гнучкість у використанні для розробників.

— Можливість спільної роботи над матеріалами та менеджмент проектів перекладу, що дозволяє ефективно залучати команди.

— Налаштовані процеси для підвищення ефективності та якості перекладів.

- Наявність всіх необхідних CAT інструментів та функцій.

Недоліки:

- Обмеження у можливості кастомізації.
- Менша ефективність інтеграцій з деякими сторонніми платформами порівняно з іншими аналогами.

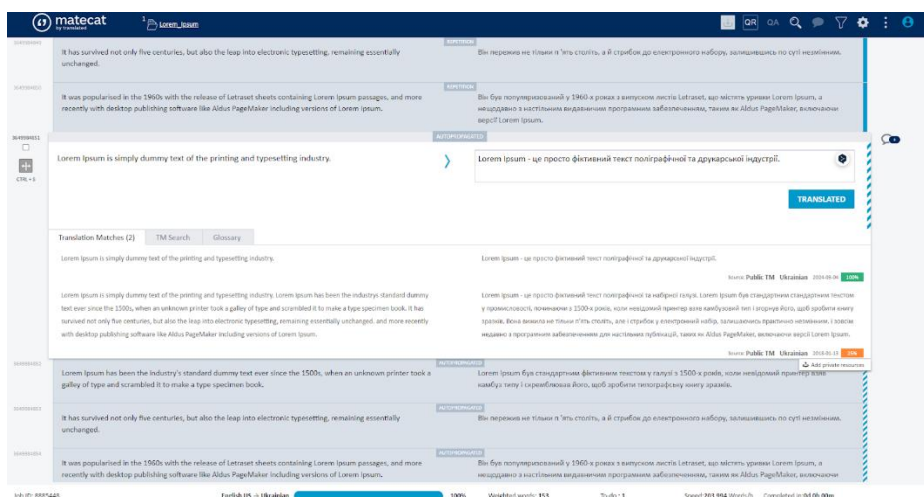


Рис. 1.2. Інтерфейс ПЗ MateCat

MateCat - це безкоштовний та відкритий open-source програмний продукт для автоматизованого перекладу, (див. рис. 1.2).

					ПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Функціональні можливості:

- Зменшення обсягу тексту за рахунок вибору найкращих співпадінь з пам'яті перекладу.
- У платній версії підтримує замовлення професійного людського та машинного перекладу, що дозволяє швидко делегувати будь-які завдання на переклад чи локалізацію текстів зовнішнім платформам.

Переваги:

- Зменшення витрат.
- Підвищена безпека за рахунок відкритого програмного коду.
- Легкість та простота використання, доступність для широкого кола користувачів.

Недоліки:

- Використання функцій машинного перекладу є платним.
- Можливі технічні проблеми у безплатній версії програми.
- Залежність від доступу до інтернету, при використанні її онлайн версії.

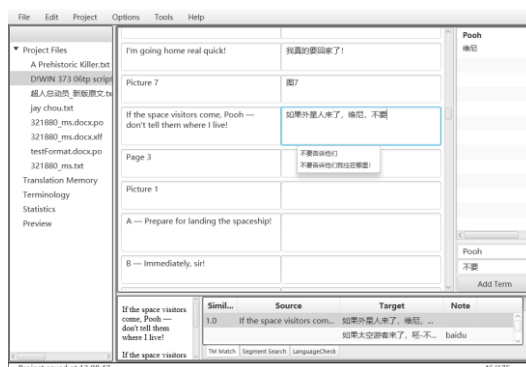


Рис. 1.3. Інтерфейс ПЗ BasicCAT

BasicCAT - це безкоштовний відкритий програмний інструмент для комп'ютерного перекладу, що фокусується на простоті та функціональності, (див. рис. 1.3).

Функції BasicCAT включають:

- Переклад на основі пам'яті перекладу та машинного перекладу.
- Підтримка машинних перекладів від Microsoft, Google, тощо.
- Автокорекція.

					ПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

- Вільне злиття та розділення сегментів.
- Інтеграція з онлайн-словниками.
- Контроль версій за допомогою системи git.
- Машинний переклад.
- Підтримка різних файлових форматів: TMX, TBX, SRX, Gettext Po, IDML, XLIFF, TXT.

- Словник значень.
- Експорт файлів з двомовними абзацами.
- Перевірка тексту на помилки.
- Експорт у формат Word для зовнішнього перегляду.
- Управління термінами.

Переваги BasicCAT:

- Безкоштовний та відкритий програмний код.
- Простота використання та освоєння.
- Широкі можливості перекладу та управління термінами.
- Інтеграція з різними сервісами машинного перекладу.

Недоліки BasicCAT:

- Обмежені можливості у порівнянні з деякими платними програмами.
- Обмежена підтримка деяких файлових форматів.

OmegaT – це безкоштовний CAT інструмент перекладу, який працює на операційних системах Windows, macOS, Linux.

Функції:

- Підтримка понад 30 форматів файлів.
- Підтримка багатофайловості у проєкті.
- Можливість нечіткого зіставлення збережених текстів.
- Підтримка користувальницьких глосаріїв з визначенням форм слів.
- Зіставлення з іншими сегментами тексту.
- Одночасне використання багатьох пам'ятей перекладу.

Можливості:

- Підтримка онлайн сервісів машинного перекладу.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

- Інтеграція з іншими програмами перекладу через спеціальні формати обміну даними, такі як TMX, TTX, TXML, XLIFF, SDLXLIFF.

- Робота з різними алфавітами та підтримка написання тексту справа-наліво.

Недоліки:

- Програма може бути складною для новачків або користувачів без попереднього досвіду роботи з подібними інструментами.

- Може мати проблеми з інтеграцією деяких форматів файлів.

- Відсутність підтримки хмарних сервісів обмежує можливості спільної роботи та обміну даними.

Створимо зведену порівняльну таблицю різних CAT програм.

Таблиця 1.1

Порівняння CAT програм

Параметр	Transifex	MateCat	BasicCAT	OmegaT
Тип ліцензії	Комерційна	Безкоштовна (open-source)	Безкоштовна (open-source)	Безкоштовна (open-source)
Автоматизація	Так	Так	Так	Так
Інтеграція	GitHub, Bitbucket, різні формати файлів	Обмежена (у безкоштовній версії)	Інтеграція з онлайн-словниками	Понад 30 форматів файлів
Колаборація	Так	Обмежена	Обмежена	Обмежена
Основні функції	Автоматизація перекладу, менеджмент проектів	Пам'ять перекладу, автоматичний переклад	Пам'ять перекладу, машинний переклад, глосарії	Пам'ять перекладу, глосарії, нечітке зіставлення

Параметр	Transifex	MateCat	BasicCAT	OmegaT
Переваги	Ефективність, гнучкість, спільна робота	Зменшення витрат, безпека, простота	Безкоштовність, простота, широкий функціонал	Безкоштовність, підтримка багатьох форматів файлів
Недоліки	Обмежена кастомізація, інтеграційні проблеми	Платні функції машинного перекладу, технічні проблеми	Обмежені можливості, мала кількість підтримуваних форматів файлів	Складність для новачків, проблеми з інтеграцією
Підтримка ОС	Всі ОС	Всі ОС	Всі ОС	Windows, macOS, Linux та інші
Формати файлів	Підтримка різних форматів	Обмежена підтримка	Gettext Po, IDML, XLIFF, TXT	Понад 30
Взаємодія з МТ	Так	Так	Так	Так

Ця таблиця надає загальний огляд основних характеристик та функцій кожної CAT системи, що допомагає порівняти їх можливості, функції та недоліки.

Як висновок, аналізуючи основні аналоги програмного продукту для автоматизованого перекладу, можна зазначити, що кожен з них має свої плюси та мінуси. Для успішного розвитку CAT системи важливо враховувати найкращі

практики та нововведення, що можуть внести покращення у функціональність та продуктивність.

1.3 Вибір архітектури та визначення вимог САТ системи

На основі аналізу програмних продуктів конкурентів визначимо вимоги до майбутньої САТ системи. Функціональними (користувацькими) вимогами додатку, що розробляється є:

- Наявність зручного інтерфейсу користувача.
- Можливість створювати проекти перекладу.
- Здатність завантажувати текстові файли (.txt).
- Наявність інтуїтивного текстового редактору.
- Інтеграція програми з API інтернет перекладачів.
- Можливість пошуку термінів слів за інтернет словниками.

До нефункціональних вимог віднесемо:

- Система має мати достатню продуктивність, щоб виконувати роботу з текстом.
- Система має бути надійною та мати стійкість до збоїв при роботі, надійно зберігати дані користувачів у файлах та базі даних.
- Система має бути доступною широкому колу користувачів, тому вона має підтримувати всі популярні ОС.

Системними вимогами є:

- Підтримка Windows, MacOS та Linux.
- Наявність сучасного процесору вище 1 Ghz, кількість оперативної пам'яті вища за 2 Гігабайти, наявність 500 Мегабайт вільного місця на диску системи.
- Наявність інтернет підключення.
- Встановлена Java 22 на комп'ютері.

Далі нами було розглянуто вибір архітектури для системи автоматизації перекладу. Визначимо основні аспекти архітектури ПЗ та обґрунтуємо вибір конкретних технологій та підходів.

					ПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Однією з вимог до розробки програми САТ є підтримка графічного інтерфейсу користувача, тому нами було використано архітектурний паттерн MVC (Model-View-Controller). Цей паттерн дозволяє відокремити модель (model), представлення даних, тобто графічний інтерфейс (view) та контролер (controller) одне від одного, що сприяє покращенню структури проекту, полегшує тестування окремих компонентів, а також забезпечує зменшення залежностей між різними частинами програми, робить розробку більш гнучкою та підтримуваною.

Програма матиме декілька ключових компонентів та рівнів:

- На рівні даних (data layer) нами було використано Hibernate ORM для надання програмі доступу до БД та зберігання її проектів, файлів, перекладів та інших об'єктів системи. Використаємо сутності (entities) Hibernate для моделювання необхідних об'єктів [5].
- Реалізуємо бізнес-логіку (business logic layer) у класах контролерів (controller) для обробки дій користувача та управління процесами додатку.
- На рівні представлення (presentation layer) використаємо технологію JavaFX для створення графічного інтерфейсу користувача (GUI).

Дане рішення дозволить забезпечити розділення відповідальності між різними компонентами додатку та сприятиме підтримці та розширенню майбутнього ПЗ.

1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Визначимося з технологіями та інструментами для створення нашої САТ системи, а також цілями їх використання. Використаємо такі технології, як:

- Maven (управління залежностями (dependencies) проекту та збірка (build) програми) [6].
- Lombok (зменшення шаблонного коду та підвищення продуктивності розробки) [7].
- Hibernate (робота з базою даних, ORM фреймворк (object-relational mapping)) [8].

					ПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

- SQLite (зберігання даних у базі даних SQLite) [9].
- Rich Text Area (робота з гнучким текстовим полем у інтерфейсі користувача)
- JavaFX (сучасний графічний інтерфейс користувача) [10].
- Мова програмування Java (основний код додатку, багатоплатформеність) [11].
- ControlsFX та GemsFX (додаткові елементи графічного інтерфейсу)

JavaFX є популярним інструментом серед розробників, що полегшує доступ до довідкових ресурсів. Вона має зручне API, широкий і гнучкий функціонал для створення різноманітних графічних компонентів програми. JavaFX підтримує платформи Windows, macOS та Linux, що є для нас необхідним, тому JavaFX підходить на роль інтерфейсу [12].

Вибір Hibernate для роботи з БД та ORM обумовлений високою продуктивністю даної ORM, можливістю використання анотацій для мапінгу об'єктів (класів), автоматичним створенням запитів до БД та іншими перевагами, що сприяють ефективній роботі з базою даних.

Maven є найкращим вибором для управління залежностями та збірки проекту для Java завдяки автоматизації збірки проекту, простоті встановлення залежностей та підтримці різноманітних додаткових плагінів [13].

ControlsFX та GemsFX було обрано для розширення можливостей графічного інтерфейсу нашої програми. Ці бібліотеки мають різноманітні додаткові графічні компоненти, що покращують взаємодію з користувачем та забезпечують функціональність інтерфейсу.

Lombok використовуємо для зменшення шаблонного коду в Java-класах за допомогою анотацій, що автоматизує генерацію стандартних методів, таких як гетери (getters), сетери (setters) та конструктори (constructors) [14]. Це дозволяє зосередитись на реалізації логіки системи, писати менше коду та підвищує продуктивність розробки.

Rich Text Area було обрано для використання гнучкого розширеного текстового поля у графічному інтерфейсі (GUI).

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

На роль бази даних нами було обрано SQLite через її легкість у використанні та простоту [15]. Ця БД ідеально підходить для невеликих проектів та прототипів, дозволяючи ефективно взаємодіяти з даними у програмах, написаних на Java [16].

Як висновок, ці технології нам підходять через їх ефективність, популярність у розробницькому середовищі та забезпечення необхідного функціоналу для нашої САТ системи.

Висновки до першого розділу

Система автоматизації перекладу (САТ) є важливою складовою сучасних інформаційних технологій у сфері лінгвістики та перекладу. Головна мета створення такої системи полягає у підвищенні ефективності та якості перекладу текстів, а також у спрощенні та автоматизації процесів перекладу.

Аналізуючи постановку задачі, нами було визначено ключові аспекти, які необхідно враховувати під час розробки системи, такі як аналіз та розробка вимог, дослідження існуючих САТ аналогів, вибір архітектури та інструментальних засобів.

Аналіз існуючих програмних продуктів у галузі авто-перекладу, таких як Transifex, MateCat, BasicCat та OmegaT, дозволив визначити загальні функції, можливості та обмеження систем САТ.

Визначено MVC (model view controller) як основний архітектурний патерн для розробки програмного продукту САТ Tool, що спрощує розробку та підтримку програмного коду нашої майбутньої САТ системи.

Щодо вибору інструментальних засобів та технологій, було обрано бібліотеки та необхідні інструменти для мови програмування Java, графічного інтерфейсу JavaFX та кросплатформності під операційні системи Linux, Windows та MacOS [17].

Результати аналізу дали чітке уявлення про те, які завдання необхідно вирішити, а також, які інструменти і технології краще використовувати для успішної реалізації поставленої задачі.

					ІПЗ.КР.Б – 121 – 24 – ІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

РОЗДІЛ 2. ПРОЕКТУВАННЯ САТ СИСТЕМИ

2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи

Створимо діаграму та проведемо аналіз варіантів використання нашої САТ системи, (рис. 2.1).

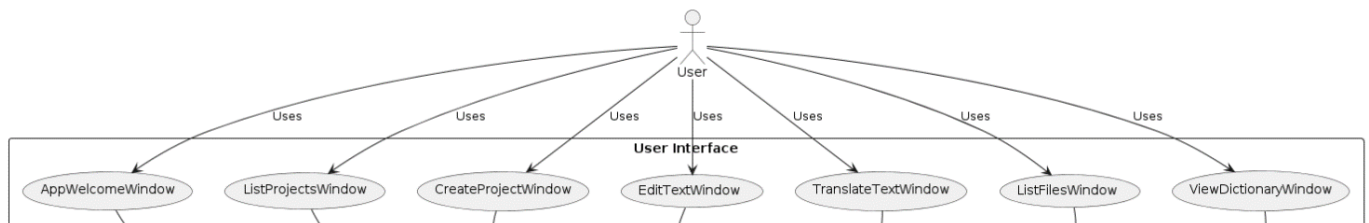


Рис. 2.1. Діаграма варіантів використання

Дана діаграма відображає взаємодію користувача з графічним інтерфейсом. До неї входять такі вікна:

- Привітання (AppWelcomeWindow). Це перше вікно, яке користувач бачить після запуску додатку. Воно містить привітання та інформацію про версію програми.
- Список проектів (ListProjectsWindow). В ньому користувач може бачити всі свої проекти, створити новий проект або вибрати проект для подальшої роботи.
- Створення проекту (CreateProjectWindow). Це вікно дозволяє користувачеві створити новий проект. Воно буде містити форму для введення даних проекту.
- Редагування тексту (EditTextWindow). У даному вікні користувач буде редагувати текст оригіналу та перекладу файлу.
- Переклад тексту (TranslateTextWindow). Це вікно буде дозволяти користувачеві ввести текст для перекладу, обрати мову оригіналу та бажану мову перекладу, обрати перекладач та побачити результати перекладу.
- Список файлів (ListFilesWindow). Користувач зможе працювати з файлами у своєму проекті, це вікно відображатиме список імпортованих файлів.

– Перегляд словника (ViewDictionaryWindow). Дане вікно допоможе користувачеві переглядати та використовувати онлайн словники. У ньому можна буде обрати бажаний веб словник, ввести слово до перекладу та вибрати, де відкривати результат запиту: в браузері чи у локальному вікні програми.

Створена діаграма дозволяє краще зрозуміти майбутні варіанти використання системи. Кожен елемент матиме свою унікальну роль у системі та сприятиме відповідній функціональності додатку.

Визначимо високорівневу структуру взаємодії елементів системи нашого додатку CAT Tool, (рис. 2.2).

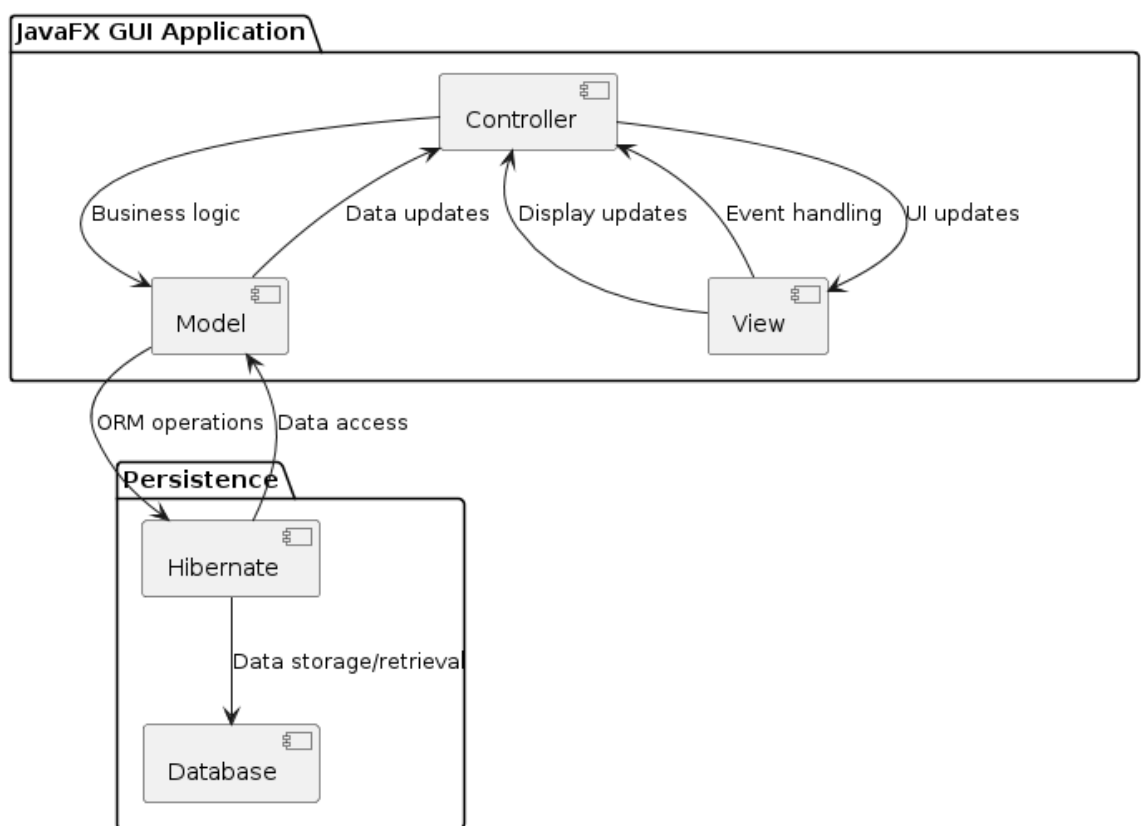


Рис. 2.2. Діаграма компонентів системи

Типовий JavaFX додаток на мові Java складається з таких елементів, як контролер (controller), модель (model), вид (view), бази даних (SQLite database), та ORM persistence фреймворку для взаємодії з базою даних (Hibernate):

- Графічний інтерфейс (JavaFX) відображає графічну частину додатку (вікна) та дозволяє користувачеві взаємодіяти з ними.

- Контролери (controllers) керують логікою відображення даних та обробляють події графічного інтерфейсу, також вони взаємодіють з моделями об'єктів (entities) та Hibernate – для доступу до даних та їх обробки.
- Моделі (models) відповідають за дані та логіку їх обробки у JavaFX.
- Об'єкти (entities) використовуються для зберігання та отримання даних з бази даних SQLite за допомогою ORM фреймворку Hibernate [18].
- Hibernate виконує об'єктно-реляційний менеджмент для забезпечення взаємодії об'єктів програми (entities) з базою даних (database) SQLite.
- База даних (database) зберігає інформацію про об'єкти (entities) нашої програми.

Далі нами було розглянуто діаграму послідовності (рис. 2.3), більш розгорнутий вигляд якої представлено в Додатку А. Дана діаграма описує взаємодію об'єктів у нашій системі та зберігає її порядок та напрямок виконання.

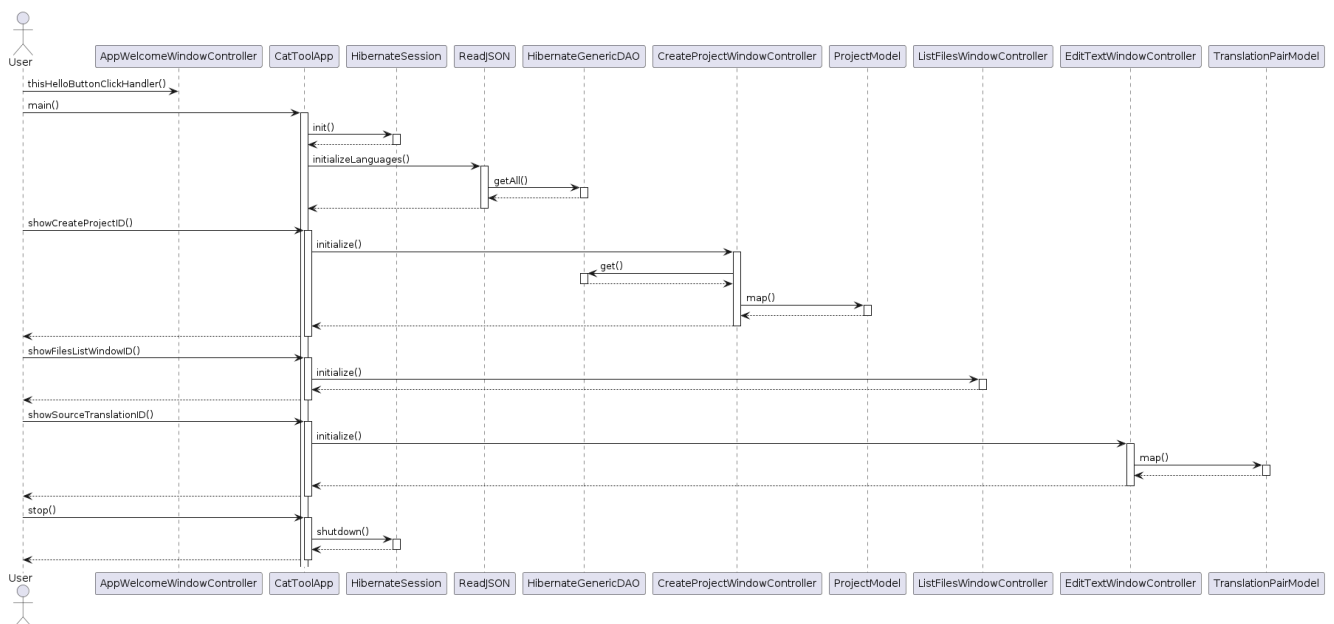


Рис. 2.3. Діаграма послідовності

Першим етапом користувач запускає додаток, що викликає функцію main у нашій САТ програмі. САТ Додаток починає своє виконання.

Другим етапом програма ініціалізує Hibernate, виконується виклик функції init що активує менеджер сесій у HibernateSession. Цей крок важливий для забезпечення доступу до даних додатку у базі SQLite через даний ORM фреймворк.

Третім етапом є ініціалізація необхідних мовних списків, за допомогою функції `initializeLanguages` у компоненті `ReadJSON`. Відбувається завантаження `json` ресурсів з папки `resources`.

На четвертому етапі користувач зможе взаємодіяти з додатком, так як з'являється вікно привітання та списку проектів, а також викликати інші вікна.

Система CAT ініціалізує вікна через функцію `initialize` у відповідному контролері (`controller`). Якщо системі необхідна інформація – вона звертатиметься до бази даних `SQLite` через `HibernateGenericDAO` для отримання необхідних даних. Наприклад за інформацією про наявні проекти, їх параметри, тощо.

На п'ятому етапі користувач завершує роботу і закриває додаток. Завершення викликає функцію `stop`. Це призводить до відключення з'єднання з БД через метод `shutdown` у `HibernateSession`.

Ми розібрали взаємодію користувача з додатком та послідовність дій у CAT системі за допомогою створеної нами діаграми послідовності (Додаток А).

На основі наявних вимог та поставлених задач створимо діаграму класів Java та проаналізуємо найбільш важливі із них. Фрагмент діаграми представлено на рис. 2.4, а повна версія – у Додатку Б.

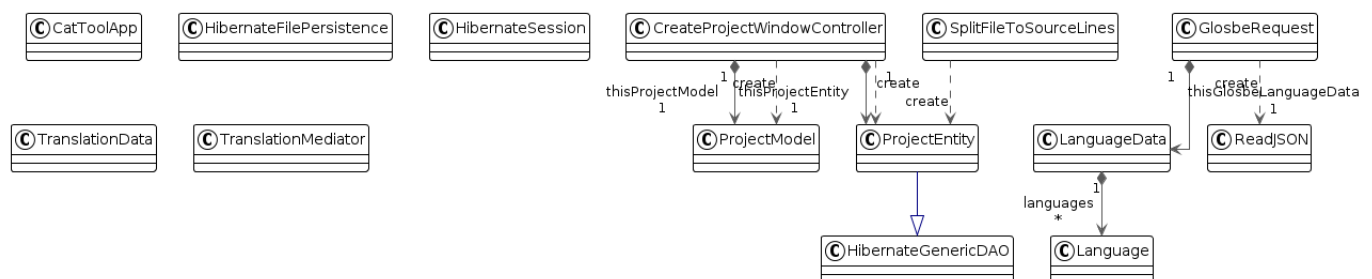


Рис. 2.4. Діаграма класів Java (часткова)

У наявності маємо такі класи:

- Клас `HibernateFilePersistence` відповідатиме за збереження файлів за допомогою `Hibernate` в БД. Метод `saveFileWithHibernate` зберігатиме файл у базі даних з використанням `Hibernate`. Метод `getFileHash` повертатиме хеш-суму файлу.
- Клас `HibernateGenericDAO` є загальним DAO (`data access object`) для взаємодії з базою даних за допомогою `Hibernate`. `HibernateGenericDAO` використовуватиметься для зберігання методів взаємодії з БД. Методи `getAll`, `get`,

merge, mergeOrPersist, delete будуть давати можливість виконувати CRUD операції з базою даних, такі як отримання всіх об'єктів певного класу, отримання об'єкта за його ідентифікатором, збереження, оновлення або видалення об'єкта.

– Клас HibernateSession представлятиме сесію Hibernate, яка використовується для взаємодії з базою даних SQLite. Даний клас використовується для ініціалізації нових сесій. Методи init та shutdown використовуватимуться для ініціалізації та завершення роботи менеджера та всіх активних сесій.

– Клас Application є основою, яка використовується для створення JavaFX додатків. Конструктор Application створюватиме новий екземпляр нашого додатку. Відповідно методи init, start, stop використовуватимуться для його ініціалізації, запуску та зупинки.

– Клас CatToolApp буде представляти основний функціонал додатку Cat Tool. Він матиме методи для показу різних вікон додатку, таких як створення проекту, перегляду списку файлів, перегляду перекладу, тощо.

– Класи AppWelcomeWindowController, CreateProjectWindowController, EditTextWindowController, ListFilesWindowController представлятимуть собою контролери JavaFX, які відповідатимуть за логіку відображення даних та взаємодії графічного інтерфейсу. Вони будуть містити методи для ініціалізації, взаємодії з інформацією, обробки подій та інше.

– Класи TranslationMediator, GlosbeRequest, MyMemoryRequest, PonsRequest, ReversoTranslateRequest, ReadJSON, SplitFileToSourceLines відповідатимуть за різні аспекти перекладу тексту та роботи з потоками інформації, такі як взаємодія з зовнішніми API перекладу, читання даних з JSON, розділення файлів на рядки та інші функції.

Це лише деякі класи (Додаток В) з нашої програми, але кожен з них виконуватиме певні функції для забезпечення функціональності нашої CAT системи.

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

2.2 Розробка бази даних системи

Опишемо структуру зберігання даних у БД нашого додатку CAT Tool. Для виконання роботи будемо використовувати легку та просту базу даних SQLite.

© File
content: blob last_modified: bigint project_id: bigint size: bigint id: integer
hash: varchar(255) name: varchar(255)

Рис. 2.5. Таблиця File

Таблиця File (див. рис. 2.5) зберігатиме:

- зміст файлу (content)
- хеш-суму файлу для визначення унікальності (hash)
- дату та час останньої модифікації файлу (last_modified)
- ім'я файлу (name)
- ключ проекту, до якого відноситься файл (project_id)
- розмір файлу в байтах (size)
- унікальний ключ файлу (id).

© Language
id: integer
code: varchar(255) name: varchar(255)

Рис. 2.6. Таблиця Language

У таблиці Language (див. рис. 2.6) відображатиметься:

- код мови (code)
- назва мови (name)
- унікальний ключ мови (id).

© LineSource
file_id: integer idx: integer source_language_id: integer id: integer
content: varchar(255)

Рис. 2.7. Таблиця LineSource

У таблиці LineSource будуть розміщуватись такі параметри як (див. рис. 2.7):

- зміст рядка джерела (content)
- ідентифікатор файлу до якого відноситься рядок (file_id)
- індекс рядка у файлі (idx)
- ключ мови оригіналу (source_language_id)
- унікальний ключ рядка джерела (id)

© LineTranslation
file_id: integer idx: integer target_language_id: integer id: integer
content: varchar(255)

Рис. 2.8. Таблиця LineTranslation

У таблиці LineTranslation буде міститись (див. рис. 2.8):

- зміст перекладеного рядка (content)
- ключ файлу до якого відноситься переклад (file_id)
- індекс рядка в межах файлу (idx)
- ключ мови перекладу (target_language_id)
- унікальний ключ перекладеного рядку тексту (id).

© Project
id: integer
description: varchar(255) name: varchar(255) sourceLanguage: varchar(255) targetLanguage: varchar(255)

Рис. 2.9. Таблиця Project

Таблиця Project складатиметься з (див. рис. 2.9):

- опису проекту (description)
- назви проекту (name)
- мови джерела (sourceLanguage)
- мови перекладу (targetLanguage)
- унікального ключа проекту (id).

© sqlite_master
type: text name: text tbl_name: text rootpage: int sql: text

Рис. 2.10. Таблиця sqlite_master

Основна master таблиця SQLite складається з (див. рис. 2.10):

- типу об'єкта в базі даних (type)
- назви об'єкта (name)
- назви таблиці (tbl_name)
- кореневої сторінки об'єкта (rootpage)
- колонки SQL запитів створення об'єктів (sql).

Розглянуті діаграми бази даних SQLite відображатимуть структуру таблиць та їхніх колонок, що використовуються для зберігання проектів, файлів, рядків джерела та перекладу, мов у додатку CAT Tool.

2.3 Проектування та реалізація алгоритмів роботи системи

У діаграмі активності виконаємо моделювання процесів та деталізацію логіки нашого CAT додатку (рис. 2.11, Додаток В).

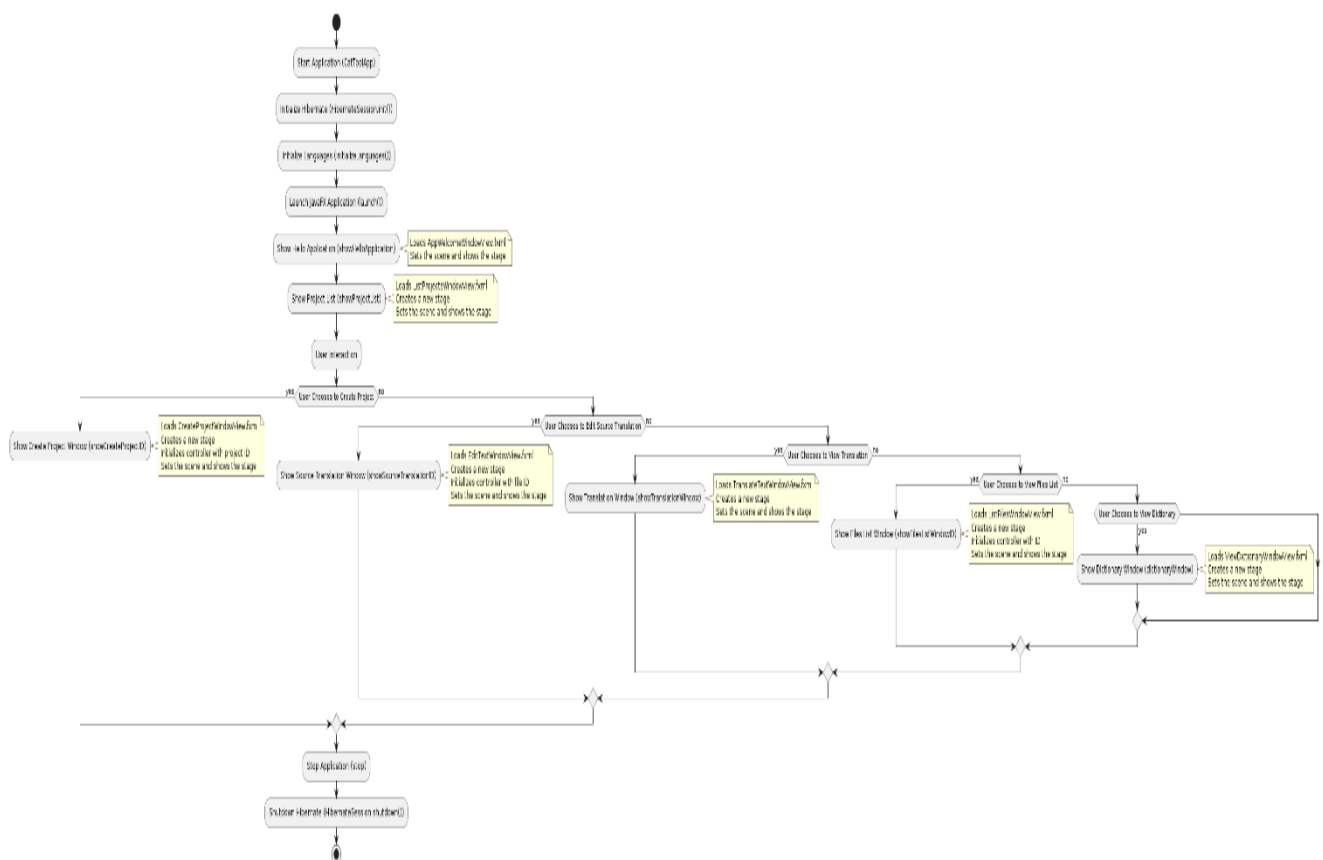


Рис. 2.11. Діаграма активності CAT Tool (Додаток В)

Розглянемо цю діаграму у деталях. Вона складається з таких частин:

- Запуск системи (start application). Цей етап позначає початок виконання програми CAT Tool.
- Ініціалізація Hibernate (init). Тут відбувається налаштування Hibernate та ініціалізація з'єднання з БД.

- Ініціалізація мов (initialize languages). На цьому кроці програма готує необхідні списки мов та завантажує їх з бази даних або з файлу у json форматі з папки resources.
- Запуск інтерфейсу JavaFX (launch). Цей етап відповідає за відображення GUI користувача, відкривається головне вікно програми CAT Tool.
- Відображення вікон (user interaction). На цьому етапі програма показує користувачеві стартове вікно з привітанням AppWelcomeWindow, вікно відображення списку проектів ListFilesWindow, а також доступні дії над ними.
- Якщо користувач вибирає опцію створення проекту - відкривається вікно CreateProjectWindow для створення нового проекту, де потрібно ввести необхідні дані проекту та налаштувати його.
- Якщо користувач вибирає опцію виконання перекладу - відкривається вікно ShowSourceTranslationWindow. Воно складається з таблички, що має колонку індексів, текстові колонки для оригіналу та перекладу тексту, які можливо редагувати.
- При відкритті вікна ViewTranslationWindow, відбувається показ вікна для виконання перекладу тексту за допомогою онлайн перекладачів тексту.
- Якщо користувач відкриває вікно ViewFilesListWindow - відбувається показ вікна зі списком файлів проекту. Дане вікно відображає назву файлу, а також дає змогу додавати та видаляти дані файли з проекту.
- ViewDictionaryWindow. Якщо користувач відкриває це вікно, відбувається відкриття інструменту пошуку значень слова за онлайн словниками. Завдяки цій функції користувач може шукати значення слів та термінів, використаних у тексті.
- Stop Application. На даному етапі користувач завершує виконання програми за допомогою закриття вікон JavaFX. Програма CAT Tool завершує свою роботу та звільняє використані ресурси.
- На етапі shutdown Hibernate завершує свою роботу, зупиняючи з'єднання з базою даних SQLite, Java завершує виконання програми.

					ІІЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Таким чином у діаграмі активності (Додаток В) нами було описано покрокову взаємодію користувача з САТ програмою.

Розглянемо алгоритми роботи та взаємодію елементів програми САТ Tool (рис. 2.12).

Алгоритм взаємодії з файлами:

- Користувач вибирає ресурс (файл).
- Якщо користувач обрав ресурс, то система завантажує його у базу даних і відображає вміст.
- Користувач може редагувати ресурс.
- Якщо користувач вніс зміни - система зберігає їх у базі даних SQLite та відображає результати.
- Якщо користувач не вносив змін, то система просто продовжує відображати вміст.

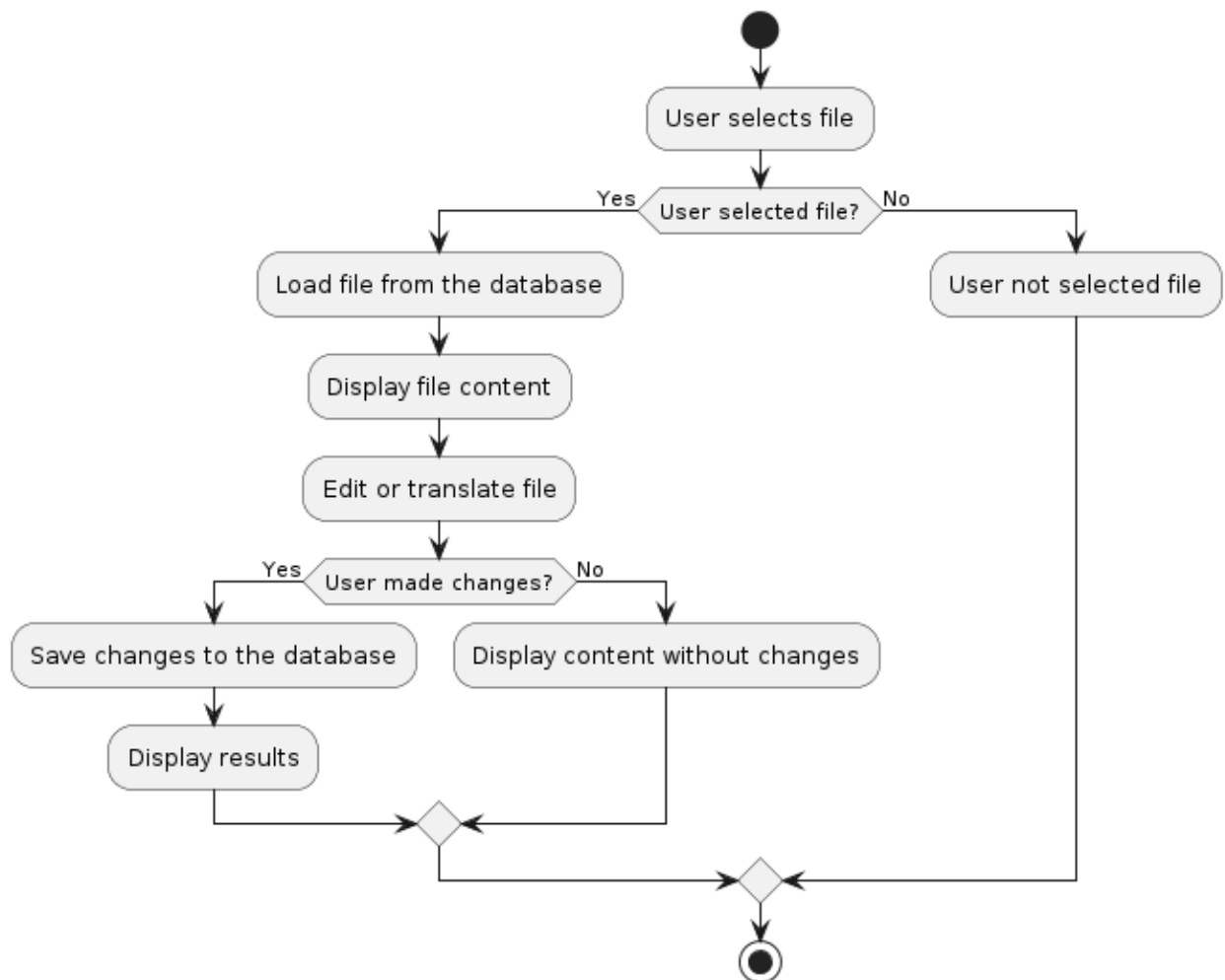


Рис. 2.12. Діаграма активності: взаємодія з файлами

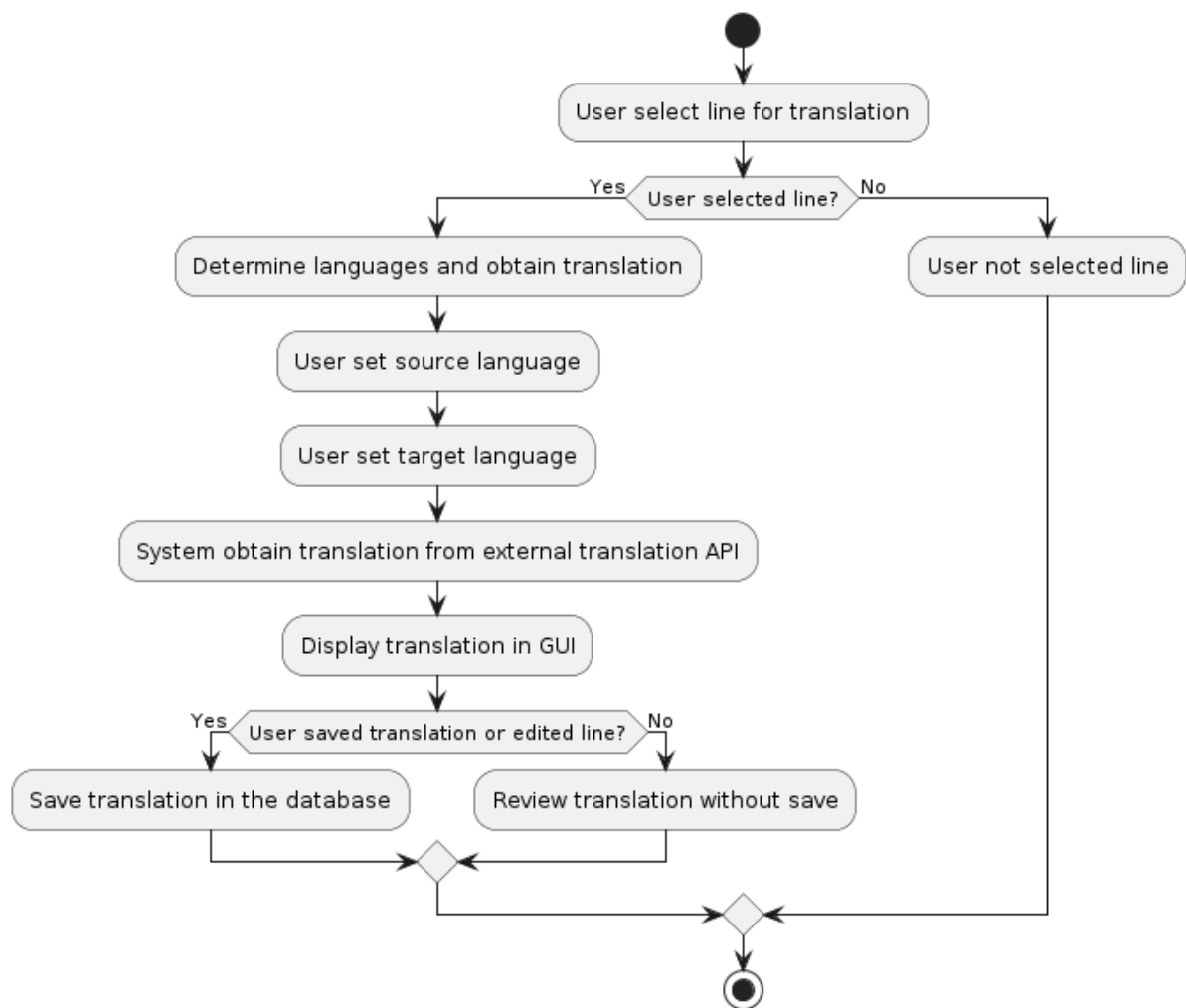


Рис. 2.13. Діаграма активності: редагування тексту

Процес перекладу рядків (рис. 2.13):

- Користувач вибирає рядок для перекладу у графічному інтерфейсі, наприклад, у вікні редагування тексту / перекладу.
- Користувач встановлює мову джерела (оригінал) та мову перекладу.
- Наступним кроком користувач вибирає необхідний йому перекладач та виконує запит на переклад.
- Система отримує переклад обраного рядка з зовнішніх сервісів перекладу.
- Отриманий переклад відображатиметься у відповідному полі графічного інтерфейсу.
- Якщо користувач зберіг переклад або відредагував рядок, то система зберігатиме переклад у базі даних.

– Якщо користувач не зберіг переклад, система дозволить переглянути переклад без збереження.

Взаємодія з зовнішніми API перекладу (рис. 2.14):

– Система надсилає HTTP запит на переклад до API перекладача (Reverso, Glosbe, MyMemory або Pons). Запит містить текст для перекладу та обрані мови (оригінал та переклад).

– Система отримує відповідь у форматі json від API. Вона складатиметься з перекладеного тексту або даних про помилку.

– Система десеріалізує отриману json відповідь у відповідний об'єкт (Entity) за допомогою бібліотеки Jackson.

– Система оновить інтерфейс користувача (GUI) та відобразить перекладений текст.

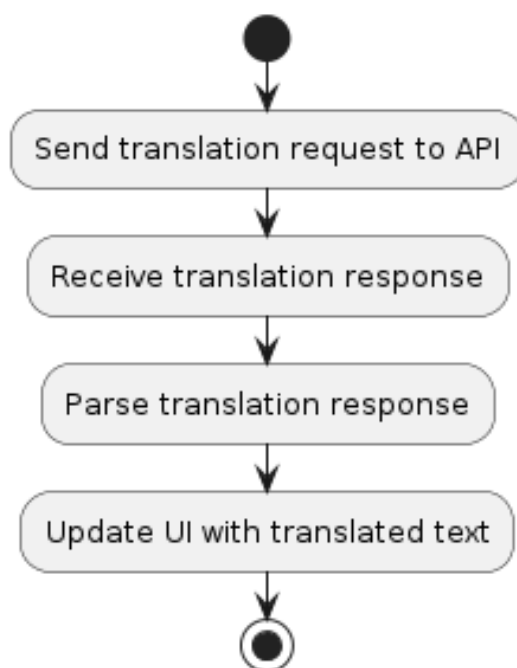


Рис. 2.14. Діаграма активності: переклад тексту

Подані алгоритми та взаємодія елементів САТ системи дозволять користувачеві зручно працювати з додатком, зберігати дані у проєкті, перекладати та редагувати тексти та ефективно виконувати необхідну перекладацьку роботу чи локалізацію.

2.4 Реалізація додатку САТ системи перекладу текстів

Реалізуємо наш запроектований додаток. Програма починає своє виконання у класі CatToolApp. Даний клас розширює клас Application, що дає змогу відображати графічний JavaFX інтерфейс користувача.

```
public class CatToolApp extends Application {}
```

У методі main виконується виклик ініціалізації бази даних через відповідний ORM фреймворк Hibernate, викликається функція initializeLanguages, що завантажує список мов з файлу Json розташованого у папці resources проекту

```
public static void main(String[] args) {  
    HibernateSession.init();  
    initializeLanguages();  
    launch();  
}
```

Далі JavaFX викликає метод launch, що в свою чергу викликає метод start, які є обов'язковими методами класу Application та його похідних.

```
@Override
```

```
public void start(Stage stage) throws IOException {  
    showHelloApplication(stage);  
    showProjectList();  
}
```

Даний метод викликає функції створення вікон showHelloApplication та showProjectList.

```
public void showHelloApplication(Stage thisStage) throws IOException {  
    thisApplication = this;  
    FXMLLoader thisLoader = new FXMLLoader(CatToolApp.class.getResource(  
        "/org/example/cattool/gui/javafx/view/AppWelcomeWindowView.fxml"));  
    Scene thisScene = new Scene(thisLoader.load(), 320, 240);  
    thisStage.setTitle("CAT Tool");  
    thisStage.setScene(thisScene);  
}
```

```

        thisStage.show();
    }

```

У даному прикладі нами було зчитано fxml форму вікна AppWelcomeWindowView за певним шляхом програми, створено нове вікно для даної форми та завантажено дану fxml форму у створене вікно. Також ми можемо встановити назву вікна та його розміри.

Типова JavaFX форма використовує XML форматування:

```

<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.VBox?>
<VBox alignment="CENTER"
    spacing="15.0"
    xmlns="http://javafx.com/javafx"
    xmlns:fx="http://javafx.com/fxml"
    fx:controller="org.example.cattool.gui.javafx.controller.AppWelcomeWindowController">
    <padding>
        <Insets bottom="15.0" left="15.0" right="15.0" top="15.0"/>
    </padding>
    <Label fx:id="thisAppWelcomeText" text="CAT Tool (ver. 1.0)" />
    <Button onAction="#thisHelloButtonClickHandler" text="Hello CAT Tool" />
</VBox>

```

Дана форма має заголовок версії XML, а також визначає тип кодування UTF-8. Далі за нею йдуть строки імпорту необхідних графічних елементів з відповідних шляхів бібліотеки JavaFX.

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.geometry.Insets?>

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

```
<?import javafx.scene.control.Button?>
```

```
<?import javafx.scene.control.Label?>
```

```
<?import javafx.scene.layout.VBox?>
```

Дана форма використовує такі елементи як VBox, padding, Insets, Label, Button.

Кожна форма нашого графічного додатку має свій контролер. Дана форма використовує контролер AppWelcomeWindowController

```
<Vbox_fx:controller="org.example.cattool.gui.javafx.controller.AppWelcomeWindowController">
```

Елементи з нашої форми дзеркально відображаються у нашому контролері у вигляді об'єктів, наприклад даний лейбл thisAppWelcomeText:

```
<Label fx:id="thisAppWelcomeText" text="CAT Tool (ver. 1.0)" />
```

Відображається як:

```
@FXML private Label thisAppWelcomeText;
```

У нашому класі AppWelcomeWindowController:

```
package org.example.cattool.gui.javafx.controller;
```

```
import javafx.fxml.FXML;
```

```
import javafx.scene.control.Label;
```

```
public class AppWelcomeWindowController {
```

```
    @FXML
```

```
    private Label thisAppWelcomeText;
```

```
    @FXML
```

```
    protected void thisHelloButtonClickHandler() {
```

```
        thisAppWelcomeText.setText("Welcome to CAT Tool!");
```

```
    }
```

```
}
```

За допомогою контролера (controller) ми можемо взаємодіяти з вікнами (views) нашого додатку, реагувати на події (events) та вносити зміни у графічну частину нашого додатку CAT.

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

За допомогою параметру `onAction` ми можемо відслідковувати дії над нашими графічними елементами через відповідні методи у контролері.

```
<Button onAction="#thisHelloButtonClickHandler" text="Hello CAT Tool" />
```

При натисканні на дану кнопку з відповідного контролера викликається метод `thisHelloButtonClickHandler`, що змінює текст кнопки на обраний у методі.

```
protected void thisHelloButtonClickHandler() {  
    thisAppWelcomeText.setText("Welcome to CAT Tool!");  
}
```

Для взаємодії з базою даних використовуємо ORM фреймворк `Hibernate`. Для його використання нам необхідно налаштувати файл `hibernate.cfg.xml`, у якому нами було вказано параметри для підключення до нашої бази даних `SQLite`: обраний драйвер, URL для підключення до БД, діалект мови `SQLite`.

```
<property  
name="hibernate.connection.driver_class">org.sqlite.JDBC</property>  
  
<property name="hibernate.connection.url">jdbc:sqlite:sqlite.db</property>  
  
<property name="hibernate.current_session_context_class">thread</property>  
  
<property_name="hibernate.dialect">org.hibernate.community.dialect.SQLiteDialect</property>  
  
<property name="hibernate.hbm2ddl.auto">update</property>  
  
<property name="hibernate.show_sql">true</property>
```

Також у даному файлі нами було вказано `entity` класи, з якими має працювати `Hibernate`:

```
<mapping class="org.example.cattool.database.hibernate.entity.FileEntity"/>  
<mapping class="org.example.cattool.database.hibernate.entity.LanguageEntity"/>  
<mapping class="org.example.cattool.database.hibernate.entity.LineSourceEntity"/>  
<mapping_class="org.example.cattool.database.hibernate.entity.LineTranslationEntity"/>  
>  
  
<mapping class="org.example.cattool.database.hibernate.entity.ProjectEntity"/>
```

`Entity` клас складається з шляху пакету, до якого він відноситься:

```
package org.example.cattool.database.hibernate.entity;
```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

Строк імпорту необхідних функцій:

```
import jakarta.persistence.*;
```

```
import lombok.*;
```

```
import org.example.cattool.database.hibernate.HibernateGenericDAO;
```

Lombok та Hibernate анотацій:

- @Data
- @Entity
- @EqualsAndHashCode
- @Getter
- @RequiredArgsConstructor
- @Setter
- @Table(name = "LineSource")
- @ToString

А також самих полів об'єктів класу:

```
public class LineSourceEntity extends HibernateGenericDAO {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Integer id;  
    @Column(name = "file_id")  
    private Integer fileId;  
    @Column(name = "idx")  
    private Integer index;  
    @Column(name = "source_language_id")  
    private Integer sourceLanguageId;  
    @Column(name = "content")  
    private String content;  
}
```

Lombok анотації дозволяють нам писати менше коду (Додаток Г) для Entity об'єктів. Нами було використано Hibernate анотації:

- @Table(name = "LineSource")

- @Id
- @GeneratedValue(strategy = GenerationType.IDENTITY)
- @Column(name = "file_id")
- @Entity

для того щоб налаштувати роботу Hibernate з нашими об'єктами, та дати змогу Hibernate правильно розпізнати, серіалізувати та десеріалізувати наші дані у базу даних SQLite.

Також всі наші об'єкти (entities) Hibernate розширюють клас HibernateGenericDAO.

```
public class LineSourceEntity extends HibernateGenericDAO {}
```

Завдяки цьому вони можуть використовувати на собі статичні generic CRUD методи для взаємодії об'єкту з БД. Entity може викликати такі методи, як:

- public <T> T mergeOrPersist() {}
- public <T> T merge() {}
- public <T> T get(Integer id) {}
- public <T> T delete() {}
- public static <T> List<T> getAll(Class<T> thisClass) {}

для створення (збереження) об'єкту у базі даних, оновлення значень об'єкту, видалення об'єкту та оновлення об'єкту у базі даних SQLite.

Додаток має функціональність виконання HTTP запитів до API перекладачів, таких як Glosbe, MyMemory, Pons та Reverso Context для перекладу тексту на мову, обрану користувачем. Для цього додаток викликає функцію:

```
public static String callGlosbe(String fromThisLanguage, String toThisLanguage, String thisTextToTranslate) {}
```

Дана функція виконує HTTP запит, що містить у собі текст для перекладу, мову оригіналу та бажану мову перекладу. Додаток отримує відповідь у форматі json та десеріалізує результат у об'єкт за допомогою бібліотеки Jackson, що має відповідну функцію серіалізації та десеріалізації даного формату [19].

```

public static TranslationData parseGlosbeResponse(String thisResponseToParse)
{
    ObjectMapper thisMapper = new ObjectMapper();
    try {
        return thisMapper.readValue(thisResponseToParse, TranslationData.class);
    } catch (Exception thisException) {
        thisException.printStackTrace();
    }
    return null;
}

```

Об'єкт TranslationData за аналогією Entity об'єкту має відповідні Lombok анотації, а також анотації @JsonProperty, для того щоб правильно десеріалізувати дані у відповідні поля класу об'єкту.

```

public class TranslationData {
    @JsonProperty("translation")
    private String translation;
    @JsonProperty("scores")
    private List<Double> scores;
}

```

Для підвантаження списків мов, підтримуваних додатком та зовнішніми перекладачами, додаток читає файли формату json з папки resources нашого додатку.

Для виконання більш складніших запитів до бази даних SQLite, додаток використовує Hibernate Criteria API. Дане API дозволяє будувати складні запити до нашої БД, що не обмежуються CRUD операціями. Розглянемо приклад такого запиту на функції getLineTranslation:

```

public static ArrayList<LineTranslationEntity> getLineTranslation
(Session thisSession, Integer thisFileId) {
    CriteriaBuilder thisBuilder = thisSession.getCriteriaBuilder();
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

```

CriteriaQuery<LineTranslationEntity> thisQuery =
thisBuilder.createQuery(LineTranslationEntity.class);
Root<LineTranslationEntity> thisRoot =
thisQuery.from(LineTranslationEntity.class);
thisQuery.where(thisBuilder.equal(thisRoot.get("fileId"), thisFileId));
return (ArrayList<LineTranslationEntity>)
thisSession.createQuery(thisQuery).getResultList();
}

```

Дана функція виконує запит об'єкта LineTranslationEntity, та вибирає всі об'єкти, чиї fileId рівні thisFileId, та повертає впорядкований список об'єктів даного типу.

У даному проекті використовується інструмент збірки Maven [20]. Даний інструмент є відповідальним за керування залежностями нашого проекту. Для імпорту бібліотек вони повинні бути додані нами у розділ dependencies файлу pom.xml.

Приклад імпорту бібліотеки:

```

<dependency>
<groupId>org.openjfx</groupId>
<artifactId>javafx-controls</artifactId>
<version>21</version>
</dependency>

```

Також версія Java 22 при розробці графічного додатку на платформі JavaFX вимагає вказувати залежності (requirements), експорти (exports) та відкриття відповідних модулів додатку у файлі module-info.java [21]. Приклад:

```

module org.example {
    exports org.example.cattool.gui.javafx.controller;
    opens org.example.cattool.gui.javafx.application to javafx.fxml;
    requires com.fasterxml.jackson.databind;
}

```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до другого розділу

У другому розділі роботи нами було розглянуто та проаналізовано основні аспекти проектування та реалізації додатку CAT Tool. Нами було визначено різні сценарії використання програмного продукту, описано структуру, методи та створено узагальнену схему CAT додатку.

Було розглянуто структуру зберігання даних у базі даних SQLite, виконано аналіз об'єктів CAT додатку. Для роботи з базою даних нами було використано ORM фреймворк Hibernate, а для ефективної взаємодії з даними були реалізовані CRUD операції над ними. Далі нами було детально описано загальні алгоритми роботи CAT системи та взаємодію її елементів між собою.

Також було розглянуто взаємодію з API різних перекладачів за допомогою надсилання HTTP запитів та десеріалізацію Json результатів у відповідні об'єкти (Entities) за допомогою бібліотеки Jackson.

У останньому пункті було детально описано фрагменти коду (Додаток Г) нашої CAT системи.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

**РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З САТ СИСТЕМОЮ
ПЕРЕКЛАДУ ТЕКСТУ**

3.1 Порядок встановлення та налаштування системи

Для запуску розробленої САТ програми необхідно встановити мову програмування Java версії 22 на комп’ютер користувача та запустити файл у форматі .jar. Виконавши подані кроки, ми налаштуємо середовище для запуску нашої Java програми. Схему ресурсів системи представлено на рис. 3.1.

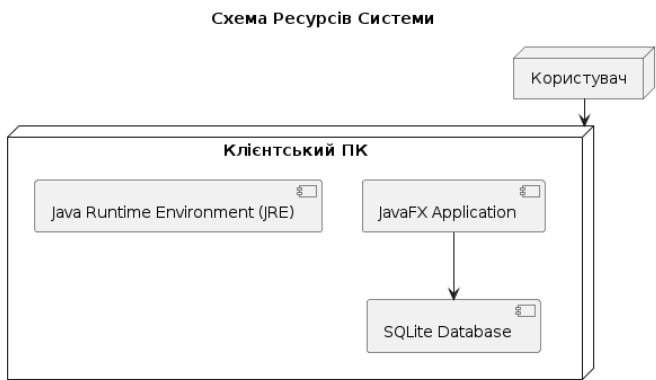


Рис. 3.1. Схема ресурсів системи

Першим кроком нам необхідно завантажити та встановити набір розробника Java (JDK) від компанії Oracle за офіційним посиланням [22] (рис. 3.2).



Рис. 3.2. Завантаження Java

Оберемо версію JDK, що підходить для нашої операційної системи. Натиснемо на посилання та завантажимо версію, що нам підходить.

Далі запустимо завантажений файл установки, оберемо шлях установки, і дочекаємося завершення процесу. Після завершення закриємо вікно інсталяції.

Для перевірки правильності установки JDK виконаємо команду `java -version` (рис. 3.3). Якщо нами було правильно встановлено JDK, у терміналі ми побачимо версію встановленої версії Java.

```
user@user-Nitro-AN515-46: ~ _ □ x
user@user-Nitro-AN515-46:~$ java --version
openjdk 22 2024-03-19
OpenJDK Runtime Environment (build 22+36-2370)
OpenJDK 64-Bit Server VM (build 22+36-2370, mixed mo
user@user-Nitro-AN515-46:~$ █
```

Рис. 3.3. Версія Java

Для запуску Java програми CAT Tool, упакованої у формат .jar, виконаємо наступні кроки:

Відкриємо командний рядок, перейдемо до директорії, де зберігається наш файл .jar за допомогою команди cd та запустимо файл .jar ввівши дану команду (рис. 3.4):

```
user@user-Nitro-AN515-46: ~ _ □ x
user@user-Nitro-AN515-46:~$ java -jar cat_tool.jar █
```

Рис. 3.4. Виконання Java програми.

Також програму CAT Tool ми можемо запустити допомогою подвійного кліку по файлу. Програма відкриється, і ми побачимо відповідний інтерфейс та повідомлення у командному рядку (рис. 3.5).

```
user@user-Nitro-AN515-46: ~/cat_tool
File Edit View Search Terminal Help
user@user-Nitro-AN515-46:~/cat_tool$ java -jar cat-tool.jar
WARNING: sun.reflect.Reflection.getCallerClass is not supported. This will impact per
formance.
Hibernate: create table File (id integer, content blob, hash varchar(255), last modif
ied bigint, name varchar(255), project_id integer, size bigint, primary key (id))
Hibernate: create table Language (id integer, code varchar(255), name varchar(255), p
rimary key (id))
Hibernate: create table LineSource (id integer, content varchar(255), file_id integer
, idx integer, source_language_id integer, primary key (id))
Hibernate: create table LineTranslation (id integer, content varchar(255), file_id in
teger, idx integer, target_language_id integer, primary key (id))
Hibernate: create table Project (id integer, description varchar(255), name varchar(2
55), sourceLanguage integer, targetLanguage integer, primary key (id))
Hibernate: select le1_0.id,le1_0.code,le1_0.name from Language le1_0
Hibernate: select le1_0.id,le1_0.code,le1_0.name from Language le1_0 where le1_0.id=?
Hibernate: insert into Language (code,name) values (?,?)
Hibernate: select last_insert_rowid()
Hibernate: update Language set code=?,name=? where id=?
Hibernate: select le1_0.id,le1_0.code,le1_0.name from Language le1_0 where le1_0.id=?
Hibernate: insert into Language (code,name) values (?,?)
Hibernate: select last_insert_rowid()
Hibernate: select le1_0.id,le1_0.code,le1_0.name from Language le1_0 where le1_0.id=?
Hibernate: insert into Language (code,name) values (?,?)
Hibernate: select last_insert_rowid()
```

Рис. 3.5. Запуск додатку CAT Tool

Під час виконання програма створить базу даних SQLite у тій самій папці, де розміщується основна CAT програма (рис. 3.6).

```

user@user-Nitro-AN515-46: ~/cat_tool
File Edit View Search Terminal Tabs Help
user@user-Nitro-AN515-46: ~/cat_tool x user@user-Nitro-AN515-46: ~/cat_tool x
user@user-Nitro-AN515-46:~/cat_tool$ ls -la
total 123124
drwxrwxr-x  2 user user    4096 чеп  1 05:51 .
drwxr-x--- 38 user user   12288 чеп  1 05:47 ..
-rw-rw-r--  1 user user 126007722 чеп  1 05:08 cat-tool.jar
-rw-rw-r--  1 user user   32768 чеп  1 05:51 sqlite.db
-rw-rw-r--  1 user user    8720 чеп  1 05:51 sqlite.db-journal
user@user-Nitro-AN515-46:~/cat_tool$

```

Рис. 3.6. База даних SQLite

Після початкової ініціалізації відкриється основне вікно програми (рис. 3.7):



Рис. 3.7. Вікно привітання CAT Tool

Переходимо до опису структури інтерфейсу САТ системи перекладу тексту.

3.2 Структура інтерфейсу CAT системи перекладу тексту

Інтерфейс додатку CAT Tool складається з:

- Вікна вітання користувача (відображення версії CAT додатку), (рис. 3.8).



Рис. 3.8. Вікно версії програми CAT Tool

- Вікна списку всіх CAT проектів (рис. 3.9):

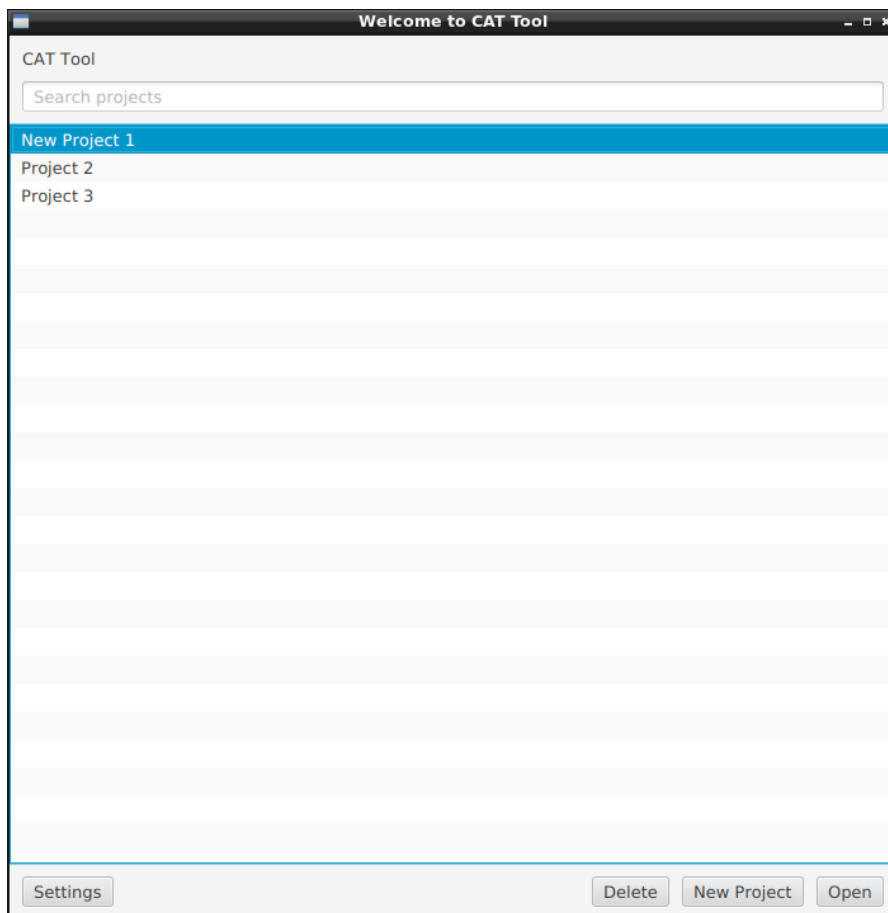


Рис. 3.9. Вікно списку проектів

- Вікна створення нового проекту перекладу (рис. 3.10):

Рис. 3.10. Вікно створення проекту CAT Tool

- Вікна списку ресурсів обраного проекту перекладу (рис. 3.11):

Рис. 3.11. Вікно ресурсів проекту

- Вікна виконання редагування перекладу тексту (рис. 3.12):

CAT Tool		
Index	Source	Translation
0	Adam Asnyk	Адам Асник
1		
2	Daremne żale	Безкорисливі шкоди
3		
4	ISBN 978-83-288-7120-5	ISBN 978-83-288-7120-5 Відомості
5		
6		
7	Daremne żale, próżny trud,	марні смутки, марні зусилля,
8	Bezsilne złorzeczenia!	Безсильні прокляття!
9	Przeżytych kształtów żaden cud	Виживши форм ніякого дива
10	Nie wróci do istnienia.	Він не повернеться до існування.
11		
12	Świat wam nie odda, idąc wstecz,	Світ не поверне вас назад,
13	Zniknionych mar szeregu:	Зниклі мар ряду:
14	Nie zdoła ogień ani miecz	Не переможе ні вогонь, ні меч
15	Powstrzymać myśli w biegu.	Залишати думки в ході.
16		
17	Trzeba z Żywymi naprzód iść,	Треба з Живими йти вперед,
18	Po życie sięgać nowe:	Для життя дістатися до нових:
19	A nie w uwiedłych laurów liść	А не в'язному лавровому листі
20	Z uporem stroić głowę!	Стійко тримайся!
21		

Рис. 3.12. Вікно виконання редагування та перекладу тексту

- Вікна онлайн перекладача тексту (рис. 3.13):

Translation window

Translate

Select Translator:

Glosbe

Language: Source

Polish

Target

Ukrainian

Source:

„Czy przyda się mówić prawdę w żartobliwy i dowcipny sposób? Czy wyśmiewając i wyszydzając zdrożności ludzkie, można je z serc ludzi wyplenić?” Takie pytanie zadaje sobie I. van Leeuwen, pierwszorzędny dziś znawca Arystofanesa i wielki jego wielbiciel; rozchodzi się oczywiście o komedię, o jej ostateczne zadania i cele istotne, tj. o jej wychowawcze, życiochronne, kulturalne działanie, którym się zbliża do wielkiej swej siostrzycy tragedii, spełniać mającej, udoskonalenie człowieka na czele ogółu sztuk pięknych.

Translation:

Чи варто казати правду в жартівливій та смішній формі? Чи можна насміхаючись і знущаючись над людськими недоліками вирвати їх з сердець людей? Таке питання ставить собі І. ван Левен, сьогоднішній знавець Аристофена і його великий поклонник; мова йде, звичайно, про комедію, про її кінцеві завдання та важливі цілі, тобто про її виховні, життєзахисні, культурні дії, якими вона наближається до своєї великої сестри трагедії, яка має втілити, вдосконалювати людину на чолі загальних витончених мистецтв.

Query

Use

Рис. 3.13. Вікно онлайн перекладача тексту

- Вікна САТ словника (пошук слів за онлайн словником у браузері) (рис. 3.14):

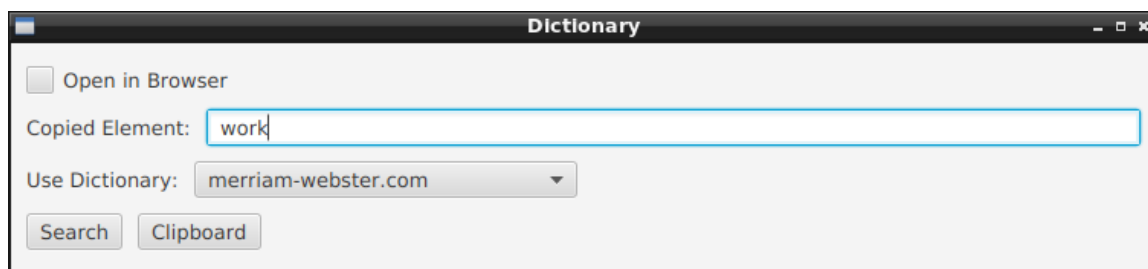


Рис. 3.14. Вікно словника тексту

- А також вікон імпорту та експорту файлів у систему (рис. 3.15):

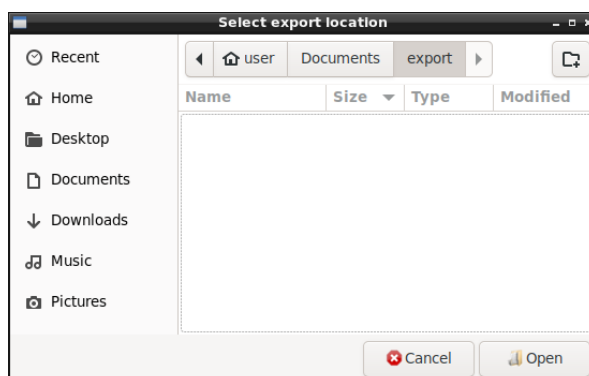


Рис. 3.15. Вікно системного імпорту та експорту файлів.

На наступному етапі проведемо тестування роботи програмного продукту.

3.3 Тестування роботи програмного продукту

Для створення нового проекту у програмі SAT Tool необхідно відкрити вікно списку всіх SAT проектів (відкривається за замовчуванням) та натиснути кнопку створення нового проекту. Також ми можемо: обрати проект, видалити його за допомогою кнопки delete, відкрити проект за допомогою подвійного кліку або виділення необхідного проекту та написання кнопки open (рис. 3.16):

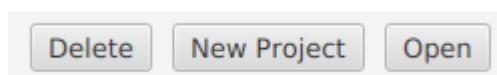


Рис. 3.16. Кнопки списку проектів

Окрім створення нового проекту ми також можемо виконати пошук по назві SAT проекту (рис. 3.17):

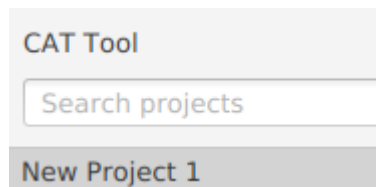


Рис. 3.17. Пошук проекту

У вікні створення проекту ми встановимо назву даного проекту та його опис (рис. 3.18):

Рис. 3.18. Встановлення назви та опису САТ проекту

А також оберемо початкову та вихідну мову проекту (рис. 3.19):

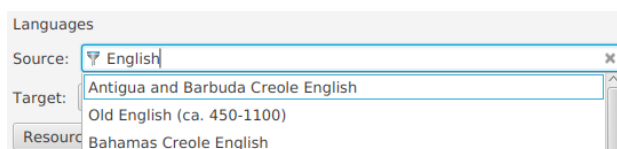


Рис. 3.19. Обрання мови проекту САТ

У нижній частині вікна проекту знаходяться кнопки, що відкривають ресурси, а також словник, перекладач проекту, кнопка експорту файлів проекту, кнопка повернення до списку проектів та кнопка збереження проекту САТ у базі даних додатку (рис. 3.20).

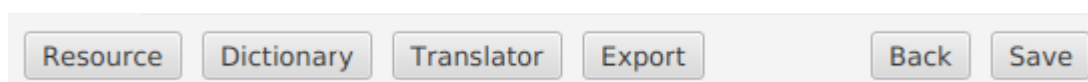


Рис. 3.20. Кнопки вікон та збереження САТ проекту

Натиснувши на кнопку resource, ми відкриємо список ресурсів до перекладу даного САТ проекту. У це вікно ми можемо додати або видалити ресурси за допомогою відповідних кнопок delete selected та add selected. Також файли у проект можливо додати просто виконавши переміщення їх у вікно списку ресурсів методом drag and drop (рис. 3.21).

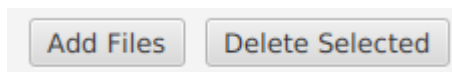


Рис. 3.21. Кнопки додавання та видалення файлів проекту

Для відкриття вікна редагування та перекладу тексту, двічі виконуємо клацання мишкою по назві необхідного ресурсу. Відкривається вікно редагування тексту. Також паралельно відкриємо перекладач тексту за допомогою кнопки Translator та пошук слів за словником за допомогою кнопки Dictionary (рис. 3.22, рис. 3.23).



Рис. 3.22. Кнопки Dictionary та Translator

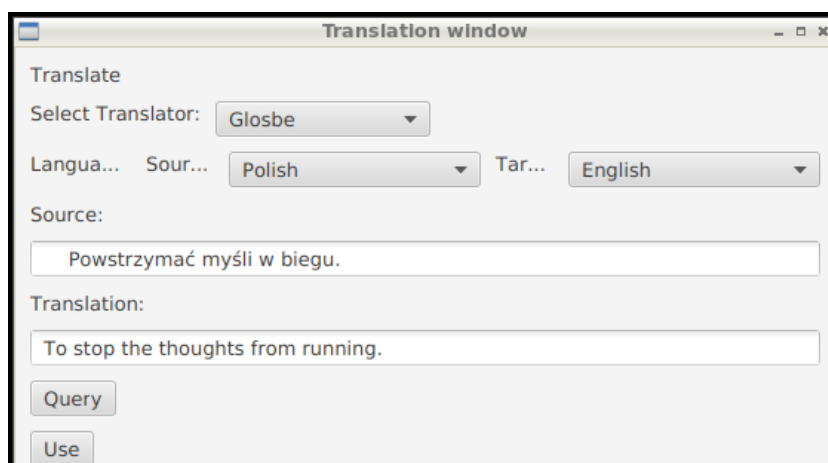


Рис. 3.23. Вікна редагування та перекладу тексту у CAT Tool

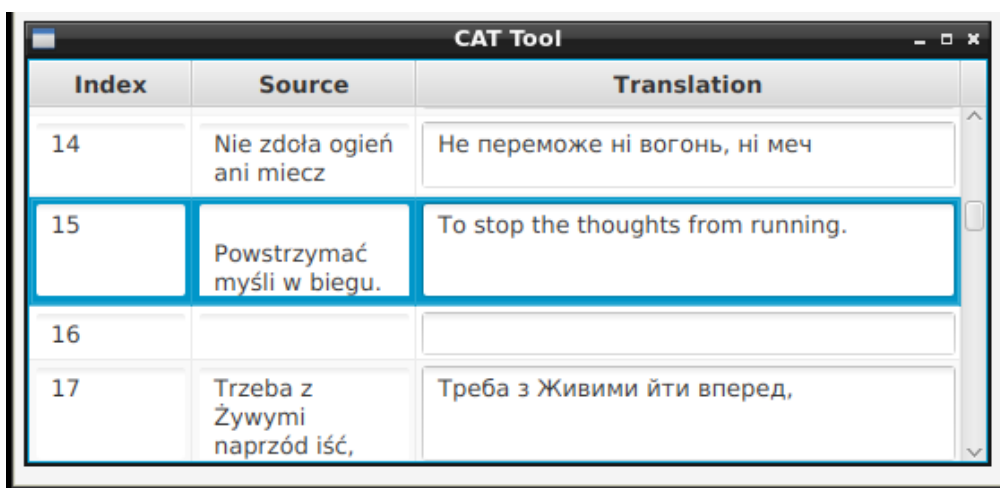


Рис. 3.24. Вікна редагування та перекладу тексту у CAT Tool (продовження зображення)

Для виконання перекладу обраної частини тексту необхідно двічі клікнути на рамку тексту оригіналу (source), щоб скопіювати вибраний текст до онлайн перекладача (рис. 3.24). Далі оберемо версію онлайн перекладача (у нашому випадку Glosbe), мову оригіналу (source) та мову перекладу (target). Для виконання перекладу натискаємо кнопку query. У результаті отримаємо перекладений текст у полі translation. Для копіювання тексту назад у вікно редагування, натиснемо кнопку use.

Для виконання пошуку значення слова за словником – виділимо та скопіюємо необхідне слово за допомогою комбінації клавіш Ctrl+C, (рис. 3.25)

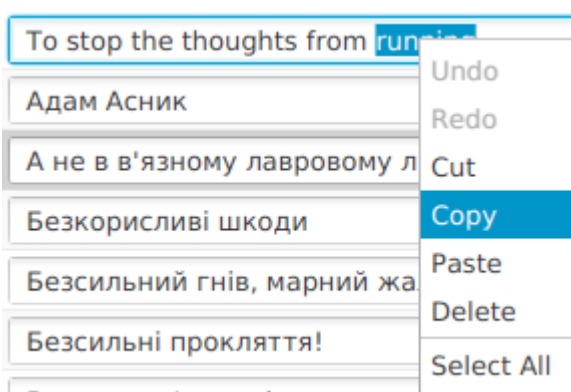


Рис. 3.25. Копіювання слова

Натиснемо кнопку clipboard у словнику, що вставить дане слово у форму пошуку словника, оберемо словник, за яким необхідно шукати дане слово та натиснемо search (рис. 3.26).



Рис. 3.26. Вікно онлайн словника додатку SAT Tool

Також ми можемо відкривати результати пошуку у зовнішньому браузері або вбудованому браузері у вікні додатку (рис. 3.27).



Рис. 3.27. Вбудований браузер

Для експорту файлів проекту натиснемо кнопку export у вікні проекту та оберемо місце для збереження файлів оригіналу та виконаного перекладу (рис. 3.28, рис. 3.29).

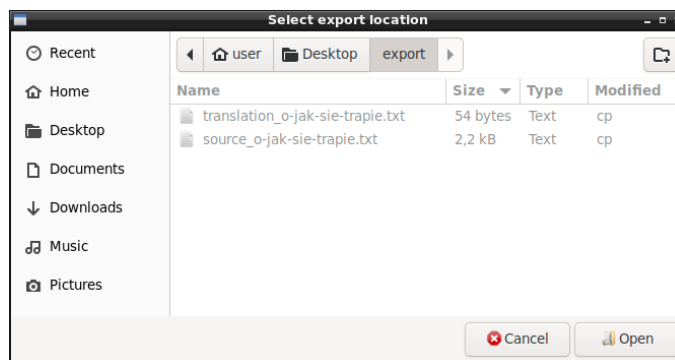


Рис. 3.28. Експорт ресурсів програми CAT Tool.

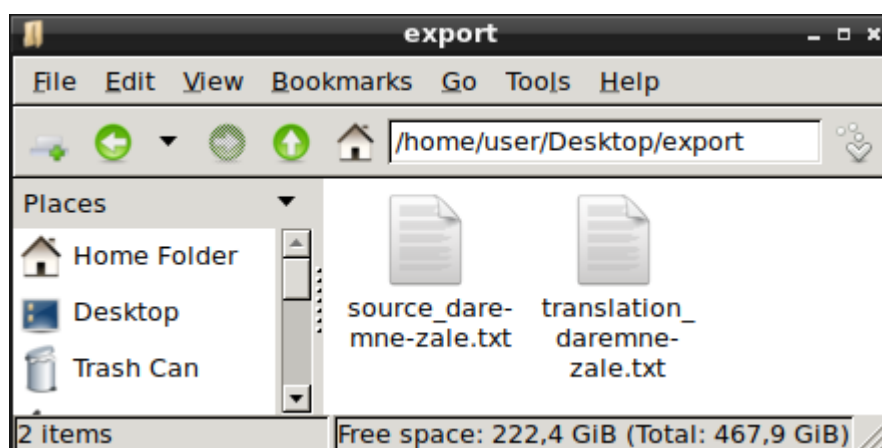


Рис. 3.29. Експортовані файли програми.

Файли експортуються у форматі TXT та складаються з оригіналу та перекладу тексту (рис. 3.30):

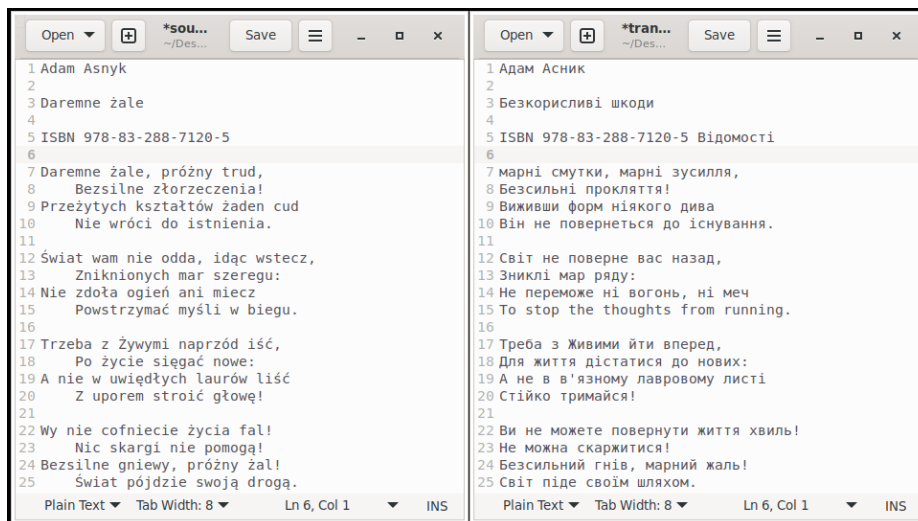


Рис. 3.30. Порівняння файлів оригіналу та перекладу

В результаті проведених дій підтверджено працездатність додатку.

Висновки до третього розділу

У третьому розділі нами було описано порядок встановлення, налаштування та використання додатку CAT Tool, що розроблений на мові програмування Java [23].

Для запуску додатку було виконано встановлення та налаштування Java Development Kit (JDK) 22 від компанії Oracle [24]. Нами було протестовано можливість запуску CAT інструменту через командний рядок або кліком на .jar файл. Перевірено правильну роботу бази даних SQLite [25].

Також ми ознайомились з інтерфейсом програми. Вона має такі вікна як: привітання користувача, управління проектами, додавання ресурсів, редагування тексту, використання онлайн перекладача та словника, а також імпорт та експорт файлів.

У кінці третього розділу нами було виконано тестування нашої програми: створено та відредаговано проект, використано онлайн інструменти перекладу, пошуку значення слова чи терміну та експорту файлів проекту, тобто на практиці перевірено зручність використання CAT інструменту.

ВИСНОВКИ

У цій дипломній роботі нами було проведено комплексне дослідження та розробку системи автоматизації перекладу (CAT Tool) з метою підвищення ефективності та якості перекладу текстів, а також спрощення та автоматизації процесів перекладу.

Основним результатом виконаної роботи є розроблений програмний продукт CAT Tool, що забезпечує автоматизацію процесів перекладу текстів, що в свою чергу підвищує ефективність та якість роботи перекладачів. Система була створена з використанням сучасних технологій і відповідає вимогам до програмних продуктів типу CAT.

За потребами користувачів нами було визначено функціональні та нефункціональні вимоги до системи.

Аналіз таких аналогічних продуктів, як Transifex, MateCat, BasicCat та OmegaT дозволив визначити ключові функції та обмеження, що було враховано під час розробки ПЗ.

Використання архітектурного патерну MVC (Model-View-Controller) забезпечило чітку структурування коду та спростило його майбутню підтримку.

Було обрано мову програмування Java, графічний інтерфейс JavaFX та БД SQLite для забезпечення кросплатформності та зручного зберігання даних.

Зберігання даних у БД та використання ORM фреймворку Hibernate для виконання CRUD операцій забезпечило надійне управління даними.

Ми реалізували базові алгоритми роботи системи, що забезпечують її функціонування відповідно вимогам користувачів.

Інтерфейс отримав такі вікна як: привітання, управління проектами, додавання ресурсів, редагування тексту, використання онлайн перекладача та словника, імпорт та експорт файлів.

Також було виконане тестування системи, що підтвердило її коректне функціонування та зручність використання користувачем.

					ПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

Розроблена система CAT Tool є практично цінною, оскільки дозволяє перекладачам та локалізаторам ефективніше виконувати свої завдання, автоматизовуючи рутинні процеси. Це знижує витрати часу на переклад та підвищує якість перекладів.

Для подальшого вдосконалення розробленої CAT системи можливо розглянути наступні напрямки дослідження та розвитку:

- Впровадження машинного навчання для покращення якості автоматичних перекладів та адаптації системи до стилю користувача.
- Розширення функціональності інтерфейсу для підтримки більш складних сценаріїв перекладу.
- Оптимізація роботи з великими обсягами даних для підвищення продуктивності системи.

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Transifex [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.transifex.com/>.
2. MateCat [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.matecat.com/>.
3. BasicCat [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.basiccatt.org/>.
4. OmegaT [Електронний ресурс] - Режим доступу до ресурсу:
<https://omegat.org.uk/>.
5. Hibernate documentation [Електронний ресурс] - Режим доступу до ресурсу:
<https://hibernate.org/orm/documentation/6.5/>.
6. Maven getting started [Електронний ресурс] - Режим доступу до ресурсу:
<https://maven.apache.org/guides/getting-started/>.
7. Lombok [Електронний ресурс] - Режим доступу до ресурсу:
<https://projectlombok.org/>.
8. Gregory G., Bauer C. Java Persistence with Hibernate, Second Edition. Manning Publications, 2015. [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.oreilly.com/library/view/java-persistence-with/9781617290459/>.
9. SQLite docs [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.sqlite.org/docs.html>.
10. Späth P., Sharan K. Learn JavaFX 17: Building User Experience and Interfaces with Java. Apress, 2022. [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.oreilly.com/library/view/learn-javafx-17/9781484278482/>.
11. Bloch J. Effective Java, 3rd Edition. dpunkt, 2018. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.oreilly.com/library/view/effective-java-3rd/9781492069669/>.
12. Jenkov JavaFX tutorials [Електронний ресурс] - Режим доступу до ресурсу:
<https://jenkov.com/tutorials/javafx/index.html>.
13. Jenkov Maven tutorials [Електронний ресурс] - Режим доступу до ресурсу:
<https://jenkov.com/tutorials/maven/maven-tutorial.html>.

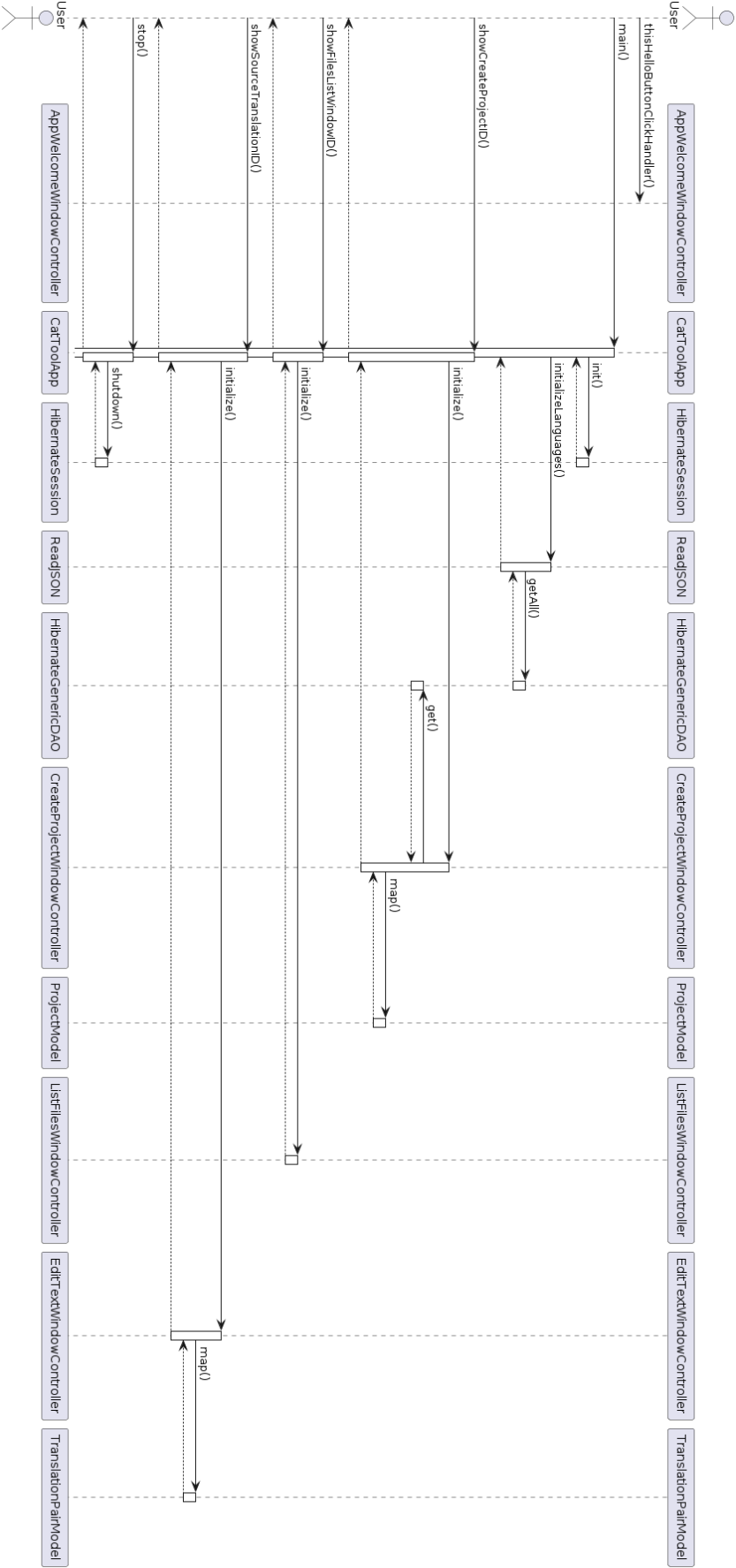
					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

14. JavaPoint Lombok instruction [Електронний ресурс] - Режим доступу до ресурсу: <https://www.javatpoint.com/lombok-java>.
15. TutorialsPoint SQLite introduction [Електронний ресурс] - Режим доступу до ресурсу: <https://www.tutorialspoint.com/sqlite/index.htm>
16. Darwin I. Java Cookbook, 4th Edition. O'Reilly Media, Inc., 2020. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.oreilly.com/library/view/java-cookbook-4th/9781492072577/>.
17. Sierra K., Bates B., Gee T. Head First Java, 3rd Edition. O'Reilly Media, Inc., 2022. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.oreilly.com/library/view/head-first-java/9781492091646/>.
18. Owens M., Allen G. The Definitive Guide to SQLite, Second Edition. Apress, 2010. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.oreilly.com/library/view/the-definitive-guide/9781430232254/>.
19. TutorialsPoint Jackson guide [Електронний ресурс] - Режим доступу до ресурсу: <https://www.tutorialspoint.com/jackson/index.htm>.
20. Baeldung Maven guide [Електронний ресурс] - Режим доступу до ресурсу: <https://www.baeldung.com/maven-guide>.
21. Verburg M., Evans B., Jason C. The Well-Grounded Java Developer, Second Edition. Manning Publications, 2022. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.oreilly.com/library/view/the-well-grounded-java/9781617298875/>.
22. Oracle Java 22 installation [Електронний ресурс] - Режим доступу до ресурсу: <https://www.oracle.com/cis/java/technologies/downloads/#java22>.
23. Oracle JavaSE tutorial [Електронний ресурс] - Режим доступу до ресурсу: <https://docs.oracle.com/javase/tutorial/>.
24. Oracle [Електронний ресурс] - Режим доступу до ресурсу: <https://www.oracle.com/>.
25. JavaPoint SQLite tutorial [Електронний ресурс] - Режим доступу до ресурсу: <https://www.javatpoint.com/sqlite-tutorial>.

ДОДАТКИ

					ІІЗ.КР.Б – 121 – 24 – ІЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Діаграма послідовності

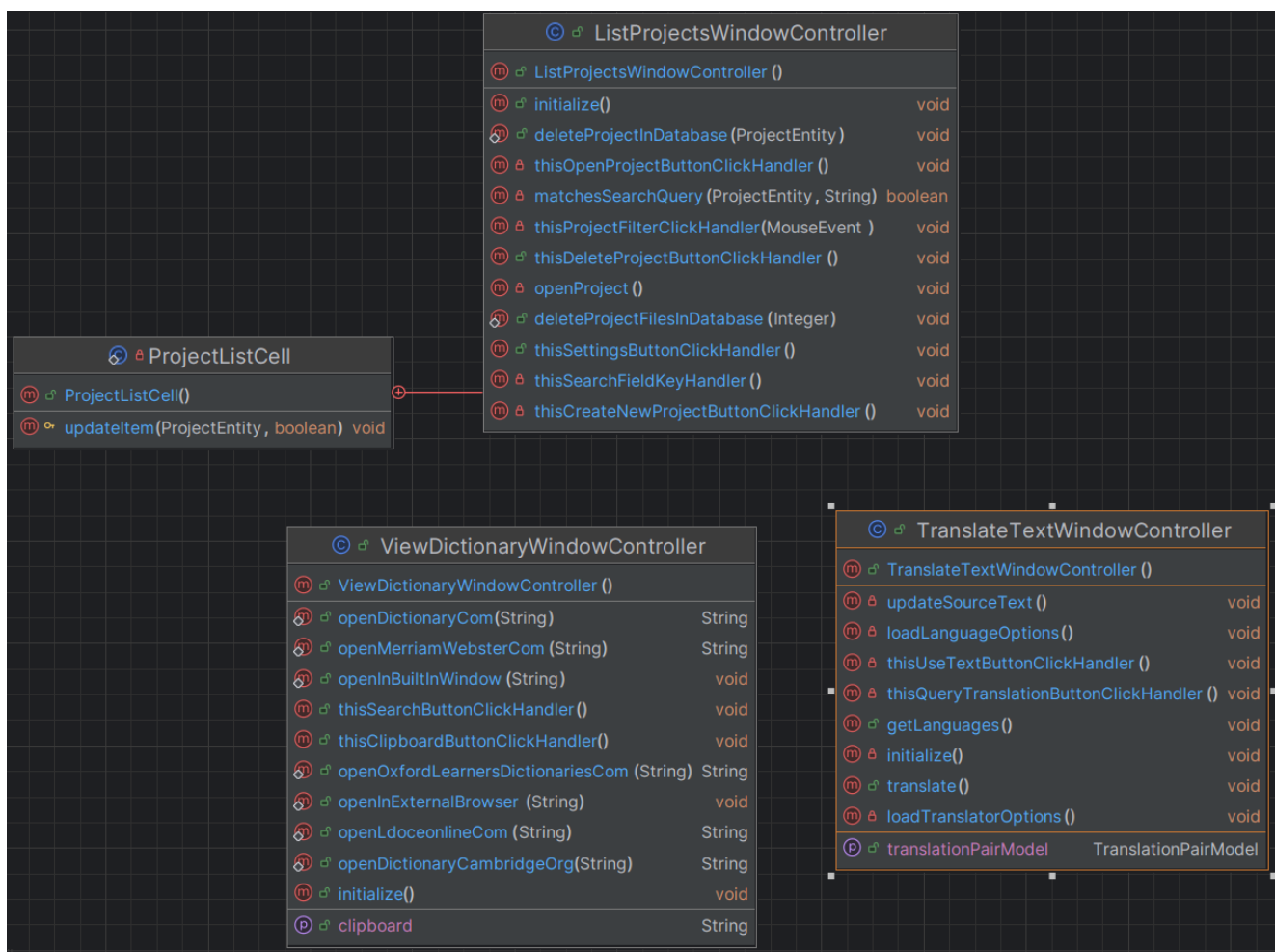
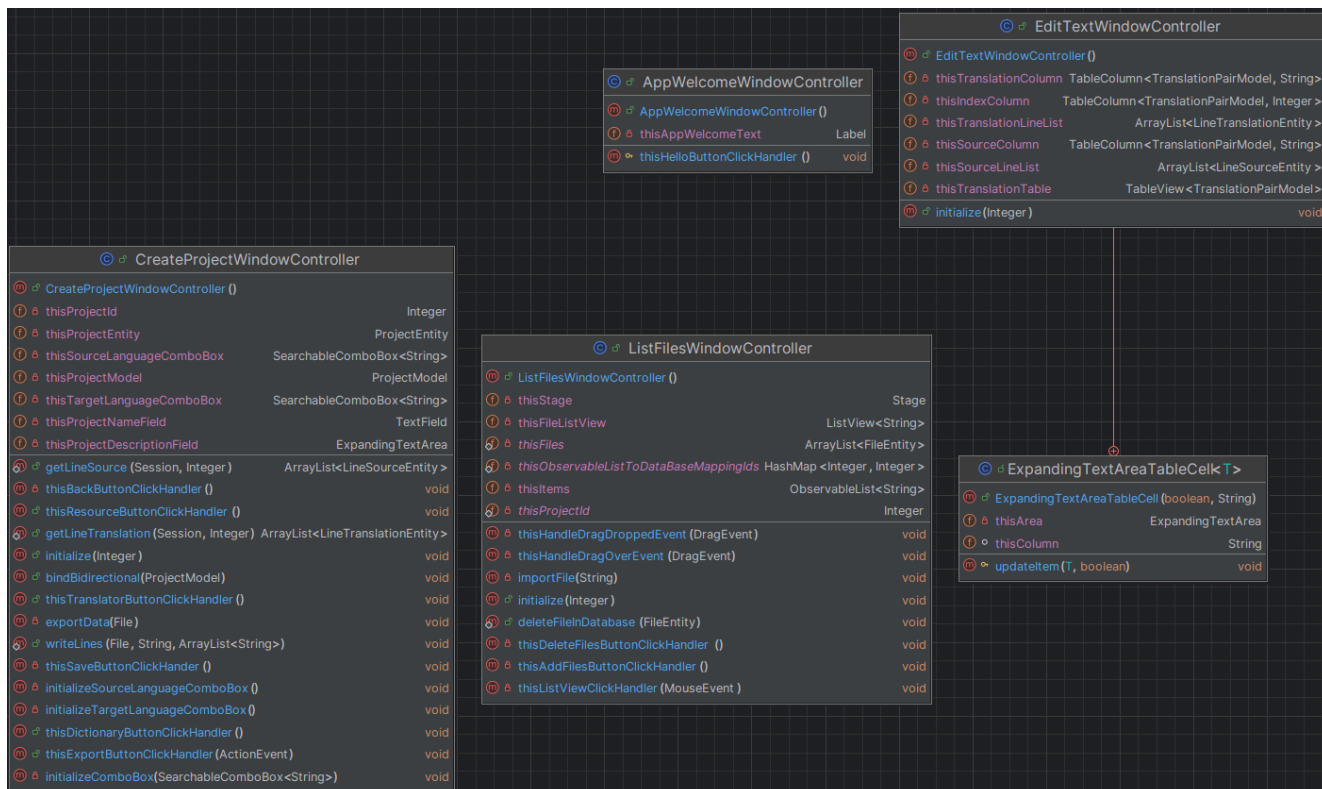


Діаграма Java класів

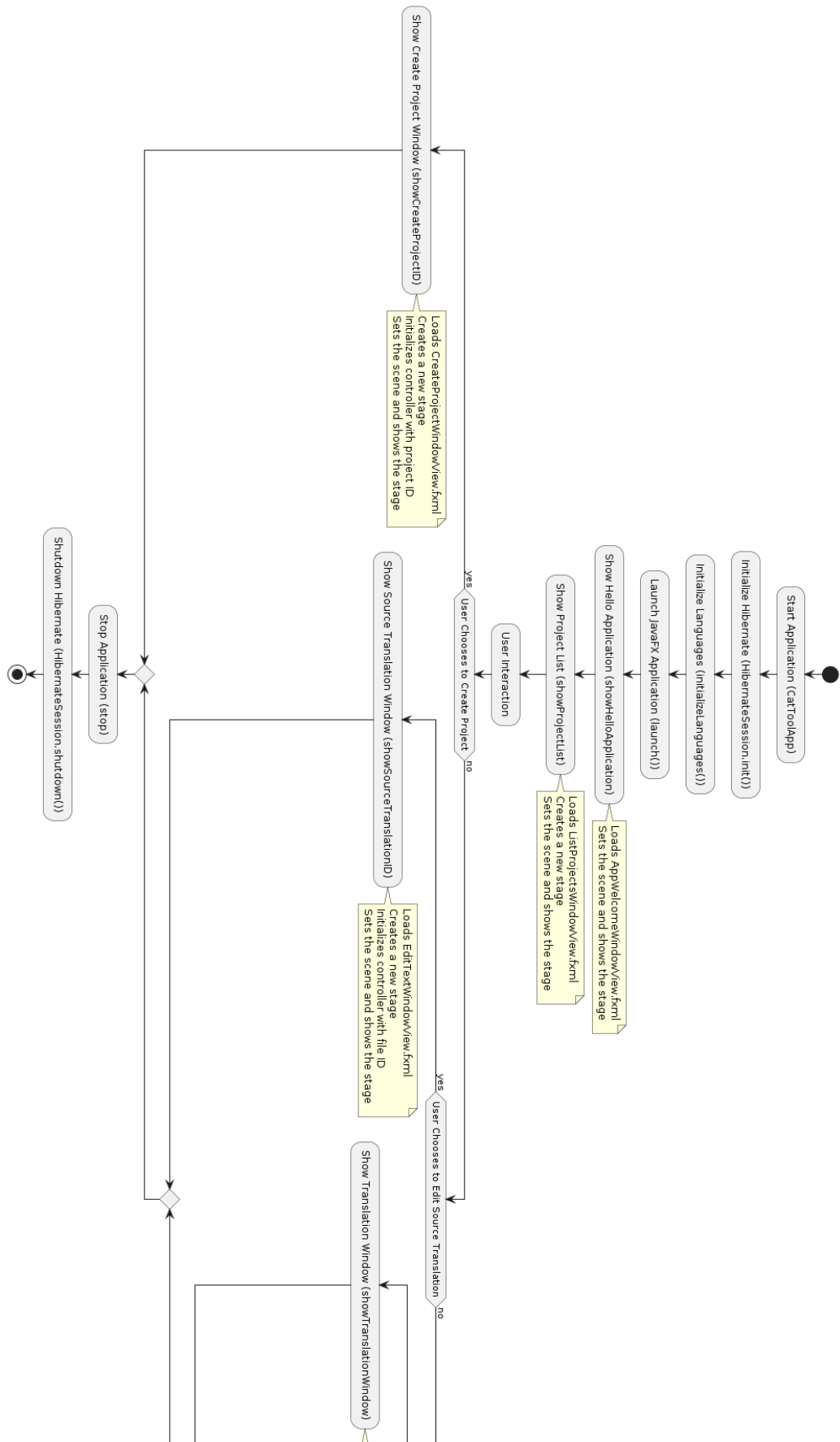
Ⓢ ↻ HibernateSession	Ⓢ ↻ HibernateGenericDAO	Ⓢ ↻ HibernateFilePersistence
Ⓢ ↻ HibernateSession ()	Ⓢ ↻ HibernateGenericDAO ()	Ⓢ ↻ HibernateFilePersistence ()
Ⓢ ↻ init() void	Ⓢ ↻ get(Integer) T	Ⓢ ↻ saveFileWithHibernate(String, Integer) void
Ⓢ ↻ shutdown() void	Ⓢ ↻ mergeOrPersist() T	Ⓢ ↻ getFileHash(String) String
Ⓢ ↻ session Session	Ⓢ ↻ delete() T	
Ⓢ ↻ sessionFactory SessionFactory	Ⓢ ↻ merge() T	
Ⓢ ↻ transaction Transaction	Ⓢ ↻ getAll(Class<T>) List<T>?	

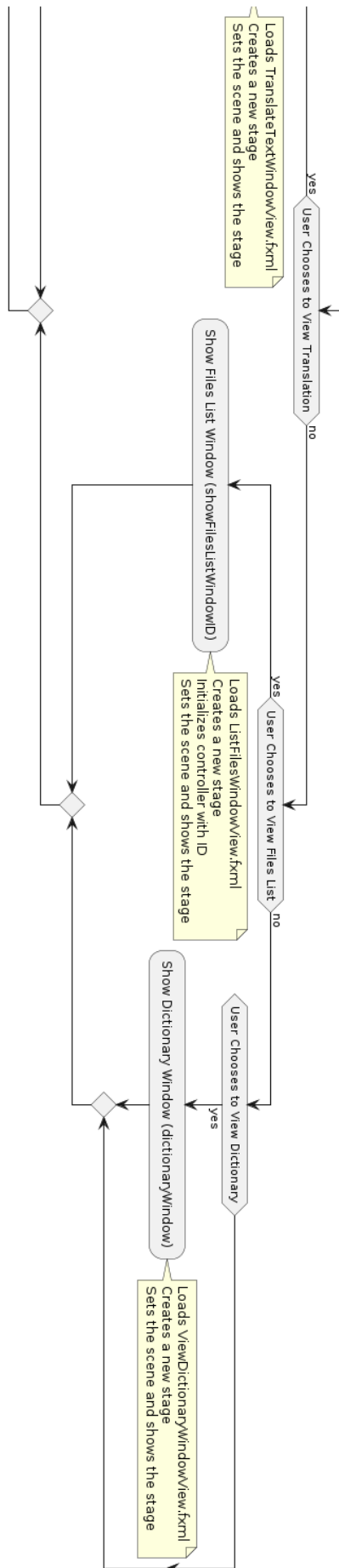
Ⓢ ↻ LanguageEntity	Ⓢ ↻ FileEntity	Ⓢ ↻ ProjectEntity	Ⓢ ↻ LineSourceEntity	Ⓢ ↻ LineTranslationEntity
Ⓢ ↻ LanguageEntity()	Ⓢ ↻ FileEntity()	Ⓢ ↻ ProjectEntity()	Ⓢ ↻ LineSourceEntity()	Ⓢ ↻ LineTranslationEntity()
Ⓢ ↻ id Integer	Ⓢ ↻ hash String	Ⓢ ↻ name String	Ⓢ ↻ fileId Integer	Ⓢ ↻ content String
Ⓢ ↻ name String	Ⓢ ↻ projectId Integer	Ⓢ ↻ targetLanguage Integer	Ⓢ ↻ id Integer	Ⓢ ↻ id Integer
Ⓢ ↻ code String	Ⓢ ↻ content byte[]	Ⓢ ↻ sourceLanguage Integer	Ⓢ ↻ index Integer	Ⓢ ↻ targetLanguageId Integer
Ⓢ ↻ equals(Object) boolean	Ⓢ ↻ lastModified Long	Ⓢ ↻ description String	Ⓢ ↻ sourceLanguageId Integer	Ⓢ ↻ index Integer
Ⓢ ↻ canEqual(Object) boolean	Ⓢ ↻ size Long	Ⓢ ↻ id Integer	Ⓢ ↻ content String	Ⓢ ↻ fileId Integer
Ⓢ ↻ hashCode() int	Ⓢ ↻ id Integer	Ⓢ ↻ getId() Integer	Ⓢ ↻ equals(Object) boolean	Ⓢ ↻ getId() Integer
Ⓢ ↻ getId() Integer	Ⓢ ↻ name String	Ⓢ ↻ equals(Object) boolean	Ⓢ ↻ canEqual(Object) boolean	Ⓢ ↻ equals(Object) boolean
Ⓢ ↻ getCode() String	Ⓢ ↻ getLastModified() Long	Ⓢ ↻ canEqual(Object) boolean	Ⓢ ↻ hashCode() int	Ⓢ ↻ canEqual(Object) boolean
Ⓢ ↻ getName() String	Ⓢ ↻ equals(Object) boolean	Ⓢ ↻ hashCode() int	Ⓢ ↻ getId() Integer	Ⓢ ↻ hashCode() int
Ⓢ ↻ setId(Integer) void	Ⓢ ↻ setSize(Long) void	Ⓢ ↻ setDescription(String) void	Ⓢ ↻ getFileId() Integer	Ⓢ ↻ getFileId() Integer
Ⓢ ↻ setName(String) void	Ⓢ ↻ setId(Integer) void	Ⓢ ↻ setName(String) void	Ⓢ ↻ setContent(String) void	Ⓢ ↻ getIndex() Integer
Ⓢ ↻ toString() String	Ⓢ ↻ setLastModified(Long) void	Ⓢ ↻ getSourceLanguage() Integer	Ⓢ ↻ getIndex() Integer	Ⓢ ↻ getTargetLanguageId() Integer
	Ⓢ ↻ toString() String	Ⓢ ↻ getTargetLanguage() Integer	Ⓢ ↻ setId(Integer) void	Ⓢ ↻ getContent() String
	Ⓢ ↻ hashCode() int	Ⓢ ↻ setTargetLanguage(Integer) void	Ⓢ ↻ getSourceLanguageId() Integer	Ⓢ ↻ setId(Integer) void
	Ⓢ ↻ getId() Integer	Ⓢ ↻ getDescription() String	Ⓢ ↻ setContent() String	Ⓢ ↻ toString() String
	Ⓢ ↻ getProjectId() Integer	Ⓢ ↻ setName(String) void	Ⓢ ↻ getId(Integer) void	Ⓢ ↻ setIndex(Integer) void
	Ⓢ ↻ setContent(byte[]) void	Ⓢ ↻ setId(Integer) void	Ⓢ ↻ setSourceLanguageId(Integer) void	Ⓢ ↻ setTargetLanguageId(Integer) void
	Ⓢ ↻ getName() String	Ⓢ ↻ toString() String	Ⓢ ↻ setFileId(Integer) void	Ⓢ ↻ setContent(String) void
	Ⓢ ↻ getSize() Long	Ⓢ ↻ setSourceLanguage(Integer) void	Ⓢ ↻ toString() String	Ⓢ ↻ setFileId(Integer) void
	Ⓢ ↻ setName(String) void			
	Ⓢ ↻ getHash() String			
	Ⓢ ↻ setHash(String) void			
	Ⓢ ↻ getContent() byte[]			

Ⓢ ↻ CatToolApp	Ⓢ ↻ Starter
Ⓢ ↻ CatToolApp()	Ⓢ ↻ Starter()
Ⓢ ↻ dictionaryWindow() void	Ⓢ ↻ main(String[]) void
Ⓢ ↻ showSourceTranslationID(Integer) void	
Ⓢ ↻ showTranslationWindow() void	
Ⓢ ↻ showHelloApplication(Stage) void	
Ⓢ ↻ showCreateProjectID(Integer) void	
Ⓢ ↻ main(String[]) void	
Ⓢ ↻ showFilesListWindowID(Integer) void	
Ⓢ ↻ showProjectList() void	
Ⓢ ↻ start(Stage) void	
Ⓢ ↻ stop() void	



Діаграма Активності





Код програми:

```
// File: cat-tool/pom.xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>cat-tool</artifactId>
  <version>1.0-SNAPSHOT</version>
  <name>cat-tool</name>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <junit.version>5.10.0</junit.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-text</artifactId>
      <version>1.12.0</version>
    </dependency>
    <dependency>
      <groupId>org.controlsfx</groupId>
      <artifactId>controlsfx</artifactId>
      <version>11.2.0</version>
    </dependency>
    <dependency>
      <groupId>com.dlsc.gemsfx</groupId>
      <artifactId>gemsfx</artifactId>
      <version>2.7.2</version>
    </dependency>
    <dependency>
      <groupId>org.hibernate.orm</groupId>
      <artifactId>hibernate-community-dialects</artifactId>
      <version>6.4.4.Final</version>
    </dependency>
    <dependency>
      <groupId>org.hibernate.orm</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>6.4.4.Final</version>
    </dependency>
    <dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-core</artifactId>
      <version>2.17.0</version>
    </dependency>
    <dependency>
      <groupId>com.fasterxml.jackson.core</groupId>
      <artifactId>jackson-databind</artifactId>
      <version>2.17.0</version>
    </dependency>
    <dependency>
      <groupId>org.openjfx</groupId>
      <artifactId>javafx-controls</artifactId>
      <version>21</version>
    </dependency>
    <dependency>
      <groupId>org.openjfx</groupId>
      <artifactId>javafx-fxml</artifactId>
```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    <version>21</version>
  </dependency>
  <dependency>
    <groupId>org.openjfx</groupId>
    <artifactId>javafx-web</artifactId>
    <version>22</version>
  </dependency>
  <dependency>
    <groupId>javax.persistence</groupId>
    <artifactId>javax.persistence-api</artifactId>
    <version>2.2</version>
  </dependency>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-api</artifactId>
    <version>5.10.0</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-engine</artifactId>
    <version>5.10.0</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>2.17.1</version>
  </dependency>
  <dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-slf4j-impl</artifactId>
    <version>2.17.1</version>
  </dependency>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.32</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>com.gluonhq</groupId>
    <artifactId>rich-text-area</artifactId>
    <version>1.1.1</version>
  </dependency>
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.7.32</version>
  </dependency>
  <dependency>
    <groupId>org.xerial</groupId>
    <artifactId>sqlite-jdbc</artifactId>
    <version>3.45.2.0</version>
  </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.openjfx</groupId>
      <artifactId>javafx-maven-plugin</artifactId>
      <version>0.0.8</version>
    </plugin>
  </plugins>
</build>

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    <executions>
      <execution>
        <id>default-cli</id>
        <configuration>
          <mainClass>org.example.cattool/org.example.cattool.gui.javaafx.application.CatToolApp</mainClass>
          <launcher>app</launcher>
          <jlinkZipName>app</jlinkZipName>
          <jlinkImageName>app</jlinkImageName>
          <noManPages>true</noManPages>
          <stripDebug>true</stripDebug>
          <noHeaderFiles>true</noHeaderFiles>
        </configuration>
      </execution>
    </executions>
  </plugin>
</plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.13.0</version>
<configuration>
  <source>22</source>
  <target>22</target>
</configuration>
</plugin>
</plugins>
</build>
</project>
// File: cat-tool/src/main/java/module-info.java
module org.example {
  exports org.example.cattool.database.hibernate.entity;
  exports org.example.cattool.gui.javaafx.application;
  exports org.example.cattool.gui.javaafx.controller;
  exports org.example.cattool.gui.javaafx.model;
  opens  org.example.cattool.gui.javaafx.model to javaafx.fxml;
  opens org.example.cattool.database.hibernate to org.hibernate.orm.core;
  opens org.example.cattool.database.hibernate.entity to javaafx.fxml, org.hibernate.orm.core;
  opens org.example.cattool.gui.javaafx.application to javaafx.fxml;
  opens org.example.cattool.gui.javaafx.controller to javaafx.fxml;
  opens org.example.cattool.translate.reverso.deserialization.json.languages to com.fasterxml.jackson.databind;
  opens org.example.cattool.translate.reverso.deserialization.json.response to com.fasterxml.jackson.databind;
  requires com.dlsc.gemsfx;
  requires com.fasterxml.jackson.databind;
  requires jakarta.persistence;
  requires java.desktop;
  requires java.naming;
  requires java.persistence;
  requires java.xml;
  requires javaafx.controls;
  requires javaafx.fxml;
  requires javaafx.web;
  requires org.apache.commons.text;
  requires org.controlsfx.controls;
  requires org.hibernate.orm.core;
  requires static lombok;
  exports org.example.cattool.translate.glosbe.deserialization.json.languages to com.fasterxml.jackson.databind;
  exports org.example.cattool.translate.glosbe.deserialization.json.response to com.fasterxml.jackson.databind;
  exports org.example.cattool.translate.mymemory to com.fasterxml.jackson.databind;
  exports org.example.cattool.translate.mymemory.deserialization.json.languages to com.fasterxml.jackson.databind;
  exports org.example.cattool.translate.mymemory.deserialization.json.response to com.fasterxml.jackson.databind;
  exports org.example.cattool.translate.pons to com.fasterxml.jackson.databind;
  exports org.example.cattool.translate.pons.deserialization.json.languages to com.fasterxml.jackson.databind;
  exports org.example.cattool.translate.pons.deserialization.json.response to com.fasterxml.jackson.databind;
}

```

```

}
// File: cat-tool/src/main/java/org/example/cattool/database/hibernate/entity/FileEntity.java
package org.example.cattool.database.hibernate.entity;
import jakarta.persistence.*;
import lombok.*;
import org.example.cattool.database.hibernate.HibernateGenericDAO;
@Data
@Entity
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@Table(name = "File")
@ToString
public class FileEntity extends HibernateGenericDAO {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    @Column(name = "project_id")
    private Integer projectId;
    @Column(name = "name")
    private String name;
    @Column(name = "size")
    private Long size;
    @Column(name = "last_modified")
    private Long lastModified;
    @Column(name = "content")
    private byte[] content;
    @Column(name = "hash")
    private String hash;
}

// File: cat-tool/src/main/java/org/example/cattool/database/hibernate/entity/LanguageEntity.java
package org.example.cattool.database.hibernate.entity;
import com.fasterxml.jackson.annotation.JsonProperty;
import jakarta.persistence.*;
import lombok.*;
import org.example.cattool.database.hibernate.HibernateGenericDAO;
@Data
@Entity
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@Table(name = "Language")
@ToString
public class LanguageEntity extends HibernateGenericDAO {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    @Column(name = "code")
    @JsonProperty("code")
    private String code;
    @Column(name = "name")
    @JsonProperty("name")
    private String name;
}

// File: cat-tool/src/main/java/org/example/cattool/database/hibernate/entity/LineSourceEntity.java
package org.example.cattool.database.hibernate.entity;
import jakarta.persistence.*;
import lombok.*;
import org.example.cattool.database.hibernate.HibernateGenericDAO;
@Data

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

```

@Entity
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@Table(name = "LineSource")
@ToString
public class LineSourceEntity extends HibernateGenericDAO {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    @Column(name = "file_id")
    private Integer fileId;
    @Column(name = "idx")
    private Integer index;
    @Column(name = "source_language_id")
    private Integer sourceLanguageId;
    @Column(name = "content")
    private String content;
}

// File: cat-tool/src/main/java/org/example/cattool/database/hibernate/entity/LineTranslationEntity.java
package org.example.cattool.database.hibernate.entity;
import jakarta.persistence.*;
import lombok.*;
import org.example.cattool.database.hibernate.HibernateGenericDAO;
@Data
@Entity
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@Table(name = "LineTranslation")
@ToString
public class LineTranslationEntity extends HibernateGenericDAO {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    @Column(name = "file_id")
    private Integer fileId;
    @Column(name = "idx")
    private Integer index;
    @Column(name = "target_language_id")
    private Integer targetLanguageId;
    @Column(name = "content")
    private String content;
}

// File: cat-tool/src/main/java/org/example/cattool/database/hibernate/entity/ProjectEntity.java
package org.example.cattool.database.hibernate.entity;
import jakarta.persistence.*;
import lombok.*;
import org.example.cattool.database.hibernate.HibernateGenericDAO;
@Data
@Entity
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@Table(name = "Project")
@ToString
public class ProjectEntity extends HibernateGenericDAO {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

```

private Integer id;
private Integer sourceLanguage;
private Integer targetLanguage;
private String description;
private String name;
}
// File: cat-tool/src/main/java/org/example/cattool/database/hibernate/HibernateFilePersistence.java
package org.example.cattool.database.hibernate;
import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import org.example.cattool.database.hibernate.entity.FileEntity;
public class HibernateFilePersistence {
    public static void saveFileWithHibernate(String thisFilePath, Integer thisProjectId) {
        File thisFile = new File(thisFilePath);
        String thisFileName = thisFile.getName();
        long thisFileSize = thisFile.length();
        long thisLastModified = thisFile.lastModified();
        try {
            String thisFileHash = getFileHash(thisFilePath);
            byte[] thisByteContent = Files.readAllBytes(Paths.get(thisFilePath));
            FileEntity thisFileEntity = new FileEntity();
            thisFileEntity = thisFileEntity.mergeOrPersist();
            thisFileEntity.setContent(thisByteContent);
            thisFileEntity.setHash(thisFileHash);
            thisFileEntity.setLastModified(thisLastModified);
            thisFileEntity.setName(thisFileName);
            thisFileEntity.setProjectId(thisProjectId);
            thisFileEntity.setSize(thisFileSize);
            thisFileEntity = thisFileEntity.mergeOrPersist();
        } catch (IOException | NoSuchAlgorithmException thisException) {
            throw new RuntimeException(thisException);
        }
    }
    public static String getFileHash(String thisFilePath) throws IOException, NoSuchAlgorithmException {
        MessageDigest thisDigest = MessageDigest.getInstance("SHA3-256");
        byte[] thisHashBytes = thisDigest.digest(Files.readAllBytes(Paths.get(thisFilePath)));
        StringBuilder thisHashBuilder = new StringBuilder();
        for(byte currentByte : thisHashBytes) {
            thisHashBuilder.append(String.format("%02x", currentByte));
        }
        return thisHashBuilder.toString();
    }
}
// File: cat-tool/src/main/java/org/example/cattool/database/hibernate/HibernateGenericDAO.java
package org.example.cattool.database.hibernate;
import jakarta.persistence.criteria.CriteriaQuery;
import jakarta.persistence.criteria.Root;
import java.util.List;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.criteria.HibernateCriteriaBuilder;
import static org.example.cattool.database.hibernate.HibernateSession.getSession;
import static org.example.cattool.database.hibernate.HibernateSession.getTransaction;
public class HibernateGenericDAO {
    public <T> T mergeOrPersist() {
        Session thisSession = getSession();
        Transaction thisTransaction = getTransaction();
        thisTransaction.begin();
    }
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

```

try {
    if(thisSession.contains(this)) {
        T thisT = (T) thisSession.merge(this);
        thisTransaction.commit();
        return thisT;
    } else {
        thisSession.persist(this);
        thisTransaction.commit();
        return (T) this;
    }
} catch (Exception thisException) {
    if(thisTransaction != null) {
        thisTransaction.rollback();
    }
    thisException.printStackTrace();
}
return null;
}

public <T> T merge() {
    Session thisSession = getSession();
    Transaction thisTransaction = getTransaction();
    thisTransaction.begin();
    try {
        T thisT = (T) thisSession.merge(this);
        thisTransaction.commit();
        return thisT;
    } catch (Exception thisException) {
        if(thisTransaction != null) {
            thisTransaction.rollback();
        }
        thisException.printStackTrace();
    }
    return null;
}

public <T> T get(Integer id) {
    Session thisSession = getSession();
    try {
        if(id != null) {
            return (T) thisSession.get(this.getClass(), id);
        }
    } catch (Exception thisException) {
        thisException.printStackTrace();
    }
    return null;
}

public <T> T delete() {
    Session thisSession = getSession();
    Transaction thisTransaction = getTransaction();
    thisTransaction.begin();
    try {
        if(thisSession.contains(this)) {
            thisSession.remove(this);
        }
        thisTransaction.commit();
        return (T) this;
    } catch (Exception thisException) {
        if(thisTransaction != null) {
            thisTransaction.rollback();
        }
        thisException.printStackTrace();
    }
    return null;
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    public static <T> List<T> getAll(Class<T> thisClass) {
        Session thisSession = getSession();
        try {
            HibernateCriteriaBuilder thisCriteriaBuilder = thisSession.getCriteriaBuilder();
            CriteriaQuery<T> thisQuery = thisCriteriaBuilder.createQuery(thisClass);
            Root<T> thisRoot = thisQuery.from(thisClass);
            thisQuery.select(thisRoot);
            return thisSession.createQuery(thisQuery).getResultList();
        } catch (Exception thisException) {
            thisException.printStackTrace();
        }
        return null;
    }
}
// File: cat-tool/src/main/java/org/example/cattool/database/hibernate/HibernateSession.java
package org.example.cattool.database.hibernate;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
public class HibernateSession {
    private static Session thisSession;
    private static SessionFactory thisSessionFactory;
    private static SessionFactory getSessionFactory() {
        try {
            if(thisSessionFactory == null || thisSessionFactory.isClosed()) {
                thisSessionFactory = new Configuration().configure("hibernate.cfg.xml").buildSessionFactory();
            }
            return thisSessionFactory;
        } catch (Exception thisException) {
            throw new ExceptionInInitializerError(thisException);
        }
    }
    public static Session getSession() {
        if(thisSession == null || !thisSession.isOpen()) {
            thisSession = getSessionFactory().openSession();
        }
        return thisSession;
    }
    public static void init() {
        getSessionFactory();
    }
    public static Transaction getTransaction() {
        return getSession().getTransaction();
    }
    public static void shutdown() {
        getSessionFactory().close();
        getSession().close();
    }
}
// File: cat-tool/src/main/java/org/example/cattool/gui/javafx/application/CatToolApp.java
package org.example.cattool.gui.javafx.application;
import java.io.IOException;
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import org.example.cattool.database.hibernate.HibernateSession;
import org.example.cattool.gui.javafx.controller.*;

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import static org.example.cattool.utils.json.ReadJSON.initializeLanguages;
public class CatToolApp extends Application {
    public static CatToolApp thisApplication;
    public static Stage thisCreateProjectStage;
    public static Stage thisDictionaryStage;
    public static Stage thisFileListStage;
    public static Stage thisListProjectsStage;
    public static Stage thisSourceTranslationStage;
    public static Stage thisTranslationWindowStage;
    @Override
    public void start(Stage stage) throws IOException {
        showHelloApplication(stage);
        showProjectList();
    }
    public static void main(String[] args) {
        HibernateSession.init();
        initializeLanguages();
        launch();
    }
    public void showHelloApplication(Stage thisStage) throws IOException {
        thisApplication = this;
        FXMLLoader thisLoader = new FXMLLoader(CatToolApp.class.getResource(
            "/org/example/cattool/gui/javafx/view/AppWelcomeWindowView.fxml"));
        Scene thisScene = new Scene(thisLoader.load(), 320, 240);
        thisStage.setTitle("CAT Tool");
        thisStage.setScene(thisScene);
        thisStage.show();
    }
    public void showProjectList() throws IOException {
        thisListProjectsStage = new Stage();
        FXMLLoader thisLoader = new FXMLLoader(getClass().getResource(
            "/org/example/cattool/gui/javafx/view/ListProjectsWindowView.fxml"));
        Scene thisScene = new Scene(thisLoader.load());
        thisListProjectsStage.setTitle("Welcome to CAT Tool");
        thisListProjectsStage.setScene(thisScene);
        thisListProjectsStage.show();
        thisListProjectsStage.setMinHeight(720);
        thisListProjectsStage.setWidth(1280);
    }
    public void showCreateProjectID(Integer thisProjectId) throws IOException {
        FXMLLoader thisLoader = new FXMLLoader(getClass().getResource(
            "/org/example/cattool/gui/javafx/view/CreateProjectWindowView.fxml"));
        AnchorPane thisRoot = thisLoader.load();
        CreateProjectWindowController thisController = thisLoader.getController();
        thisController.initialize(thisProjectId);
        Stage thisStage = new Stage();
        thisStage.setScene(new Scene(thisRoot));
        thisStage.setTitle("Project " + thisProjectId);
        thisStage.show();
        thisStage.setMinHeight(500);
        thisStage.setMinWidth(500);
    }
    public void showSourceTranslationID(Integer thisFileId) throws IOException {
        FXMLLoader thisLoader = new FXMLLoader(getClass().getResource(
            "/org/example/cattool/gui/javafx/view/EditTextWindowView.fxml"));
        Parent thisRoot = thisLoader.load();
        EditTextWindowController thisController = thisLoader.getController();
        thisController.initialize(thisFileId);
        thisSourceTranslationStage = new Stage();
        thisSourceTranslationStage.setTitle("CAT Tool");
        thisSourceTranslationStage.setScene(new Scene(thisRoot));
        thisSourceTranslationStage.show();
    }
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    public void showTranslationWindow() throws IOException {
        FXMLLoader thisLoader = new FXMLLoader(getClass().getResource(
            "/org/example/cattool/gui/javafx/view/TranslateTextWindowView.fxml"));
        Parent thisRoot = thisLoader.load();
        TranslateTextWindowController thisController = thisLoader.getController();
        thisTranslationWindowStage = new Stage();
        thisTranslationWindowStage.setTitle("Translation window");
        thisTranslationWindowStage.setScene(new Scene(thisRoot));
        thisTranslationWindowStage.show();
    }
    public void showFilesListWindowID(Integer thisId) throws IOException {
        FXMLLoader thisLoader = new FXMLLoader(getClass().getResource(
            "/org/example/cattool/gui/javafx/view/ListFilesWindowView.fxml"));
        Parent thisRoot = thisLoader.load();
        ListFilesWindowController thisController = thisLoader.getController();
        thisController.initialize(thisId);
        thisFileListStage = new Stage();
        thisFileListStage.setTitle("Resources");
        thisFileListStage.setScene(new Scene(thisRoot));
        thisFileListStage.setMinHeight(720);
        thisFileListStage.setWidth(1280);
        thisFileListStage.show();
    }
    public void dictionaryWindow() throws IOException {
        FXMLLoader thisLoader = new FXMLLoader(getClass().getResource(
            "/org/example/cattool/gui/javafx/view/ViewDictionaryWindowView.fxml"));
        Parent thisRoot = thisLoader.load();
        ViewDictionaryWindowController thisController = thisLoader.getController();
        thisDictionaryStage = new Stage();
        thisDictionaryStage.setTitle("Dictionary");
        thisDictionaryStage.setScene(new Scene(thisRoot));
        thisDictionaryStage.show();
    }
    @Override
    public void stop() {
        HibernateSession.shutdown();
    }
}
// File: cat-tool/src/main/java/org/example/cattool/gui/javafx/controller/AppWelcomeWindowController.java
package org.example.cattool.gui.javafx.controller;
import javafx.fxml.FXML;
import javafx.scene.control.Label;
public class AppWelcomeWindowController {
    @FXML
    private Label thisAppWelcomeText;
    @FXML
    protected void thisHelloButtonClickHandler() {
        thisAppWelcomeText.setText("Welcome to CAT Tool!");
    }
}
// File: cat-tool/src/main/java/org/example/cattool/gui/javafx/controller/CreateProjectWindowController.java
package org.example.cattool.gui.javafx.controller;
import com.dlsc.gemsfx.ExpandingTextArea;
import jakarta.persistence.criteria.CriteriaBuilder;
import jakarta.persistence.criteria.CriteriaQuery;
import jakarta.persistence.criteria.Root;
import java.io.*;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.ArrayList;
import javafx.collections.FXCollections;

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.*;
import javafx.scene.control.cell.ComboBoxListCell;
import javafx.scene.control.skin.ComboBoxListViewSkin;
import javafx.scene.control.skin.VirtualFlow;
import javafx.stage.DirectoryChooser;
import javafx.stage.Stage;
import javafx.util.Callback;
import org.controlsfx.control.SearchableComboBox;
import org.example.cattool.database.hibernate.HibernateGenericDAO;
import org.example.cattool.database.hibernate.entity.*;
import org.example.cattool.gui.javaafx.application.CatToolApp;
import org.example.cattool.gui.javaafx.model.ProjectModel;
import org.hibernate.Session;
import static org.example.cattool.database.hibernate.HibernateSession.getSession;

public class CreateProjectWindowController {
    private Integer thisProjectId;
    private ProjectEntity thisProjectEntity;
    private ProjectModel thisProjectModel;
    @FXML
    private TextField thisProjectNameField;
    @FXML
    private ExpandingTextArea thisProjectDescriptionField;
    @FXML
    private SearchableComboBox<String> thisSourceLanguageComboBox;
    @FXML
    private SearchableComboBox<String> thisTargetLanguageComboBox;
    @FXML

    public void initialize(Integer thisProjectId) {
        if(thisProjectId != null) {
            this.thisProjectId = thisProjectId;
        }
        initializeSourceLanguageComboBox();
        initializeTargetLanguageComboBox();
        if(thisProjectEntity == null) {
            thisProjectEntity = new ProjectEntity();
        }
        thisProjectEntity = thisProjectEntity.get(thisProjectId);
        thisProjectModel = new ProjectModel();
        if(thisProjectEntity != null) {
            thisProjectModel.map(thisProjectEntity);
        }
        bindBidirectional(thisProjectModel);
        if(thisProjectEntity != null) {
            thisProjectNameField.setText(thisProjectEntity.getName());
            thisProjectDescriptionField.setText(thisProjectEntity.getDescription());
            thisSourceLanguageComboBox.getSelectionModel().select(thisProjectEntity.getSourceLanguage() - 1);
            thisTargetLanguageComboBox.getSelectionModel().select(thisProjectEntity.getTargetLanguage() - 1);
        }
    }

    public void bindBidirectional(ProjectModel thisProjectModel) {
        thisProjectNameField.textProperty().bindBidirectional(thisProjectModel.getThisName());
        thisProjectDescriptionField.textProperty().bindBidirectional(thisProjectModel.getThisDescription());
        thisSourceLanguageComboBox.valueProperty().bindBidirectional(thisProjectModel.getThisSourceLanguage());
        thisTargetLanguageComboBox.valueProperty().bindBidirectional(thisProjectModel.getThisTargetLanguage());
    }

    private void initializeComboBox(SearchableComboBox<String> thisComboBox) {
        ArrayList<LanguageEntity> thisLanguageEntities = (ArrayList<LanguageEntity>)
        HibernateGenericDAO.getAll(LanguageEntity.class);
        ArrayList<String> thisLanguageNames = new ArrayList<>();

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if(thisLanguageEntities != null) {
            for(LanguageEntity currentEntity : thisLanguageEntities) {
                thisLanguageNames.add(currentEntity.getName());
            }
        }
        ObservableList<String> thisLanguagesList = FXCollections.observableArrayList(thisLanguageNames);
        thisComboBox.setCellFactory(ComboBoxListCell.forListView(new Callback<ListView<String>,
ListCell<String>>() {
            @Override
            public ListCell<String> call(ListView<String> param) {
                return new ListCell<>() {
                    @Override
                    protected void updateItem(String thisItem, boolean isEmpty) {
                        super.updateItem(thisItem, isEmpty);
                        if(thisItem != null && !isEmpty) {
                            setText(thisItem);
                        } else {
                            setText(null);
                        }
                    }
                }
            }
        });
    }
    }.toString()));
    thisComboBox.setOnShowing(thisEvent -> {
        if(thisComboBox.getSkin() instanceof ComboBoxListViewSkin) {
            ComboBoxListViewSkin<?> thisSkin = (ComboBoxListViewSkin<?>) thisComboBox.getSkin();
            VirtualFlow<?> thisFlow = (VirtualFlow<?>) thisSkin.getChildren().get(0);
            thisFlow.setCellCount(10000);
        }
    });
    thisComboBox.setVisibleRowCount(10);
    thisComboBox.getItems().addAll(
        thisLanguagesList
    );
}
private void initializeSourceLanguageComboBox() {
    initializeComboBox(thisSourceLanguageComboBox);
}
private void initializeTargetLanguageComboBox() {
    initializeComboBox(thisTargetLanguageComboBox);
}
@FXML
private void thisSaveButtonClickHandler() {
    if(thisProjectEntity == null) {
        thisProjectEntity = new ProjectEntity();
    }
    thisProjectModel.update(thisProjectEntity);
    thisProjectEntity = thisProjectEntity.merge();
    System.out.println("Setting id: " + thisProjectEntity.getId());
    thisProjectId = thisProjectEntity.getId();
}
@FXML
private void thisBackButtonClickHandler() throws IOException {
    Stage thisStage = CatToolApp.thisCreateProjectStage;
    if(thisStage != null) {
        thisStage.close();
    }
    CatToolApp.thisApplication.showProjectList();
}
public void thisResourceButtonClickHandler() {
    try {
        CatToolApp.thisApplication.showFilesListWindowID(thisProjectEntity.getId());
    }
}

```

```

    } catch (IOException thisException) {
        throw new RuntimeException(thisException);
    }
}
public void thisDictionaryButtonClickHandler() {
    try {
        CatToolApp.thisApplication.dictionaryWindow();
    } catch (IOException thisException) {
        throw new RuntimeException(thisException);
    }
}
public void thisTranslatorButtonClickHandler() {
    try {
        CatToolApp.thisApplication.showTranslationWindow();
    } catch (IOException thisException) {
        throw new RuntimeException(thisException);
    }
}
public void thisExportButtonClickHandler(ActionEvent thisEvent) {
    DirectoryChooser thisDirectorySelector = new DirectoryChooser();
    thisDirectorySelector.setTitle("Select export location");
    Stage thisStage = (Stage) ((javafx.scene.Node) thisEvent.getSource()).getScene().getWindow();
    File thisDirectory = thisDirectorySelector.showDialog(thisStage);
    if(thisDirectory != null) {
        exportData(thisDirectory);
    } else {
        System.out.println("Not selected dir");
    }
}
private void exportData(File thisDirectory) {
    Session thisSession = getSession();
    CriteriaBuilder thisBuilder = thisSession.getCriteriaBuilder();
    CriteriaQuery<FileEntity> thisQuery = thisBuilder.createQuery(FileEntity.class);
    Root<FileEntity> thisRoot = thisQuery.from(FileEntity.class);
    thisQuery.where(thisBuilder.equal(thisRoot.get("projectId"), thisProjectId));
    ArrayList<FileEntity> thisFilesList = (ArrayList<FileEntity>) thisSession.createQuery(thisQuery).getResultList();
    for(FileEntity currentFileEntity : thisFilesList) {
        ArrayList<LineSourceEntity> thisSourceLines = getLineSource(thisSession, currentFileEntity.getId());
        ArrayList<LineTranslationEntity> thisTranslationLines = getLineTranslation(thisSession,
currentFileEntity.getId());
        ArrayList<String> newSourceLinesList = new ArrayList<String>();
        ArrayList<String> newTranslationLinesList = new ArrayList<String>();
        for(LineSourceEntity currentEntity : thisSourceLines) {
            newSourceLinesList.add(currentEntity.getContent());
        }
        for(LineTranslationEntity currentEntity : thisTranslationLines) {
            newTranslationLinesList.add(currentEntity.getContent());
        }
        writeLines(thisDirectory, "source_" + currentFileEntity.getName(), newSourceLinesList);
        writeLines(thisDirectory, "translation_" + currentFileEntity.getName(), newTranslationLinesList);
    }
}
public static ArrayList<LineSourceEntity> getLineSource(Session thisSession, Integer thisFileId) {
    CriteriaBuilder thisBuilder = thisSession.getCriteriaBuilder();
    CriteriaQuery<LineSourceEntity> thisQuery = thisBuilder.createQuery(LineSourceEntity.class);
    Root<LineSourceEntity> thisRoot = thisQuery.from(LineSourceEntity.class);
    thisQuery.where(thisBuilder.equal(thisRoot.get("fileId"), thisFileId));
    return (ArrayList<LineSourceEntity>) thisSession.createQuery(thisQuery).getResultList();
}
public static ArrayList<LineTranslationEntity> getLineTranslation(Session thisSession, Integer thisFileId) {
    CriteriaBuilder thisBuilder = thisSession.getCriteriaBuilder();
    CriteriaQuery<LineTranslationEntity> thisQuery = thisBuilder.createQuery(LineTranslationEntity.class);

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Root<LineTranslationEntity> thisRoot = thisQuery.from(LineTranslationEntity.class);
thisQuery.where(thisBuilder.equal(thisRoot.get("fileId"), thisFileId));
return (ArrayList<LineTranslationEntity>) thisSession.createQuery(thisQuery).getResultList();
}
public static void writeLines(File thisDirectory, String thisName, ArrayList<String> thisFileLines) {
    Path thisFilePath = Paths.get(thisDirectory.getAbsolutePath(), thisName);
    try(BufferedWriter thisWriter = new BufferedWriter(new FileWriter(thisFilePath.toFile())) {
        for(String currentLine : thisFileLines) {
            thisWriter.write(currentLine);
            thisWriter.newLine();
        }
    } catch(IOException thisException) {
        thisException.printStackTrace();
    }
}
}
}
// File: cat-tool/src/main/java/org/example/cattool/gui/javafx/controller/EditTextWindowController.java
package org.example.cattool.gui.javafx.controller;
import com.dlsc.gemsfx.ExpandingTextArea;
import jakarta.persistence.criteria.CriteriaBuilder;
import jakarta.persistence.criteria.CriteriaQuery;
import jakarta.persistence.criteria.Root;
import java.util.ArrayList;
import javafx.beans.property.IntegerProperty;
import javafx.beans.property.StringProperty;
import javafx.fxml.FXML;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;
import org.example.cattool.database.hibernate.entity.LineSourceEntity;
import org.example.cattool.database.hibernate.entity.LineTranslationEntity;
import org.example.cattool.gui.javafx.model.TranslationPairModel;
import org.example.cattool.gui.javafx.mediator.TranslationMediator;
import org.hibernate.Session;
import static org.example.cattool.database.hibernate.HibernateSession.getSession;
public class EditTextWindowController {
    private ArrayList<LineSourceEntity> thisSourceLineList;
    private ArrayList<LineTranslationEntity> thisTranslationLineList;
    @FXML
    private TableView<TranslationPairModel> thisTranslationTable;
    @FXML
    private TableColumn<TranslationPairModel, Integer> thisIndexColumn;
    @FXML
    private TableColumn<TranslationPairModel, String> thisSourceColumn;
    @FXML
    private TableColumn<TranslationPairModel, String> thisTranslationColumn;
    public void initialize(Integer thisFileId) {
        thisIndexColumn.setCellFactory(thisColumn -> new ExpandingTextAreaTableCell(false, "index"));
        thisIndexColumn.setCellValueFactory(new PropertyValueFactory<>("thisIndex"));
        thisSourceColumn.setCellFactory(thisColumn -> new ExpandingTextAreaTableCell(false, "source"));
        thisSourceColumn.setCellValueFactory(new PropertyValueFactory<>("thisSource"));
        thisTranslationColumn.setCellFactory(thisColumn -> new ExpandingTextAreaTableCell(true, "translation"));
        thisTranslationColumn.setCellValueFactory(new PropertyValueFactory<>("thisTranslation"));
        Session thisSession = getSession();
        CriteriaBuilder thisBuilder = thisSession.getCriteriaBuilder();
        CriteriaQuery<LineSourceEntity> thisQuery = thisBuilder.createQuery(LineSourceEntity.class);
        Root<LineSourceEntity> thisRoot = thisQuery.from(LineSourceEntity.class);
        if(thisFileId != null) {
            thisQuery.where(thisBuilder.equal(thisRoot.get("fileId"), thisFileId));
        }
        thisSourceLineList = (ArrayList<LineSourceEntity>) thisSession.createQuery(thisQuery).getResultList();
        CriteriaBuilder thisBuilder2 = thisSession.getCriteriaBuilder();
        CriteriaQuery<LineTranslationEntity> thisQuery2 = thisBuilder.createQuery(LineTranslationEntity.class);

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Root<LineTranslationEntity> thisRoot2 = thisQuery2.from(LineTranslationEntity.class);
if(thisFileId != null) {
    thisQuery2.where(thisBuilder2.equal(thisRoot2.get("fileId"), thisFileId));
}
thisTranslationLineList = (ArrayList<LineTranslationEntity>) thisSession.createQuery(thisQuery2).getResultList();
int thisMaxIndex = Math.min(thisSourceLineList.size(), thisTranslationLineList.size());
for(int currentIndex = 0; currentIndex < thisMaxIndex; currentIndex++) {
    LineSourceEntity thisSourceEntity = thisSourceLineList.get(currentIndex);
    LineTranslationEntity thisTranslationEntity = thisTranslationLineList.get(currentIndex);
    TranslationPairModel thisModel = new TranslationPairModel();
    thisModel.thisIndex.set(currentIndex);
    thisModel.map(thisSourceEntity, thisTranslationEntity);
    thisTranslationTable.getItems().add(thisModel);
}
thisTranslationTable.focusedProperty().addListener((thisObservable, itsOldValue, itsNewValue) -> {
    TranslationPairModel thisSelectedModel = thisTranslationTable.getSelectionModel().getSelectedItem();
    if(TranslationMediator.thisTranslateTextWindowController != null) {
        TranslationMediator.thisTranslateTextWindowController.setTranslationPairModel(thisSelectedModel);
    }
});
thisTranslationTable.setColumnResizePolicy(Tableview.CONSTRAINED_RESIZE_POLICY);
TranslationMediator.thisTranslationTable = thisTranslationTable;
}
public class ExpandingTextAreaTableCell<T> extends TableCell<TranslationPairModel, T> {
    private ExpandingTextArea thisArea;
    String thisColumn;
    public ExpandingTextAreaTableCell(boolean isEditable, String thisColumn) {
        thisArea = new ExpandingTextArea();
        thisArea.setWrapText(true);
        thisArea.setEditable(isEditable);
        this.thisColumn = thisColumn;
        if(!isEditable) {
            thisArea.setStyle("-fx-border-width: 0; -fx-background-color: transparent;");
        }
        thisArea.textProperty().addListener((thisObservable, itsOldValue, itsNewValue) -> {
            TranslationPairModel thisModel = getTableRow().getItem();
            if(thisModel != null) {
                if(this.thisColumn.equals("source")) {
                    String thisOldModelValue = thisModel.getThisSource().get();
                    if(!thisOldModelValue.equals(itsNewValue)) {
                        thisModel.getThisSource().set(itsNewValue);
                        thisModel.update();
                    }
                }
                if(this.thisColumn.equals("translation")) {
                    String thisOldModelValue = thisModel.getThisSource().get();
                    if(!thisOldModelValue.equals(itsNewValue)) {
                        thisModel.getThisTranslation().set(itsNewValue);
                        thisModel.update();
                    }
                }
            }
        });
        this.setGraphic(thisArea);
        this.setContentDisplay(ContentDisplay.GRAPHIC_ONLY);
    }
    @Override
    protected void updateItem(T thisItem, boolean isEmpty) {
        super.updateItem(thisItem, isEmpty);
        if(isEmpty || thisItem == null) {
            setText(null);
            setGraphic(null);
        }
    }
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        } else {
            if(thisItem instanceof IntegerProperty) {
                thisArea.setText(String.valueOf(((IntegerProperty) thisItem).get()));
            } else if(thisItem instanceof StringProperty) {
                thisArea.setText(String.valueOf(((StringProperty) thisItem).get()));
            }
            setText(null);
            setGraphic(thisArea);
        }
    }
}

// File: cat-tool/src/main/java/org/example/cattool/gui/javafx/controller/ListFilesWindowController.java
package org.example.cattool.gui.javafx.controller;
import jakarta.persistence.criteria.CriteriaBuilder;
import jakarta.persistence.criteria.CriteriaQuery;
import jakarta.persistence.criteria.Root;
import java.io.File;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.scene.control.ListView;
import javafx.scene.input.*;
import javafx.stage.FileChooser;
import javafx.stage.Stage;
import org.example.cattool.database.hibernate.entity.FileEntity;
import org.example.cattool.database.hibernate.entity.LineSourceEntity;
import org.example.cattool.database.hibernate.entity.LineTranslationEntity;
import org.example.cattool.gui.javafx.application.CatToolApp;
import org.example.cattool.database.hibernate.HibernateFilePersistence;
import org.example.cattool.utils.split.SplitFileToSourceLines;
import org.hibernate.Session;
import static org.example.cattool.database.hibernate.HibernateSession.getSession;
public class ListFilesWindowController {
    @FXML
    private ListView<String> thisFileListView;
    private ObservableList<String> thisItems;
    private Stage thisStage;
    private static ArrayList<FileEntity> thisFiles = new ArrayList<FileEntity>();
    private static HashMap<Integer, Integer> thisObservableListToDataBaseMappingIds = new HashMap<Integer,
Integer>();
    private static Integer thisProjectId;
    @FXML
    private void thisHandleDragOverEvent(DragEvent thisEvent) {
        if(thisEvent.getDragboard().hasFiles()) {
            thisEvent.acceptTransferModes(TransferMode.COPY);
        }
        thisEvent.consume();
    }
    @FXML
    private void thisHandleDragDroppedEvent(DragEvent thisEvent) {
        Dragboard thisDragboard = thisEvent.getDragboard();
        boolean isSuccess = false;
        if(thisDragboard.hasFiles()) {
            List<File> thisFiles = thisDragboard.getFiles();
            for(File currentFile : thisFiles) {
                importFile(currentFile.getAbsolutePath());
            }
            isSuccess = true;
        }
    }
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    thisEvent.setDropCompleted(isSuccess);
    thisEvent.consume();
}
private void importFile(String thisPath) {
    System.out.println("Importing new file: " + thisPath);
    HibernateFilePersistence.saveFileWithHibernate(thisPath, thisProjectId);
    initialize(thisProjectId);
}
@FXML
public void initialize(Integer thisId) {
    thisProjectId = thisId;
    if(thisItems != null) {
        thisItems.removeAll();
    }
    thisObservableListToDataBaseMappingIds.clear();
    thisFiles.clear();
    thisItems = FXCollections.observableArrayList();
    thisFileListView.setItems(thisItems);
    Session thisSession = getSession();
    CriteriaBuilder thisBuilder = thisSession.getCriteriaBuilder();
    CriteriaQuery<FileEntity> thisQuery = thisBuilder.createQuery(FileEntity.class);
    Root<FileEntity> thisRoot = thisQuery.from(FileEntity.class);
    if(thisProjectId != null) {
        thisQuery.where(thisBuilder.equal(thisRoot.get("projectId"), thisId));
    }
    thisFiles = (ArrayList<FileEntity>) thisSession.createQuery(thisQuery).getResultList();
    if(thisFiles != null) {
        for(int currentIndex = 0; currentIndex < thisFiles.size(); currentIndex++) {
            FileEntity thisEntity = thisFiles.get(currentIndex);
            Integer thisIdInDatabase = thisEntity.getId();
            Integer thisListId = currentIndex;
            thisObservableListToDataBaseMappingIds.put(thisListId, thisIdInDatabase);
            thisItems.add(thisFiles.get(currentIndex).getName());
        }
    }
}
@FXML
private void thisAddFilesButtonClickHandler() {
    FileChooser thisFileSelector = new FileChooser();
    thisFileSelector.setTitle("Select files to import");
    List<File> thisImportedFiles = thisFileSelector.showOpenMultipleDialog(thisStage);
    if(thisImportedFiles != null) {
        for(File currentFile : thisImportedFiles) {
            thisItems.add(currentFile.getName());
            importFile(currentFile.getAbsolutePath());
        }
    }
}
@FXML
private void thisDeleteFilesButtonClickHandler() {
    int thisSelectedIndex = thisFileListView.getSelectionModel().getSelectedIndex();
    if(thisSelectedIndex >= 0) {
        thisItems.remove(thisSelectedIndex);
        thisObservableListToDataBaseMappingIds.get(thisSelectedIndex);
        thisFiles.get(thisSelectedIndex).delete();
        deleteFileInDatabase(thisFiles.get(thisSelectedIndex));
        thisFiles.remove(thisSelectedIndex);
    }
    initialize(thisProjectId);
}
public static void deleteFileInDatabase(FileEntity thisFileEntity) {

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Session thisSession = getSession();
FileEntity thisFile = thisFileEntity.delete();
Integer thisFileId = thisFile.getId();
CriteriaBuilder thisSourceQueryBuilder = thisSession.getCriteriaBuilder();
CriteriaQuery<LineSourceEntity> thisSourceQuery = thisSourceQueryBuilder.createQuery(LineSourceEntity.class);
Root<LineSourceEntity> thisRoot = thisSourceQuery.from(LineSourceEntity.class);
thisSourceQuery.where(thisSourceQueryBuilder.equal(thisRoot.get("fileId"), thisFileId));
ArrayList<LineSourceEntity> thisSourceEntityList = (ArrayList<LineSourceEntity>)
thisSession.createQuery(thisSourceQuery).getResultList();
for(LineSourceEntity currentEntity : thisSourceEntityList) {
    LineSourceEntity thisLine = currentEntity.delete();
}
CriteriaBuilder thisTranslationQueryBuilder = thisSession.getCriteriaBuilder();
CriteriaQuery<LineTranslationEntity> thisTranslationQuery =
thisTranslationQueryBuilder.createQuery(LineTranslationEntity.class);
Root<LineTranslationEntity> thisTranslationRoot = thisTranslationQuery.from(LineTranslationEntity.class);
thisTranslationQuery.where(thisTranslationQueryBuilder.equal(thisTranslationRoot.get("fileId"), thisFileId));
ArrayList<LineTranslationEntity> lineTranslationEntityList = (ArrayList<LineTranslationEntity>)
thisSession.createQuery(thisTranslationQuery).getResultList();
for(LineTranslationEntity currentEntity : lineTranslationEntityList) {
    LineTranslationEntity thisLine = currentEntity.delete();
}
}
@FXML
private void thisListViewClickHandler(MouseEvent thisEvent) {
    if(thisEvent.getButton() == MouseButton.PRIMARY && thisEvent.getClickCount() == 2) {
        String thisSelectedItem = thisFileListView.getSelectionModel().getSelectedItem();
        if(thisSelectedItem != null) {
            Integer thisIdentifierInDatabase =
thisObservableListToDataBaseMappingIds.get(thisFileListView.getSelectionModel().getSelectedIndex());
            SplitFileToSourceLines.splitToSourceLines(thisIdentifierInDatabase);
            try {
                CatToolApp.thisApplication.showSourceTranslationID(thisIdentifierInDatabase);
            } catch(Exception thisException) {
                thisException.printStackTrace();
            }
        }
    }
}
}
}
}
// File: cat-tool/src/main/java/org/example/cattool/gui/javafx/controller/ListProjectsWindowController.java
package org.example.cattool.gui.javafx.controller;
import jakarta.persistence.criteria.CriteriaBuilder;
import jakarta.persistence.criteria.CriteriaQuery;
import jakarta.persistence.criteria.Root;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.function.Predicate;
import java.util.stream.Collectors;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.fxml.FXML;
import javafx.scene.control.ListCell;
import javafx.scene.control.ListView;
import javafx.scene.control.TextField;
import javafx.scene.input.MouseButton;
import javafx.scene.input.MouseEvent;
import javafx.stage.Stage;
import org.example.cattool.database.hibernate.HibernateGenericDAO;
import org.example.cattool.database.hibernate.entity.*;
import org.example.cattool.gui.javafx.application.CatToolApp;

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		83

```

import org.hibernate.Session;
import static org.example.cattool.database.hibernate.HibernateSession.getSession;
public class ListProjectsWindowController {
    private ObservableList<ProjectEntity> thisProjectEntities = FXCollections.observableArrayList();
    private static HashMap<Integer, Integer> thisObservableListToDataBaseMappingIds = new HashMap<Integer,
Integer>();
    @FXML
    private TextField thisSearchField;
    @FXML
    private ListView<ProjectEntity> thisProjectFilterView;
    public void initialize() {
        thisProjectEntities.clear();
        thisObservableListToDataBaseMappingIds.clear();
        ArrayList<ProjectEntity> thisProjectEntityList = (ArrayList<ProjectEntity>)
HibernateGenericDAO.getAll(ProjectEntity.class);
        for(int currentIndex = 0; currentIndex < thisProjectEntityList.size(); currentIndex++) {
            ProjectEntity thisProjectEntity = thisProjectEntityList.get(currentIndex);
            thisObservableListToDataBaseMappingIds.put(currentIndex, thisProjectEntity.getId());
            thisProjectEntities.add(thisProjectEntity);
        }
        thisProjectFilterView.setItems(thisProjectEntities);
        thisProjectFilterView.setCellFactory(thisColumn -> new ProjectListCell());
    }
    @FXML
    private void thisCreateNewProjectButtonClickHandler() {
        try {
            Stage thisStage = CatToolApp.thisListProjectsStage;
            thisStage.close();
            CatToolApp.thisApplication.showCreateProjectID(null);
        } catch(IOException thisException) {
            thisException.printStackTrace();
        }
    }
    @FXML
    private void thisOpenProjectButtonClickHandler() {
        openProject();
    }
    private void openProject() {
        ProjectEntity thisSelectedProject = thisProjectFilterView.getSelectionModel().getSelectedItem();
        if(thisSelectedProject != null) {
            Integer thisId = thisSelectedProject.getId();
            if(thisId != null) {
                try {
                    CatToolApp.thisApplication.showCreateProjectID(thisSelectedProject.getId());
                } catch(IOException thisException) {
                    throw new RuntimeException(thisException);
                }
            }
        }
    }
    @FXML
    private void thisSearchFieldKeyHandler() {
        String thisSearchText = thisSearchField.getText().toLowerCase().trim();
        if(thisSearchText.isEmpty()) {
            thisProjectFilterView.setItems(thisProjectEntities);
            return;
        }
        Predicate<ProjectEntity> thisSearchPredicate = thisProjectEntity ->
            matchesSearchQuery(thisProjectEntity, thisSearchText);
        ObservableList<ProjectEntity> thisFilteredList = thisProjectEntities.stream()
            .filter(thisSearchPredicate)
            .collect(Collectors.toCollection(FXCollections::observableArrayList));
    }
}

```

```

        thisProjectFilterView.setItems(thisFilteredList);
    }
    private boolean matchesSearchQuery(ProjectEntity thisProjectEntity, String thisSearchQuery) {
        String thisName = thisProjectEntity.getName();
        String thisDescription = thisProjectEntity.getDescription();
        if(thisName != null && thisName.toLowerCase().contains(thisSearchQuery)) {
            return true;
        }
        if(thisDescription != null && thisDescription.toLowerCase().contains(thisSearchQuery)) {
            return true;
        }
        return false;
    }
    public void thisDeleteProjectButtonClickHandler() {
        int thisSelectedIndex = thisProjectFilterView.getSelectionModel().getSelectedIndex();
        deleteProjectInDatabase(thisProjectEntities.get(thisSelectedIndex));
        thisProjectEntities.get(thisSelectedIndex).delete();
        initialize();
    }
    public static void deleteProjectInDatabase(ProjectEntity thisEntity) {
        Integer thisProjectIdToDelete = thisEntity.getId();
        thisEntity = thisEntity.delete();
        deleteProjectFilesInDatabase(thisProjectIdToDelete);
    }
    public static void deleteProjectFilesInDatabase(Integer thisProjectIdToDelete) {
        Session thisSession = getSession();
        CriteriaBuilder thisFileBuilder = thisSession.getCriteriaBuilder();
        CriteriaQuery<FileEntity> thisFileQuery = thisFileBuilder.createQuery(FileEntity.class);
        Root<FileEntity> thisFileRoot = thisFileQuery.from(FileEntity.class);
        thisFileQuery.where(thisFileBuilder.equal(thisFileRoot.get("projectId"), thisProjectIdToDelete));
        ArrayList<FileEntity> thisFileEntities = (ArrayList<FileEntity>)
thisSession.createQuery(thisFileQuery).getResultList();
        for(FileEntity currentFile : thisFileEntities) {
            ListFilesWindowController.deleteFileInDatabase(currentFile);
        }
    }
    @FXML
    private void thisProjectFilterClickHandler(MouseEvent thisEvent) {
        if(thisEvent.getButton() == MouseButton.PRIMARY && thisEvent.getClickCount() == 2) {
            openProject();
        }
    }
    public void thisSettingsButtonClickHandler() {
        System.out.println("Opening Settings");
    }
    private static class ProjectListCell extends ListCell<ProjectEntity> {
        @Override
        protected void updateItem(ProjectEntity thisEntity, boolean isEmpty) {
            super.updateItem(thisEntity, isEmpty);
            if(isEmpty || thisEntity == null) {
                setText(null);
            } else {
                setText(thisEntity.getName());
            }
        }
    }
}
// File: cat-tool/src/main/java/org/example/cattool/gui/javafx/controller/TranslateTextWindowController.java
package org.example.cattool.gui.javafx.controller;
import com.dlsc.gemsfx.ExpandingTextArea;
import java.io.IOException;
import javafx.beans.value.ChangeListener;

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						85
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.fxml.FXML;
import javafx.scene.control.Button;
import javafx.scene.control.ChoiceBox;
import org.example.cattool.gui.javafx.model.TranslationPairModel;
import org.example.cattool.gui.javafx.mediator.TranslationMediator;
import org.example.cattool.translate.glosbe.GlosbeRequest;
import org.example.cattool.translate.mymemory.MyMemoryRequest;
import org.example.cattool.translate.mymemory.deserialization.json.response.ResponseData;
import org.example.cattool.translate.mymemory.deserialization.json.response.TranslationResponse;
import org.example.cattool.translate.pons.PonsRequest;
import org.example.cattool.translate.reverso.ReversoTranslateRequest;
public class TranslateTextWindowController {
    private TranslationPairModel thisTranslationPairModel;
    public static final String thisGlosbe = "Glosbe";
    public static final String thisMyMemory = "MyMemory";
    public static final String thisPons = "Pons";
    public static final String thisReversoContext = "Reverso Context";
    @FXML
    public Button thisUseText;
    @FXML
    private ChoiceBox<String> thisSourceLanguageChoiceBox;
    @FXML
    private ChoiceBox<String> thisTargetLanguageChoiceBox;
    @FXML
    private ExpandingTextArea thisSourceText;
    @FXML
    private ExpandingTextArea thisTranslatedText;
    @FXML
    private ChoiceBox<String> thisTranslatorChoiceBox;
    public void setTranslationPairModel(TranslationPairModel thisModel) {
        this.thisTranslationPairModel = thisModel;
        updateSourceText();
    }
    private void updateSourceText() {
        if(thisTranslationPairModel != null) {
            String thisSourceContent = thisTranslationPairModel.getThisSource().get();
            thisSourceText.setText(thisSourceContent);
        }
    }
    @FXML
    private void initialize() {
        loadTranslatorOptions();
        loadLanguageOptions();
        thisTranslatorChoiceBox.getSelectionModel().selectFirst();
        TranslationMediator.thisTranslateTextWindowController = this;
        thisTranslatorChoiceBox.getSelectionModel().selectedItemProperty().addListener(
            new ChangeListener<String>() {
                @Override
                public void changed(ObservableValue<? extends String> thisObservable, String itsOldValue, String
itsNewValue) {
                    getLanguages();
                }
            }
        );
    }
    private void loadTranslatorOptions() {
        thisTranslatorChoiceBox.getItems().addAll(thisReversoContext, thisGlosbe, thisMyMemory, thisPons);
        thisTranslatorChoiceBox.getSelectionModel().selectFirst();
    }
    private void loadLanguageOptions() {

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						86
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        getLanguages();
    }
    @FXML
    private void thisUseTextButtonClickHandler() {
        String thisTranslated = thisTranslatedText.getText();

        TranslationMediator.thisTranslationTable.getSelectionModel().getSelectedItem().getThisTranslation().set(thisTranslated);
        TranslationMediator.thisTranslationTable.refresh();
    }
    @FXML
    private void thisQueryTranslationButtonClickHandler() {
        translate();
    }
    public void getLanguages() {
        String thisTranslator = thisTranslatorChoiceBox.getSelectionModel().getSelectedItem();
        if(thisTranslator.equals(thisReversoContext)) {

thisSourceLanguageChoiceBox.setItems(FXCollections.observableArrayList(ReversoTranslateRequest.getLanguages()));

thisTargetLanguageChoiceBox.setItems(FXCollections.observableArrayList(ReversoTranslateRequest.getLanguages()));
        } else if(thisTranslator.equals(thisGlosbe)) {
            thisSourceLanguageChoiceBox.setItems(FXCollections.observableArrayList(GlosbeRequest.getLanguages()));
            thisTargetLanguageChoiceBox.setItems(FXCollections.observableArrayList(GlosbeRequest.getLanguages()));
        } else if(thisTranslator.equals(thisMyMemory)) {

thisSourceLanguageChoiceBox.setItems(FXCollections.observableArrayList(MyMemoryRequest.getLanguages()));
thisTargetLanguageChoiceBox.setItems(FXCollections.observableArrayList(MyMemoryRequest.getLanguages()));
        } else if(thisTranslator.equals(thisPons)) {
            thisSourceLanguageChoiceBox.setItems(FXCollections.observableArrayList(PonsRequest.getLanguages()));
            thisTargetLanguageChoiceBox.setItems(FXCollections.observableArrayList(PonsRequest.getLanguages()));
        }
    }
    public void translate() {
        String thisTranslator = thisTranslatorChoiceBox.getSelectionModel().getSelectedItem();
        String thisSourceLanguage = thisSourceLanguageChoiceBox.getValue();
        String thisTargetLanguage = thisTargetLanguageChoiceBox.getValue();
        String thisSource = thisSourceText.getText();
        if(thisTranslator.equals(thisReversoContext)) {
            if(ReversoTranslateRequest.checkDirections(thisSourceLanguage, thisTargetLanguage)) {
                try {
                    thisTranslatedText.setText(
                        ReversoTranslateRequest.parseReversoResponse(
                            ReversoTranslateRequest.reversoTranslate(thisSourceLanguage, thisTargetLanguage, thisSource)
                        ).getTranslation()[0]
                    );
                } catch (IOException thisException) {
                    throw new RuntimeException(thisException);
                }
            }
        } else if(thisTranslator.equals(thisGlosbe)) {
            thisTranslatedText.setText(
                GlosbeRequest.parseGlosbeResponse(
                    GlosbeRequest.callGlosbe(thisSourceLanguage, thisTargetLanguage, thisSource)
                ).getTranslation()
            );
        } else if(thisTranslator.equals(thisMyMemory)) {
            if(thisSource.length() >= 500) {
                thisTranslatedText.setText("[Exception] Max query length is 500 chars");
            } else {
                TranslationResponse thisMyMemoryResponse = MyMemoryRequest.parseMyMemoryResponse(
                    MyMemoryRequest.callMyMemory(thisSourceLanguage, thisTargetLanguage, thisSource)
                );
            }
        }
    }

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						87
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if(thisMyMemoryResponse == null) {
            thisTranslatedText.setText(
                "[Exception] Empty or no response"
            );
        } else {
            ResponseData thisMyMemoryResponseData = thisMyMemoryResponse.getResponseData();
            String thisMyMemoryResponseText = thisMyMemoryResponseData.getTranslatedText();
            if(thisMyMemoryResponseText == null || thisMyMemoryResponseText.isEmpty()) {
                thisTranslatedText.setText(
                    "[Exception] Empty or no response"
                );
            } else {
                thisTranslatedText.setText(
                    thisMyMemoryResponseText
                );
            }
        }
    }
}
else if(thisTranslator.equals(thisPons)) {
    thisTranslatedText.setText(
        PonsRequest.parsePonsResponse(
            PonsRequest.callPons(thisSourceLanguage, thisTargetLanguage, thisSource)
        ).getText()
    );
}
}
}
// File: cat-tool/src/main/java/org/example/cattool/gui/javafx/controller/ViewDictionaryWindowController.java
package org.example.cattool.gui.javafx.controller;
import com.dlsc.gemsfx.ExpandingTextArea;
import java.awt.*;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import javafx.fxml.FXML;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.CheckBox;
import javafx.scene.control.ComboBox;
import javafx.scene.input.Clipboard;
import javafx.scene.web.WebView;
import javafx.stage.Stage;
public class ViewDictionaryWindowController {
    public static final String thisDictionaryCambridgeOrg = "https://dictionary.cambridge.org/dictionary/english/";
    public static final String thisDictionaryCom = "https://www.dictionary.com/browse/";
    public static final String thisLdoceonlineCom = "https://www.ldoceonline.com/dictionary/";
    public static final String thisMerriamWebsterCom = "https://www.merriam-webster.com/dictionary/";
    public static final String thisOxfordLearnersDictionariesCom =
"https://www.oxfordlearnersdictionaries.com/definition/english/";
    @FXML
    public Button thisClipboardButton;
    @FXML
    private CheckBox thisOpenInBrowserCheckbox;
    @FXML
    private ExpandingTextArea thisCopiedElementField;
    @FXML
    private ComboBox<String> thisUseDictionaryComboBox;
    public static String openDictionaryCambridgeOrg(String thisClipboard) {
        return thisDictionaryCambridgeOrg + thisClipboard;
    }
    public static String openDictionaryCom(String thisClipboard) {

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						88
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return thisDictionaryCom + thisClipboard;
    }
    public static String openMerriamWebsterCom(String thisClipboard) {
        return thisMerriamWebsterCom + thisClipboard;
    }
    public static String openOxfordLearnersDictionariesCom(String thisClipboard) {
        return thisOxfordLearnersDictionariesCom + thisClipboard + "?q=" + thisClipboard;
    }
    public static String openLdoceonlineCom(String thisClipboard) {
        return thisLdoceonlineCom + thisClipboard;
    }
    @FXML
    public void initialize() {
        thisUseDictionaryComboBox.getItems().addAll(
            "dictionary.com",
            "ldoceonline.com",
            "merriam-webster.com",
            "oxfordlearnersdictionaries.com",
            "dictionary.cambridge.org"
        );
    }
    public static String getClipboard() {
        Clipboard thisClipboard = Clipboard.getSystemClipboard();
        if(thisClipboard.hasString()) {
            return thisClipboard.getString();
        } else return "Empty Clipboard";
    }
    @FXML
    public void thisSearchButtonClickHandler() {
        String currentSelectedDictionary = thisUseDictionaryComboBox.getValue();
        String thisCopiedElement = thisCopiedElementField.getText();
        boolean openInBrowserIsSelected = thisOpenInBrowserCheckbox.isSelected();
        String thisLink = "";
        switch(currentSelectedDictionary) {
            case "dictionary.cambridge.org":
                thisLink = openDictionaryCambridgeOrg(thisCopiedElement);
                break;
            case "dictionary.com":
                thisLink = openDictionaryCom(thisCopiedElement);
                break;
            case "merriam-webster.com":
                thisLink = openMerriamWebsterCom(thisCopiedElement);
                break;
            case "oxfordlearnersdictionaries.com":
                thisLink = openOxfordLearnersDictionariesCom(thisCopiedElement);
                break;
            case "ldoceonline.com":
                thisLink = openLdoceonlineCom(thisCopiedElement);
                break;
        }
        if(openInBrowserIsSelected) {
            openInExternalBrowser(thisLink);
        } else {
            openInBuiltInWindow(thisLink);
        }
    }
    public static void openInExternalBrowser(String thisLink) {
        String thisOperatingSystem = System.getProperty("os.name").toLowerCase();
        if(thisOperatingSystem.contains("win")) {
            try {
                if(Desktop.isDesktopSupported()) {
                    Desktop thisDesktop = Desktop.getDesktop();

```

```

        if(thisDesktop.isSupported(Desktop.Action.BROWSE)) {
            URI thisURI = new URI(thisLink);
            thisDesktop.browse(thisURI);
        } else {
            System.out.println("Browsing not supported");
        }
    } else {
        System.out.println("Desktop not supported");
    }
} catch(IOException | URISyntaxException thisException) {
    thisException.printStackTrace();
}
}
}
else if(thisOperatingSystem.contains("mac")) {
    try {
        Runtime.getRuntime().exec("open " + thisLink);
    } catch(IOException thisException) {
        throw new RuntimeException(thisException);
    }
}
else if(thisOperatingSystem.contains("nux") || thisOperatingSystem.contains("nix")) {
    try {
        Runtime.getRuntime().exec("xdg-open " + thisLink);
    } catch(IOException thisException) {
        throw new RuntimeException(thisException);
    }
}
}
}
public static void openInBuiltInWindow(String thisLink) {
    javafx.application.Platform.runLater() -> {
        WebView thisView = new WebView();
        Scene thisScene = new Scene(thisView);
        Stage thisStage = new Stage();
        thisStage.setTitle("Dictionary");
        thisStage.setScene(thisScene);
        thisView.getEngine().load(thisLink);
        thisStage.show();
    });
}
@FXML
public void thisClipboardButtonClickHandler() {
    thisCopiedElementField.setText(getClipboard());
}
}
// File: cat-tool/src/main/java/org/example/cattool/gui/javafx/mediator/TranslationMediator.java
package org.example.cattool.gui.javafx.mediator;
import javafx.scene.control.TableView;
import org.example.cattool.gui.javafx.controller.TranslateTextWindowController;
import org.example.cattool.gui.javafx.model.TranslationPairModel;
public class TranslationMediator {
    public static TranslateTextWindowController thisTranslateTextWindowController;
    public static TableView<TranslationPairModel> thisTranslationTable;
}
// File: cat-tool/src/main/java/org/example/cattool/gui/javafx/model/ProjectModel.java
package org.example.cattool.gui.javafx.model;
import jakarta.persistence.criteria.CriteriaBuilder;
import jakarta.persistence.criteria.CriteriaQuery;
import jakarta.persistence.criteria.Root;
import javafx.beans.property.SimpleStringProperty;
import javafx.beans.property.StringProperty;
import lombok.*;
import org.example.cattool.database.hibernate.entity.LanguageEntity;

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import org.example.cattool.database.hibernate.entity.ProjectEntity;
import org.hibernate.Session;
import static org.example.cattool.database.hibernate.HibernateSession.getSession;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
public class ProjectModel {
    public StringProperty thisDescription = new SimpleStringProperty();
    public StringProperty thisName = new SimpleStringProperty();
    public StringProperty thisSourceLanguage = new SimpleStringProperty();
    public StringProperty thisTargetLanguage = new SimpleStringProperty();
    static LanguageEntity thisLanguageEntity = new LanguageEntity();
    public void map(ProjectEntity thisProjectEntity) {
        thisDescription.set(thisProjectEntity.getDescription());
        thisName.set(thisProjectEntity.getName());
        thisSourceLanguage.set(getLanguage(thisProjectEntity.getSourceLanguage()));
        thisTargetLanguage.set(getLanguage(thisProjectEntity.getTargetLanguage()));
    }
    public void update(ProjectEntity thisProjectEntity) {
        thisProjectEntity.setDescription(thisDescription.get());
        thisProjectEntity.setName(thisName.get());
        thisProjectEntity.setSourceLanguage(getLanguageId(thisSourceLanguage.get()));
        thisProjectEntity.setTargetLanguage(getLanguageId(thisTargetLanguage.get()));
    }
    public static String getLanguage(Integer thisId) {
        thisLanguageEntity.get(thisId);
        return thisLanguageEntity.getName();
    }
}
public static Integer getLanguageId(String thisLanguage) {
    Session thisSession = getSession();
    CriteriaBuilder thisLanguageBuilder = thisSession.getCriteriaBuilder();
    CriteriaQuery<LanguageEntity> thisLanguageQuery = thisLanguageBuilder.createQuery(LanguageEntity.class);
    Root<LanguageEntity> thisLanguageRoot = thisLanguageQuery.from(LanguageEntity.class);
    thisLanguageQuery.where(thisLanguageBuilder.equal(thisLanguageRoot.get("name"), thisLanguage));
    thisLanguageEntity = thisSession.createQuery(thisLanguageQuery).getSingleResult();
    return thisLanguageEntity.getId();
}
}
// File: cat-tool/src/main/java/org/example/cattool/gui/javafx/model/TranslationPairModel.java
package org.example.cattool.gui.javafx.model;
import javafx.beans.property.IntegerProperty;
import javafx.beans.property.SimpleIntegerProperty;
import javafx.beans.property.SimpleStringProperty;
import javafx.beans.property.StringProperty;
import lombok.*;
import org.example.cattool.database.hibernate.entity.LineSourceEntity;
import org.example.cattool.database.hibernate.entity.LineTranslationEntity;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
public class TranslationPairModel {
    public IntegerProperty thisIndex = new SimpleIntegerProperty();
    public LineSourceEntity thisLineSourceEntity;
    public LineTranslationEntity thisLineTranslationEntity;
    public StringProperty thisSource = new SimpleStringProperty();
    public StringProperty thisTranslation = new SimpleStringProperty();
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						91
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    public void map(LineSourceEntity thisTranslationLineSourceEntity, LineTranslationEntity
thisTranslationLineTranslationEntity) {
        thisSource.set(thisTranslationLineSourceEntity.getContent());
        thisTranslation.set(thisTranslationLineTranslationEntity.getContent());
        this.thisLineSourceEntity = thisTranslationLineSourceEntity;
        thisLineTranslationEntity = thisTranslationLineTranslationEntity;
    }
    public void update() {
        this.thisLineSourceEntity.setContent(thisSource.get());
        this.thisLineTranslationEntity.setContent(thisTranslation.get());
        thisLineSourceEntity.mergeOrPersist();
        thisLineTranslationEntity.mergeOrPersist();
    }
}
// File: cat-tool/src/main/java/org/example/cattool/translate/glosbe/deserialization/json/languages/LanguageData.java
package org.example.cattool.translate.glosbe.deserialization.json.languages;
import com.fasterxml.jackson.annotation.JsonProperty;
import java.util.List;
import lombok.*;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
public class LanguageData {
    @JsonProperty("Languages")
    private List<Language> languages;
}
// File: cat-tool/src/main/java/org/example/cattool/translate/glosbe/deserialization/json/languages/Language.java
package org.example.cattool.translate.glosbe.deserialization.json.languages;
import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.*;
import java.util.List;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
public class Language {
    @JsonProperty("code")
    private String code;
    @JsonProperty("iso693_3")
    private String iso693_3;
    @JsonProperty("name")
    private String name;
    @JsonProperty("alternativeNames")
    private List<String> alternativeNames;
}
// File: cat-tool/src/main/java/org/example/cattool/translate/glosbe/deserialization/json/response/TranslationData.java
package org.example.cattool.translate.glosbe.deserialization.json.response;
import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.*;
import java.util.List;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
public class TranslationData {

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дата		

```

@JsonProperty("translation")
private String translation;
@JsonProperty("scores")
private List<Double> scores;
}
// File: cat-tool/src/main/java/org/example/cattool/translate/glosbe/GlosbeRequest.java
package org.example.cattool.translate.glosbe;
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import org.example.cattool.utils.json.ReadJSON;
import org.example.cattool.translate.glosbe.deserialization.json.languages.Language;
import org.example.cattool.translate.glosbe.deserialization.json.languages.LanguageData;
import org.example.cattool.translate.glosbe.deserialization.json.response.TranslationData;
public class GlosbeRequest {
    public static ArrayList<String> thisGlosbeLanguagesList;
    public static LanguageData thisGlosbeLanguageData;
    public static String convertFullToShort(String thisLanguage) {
        if(thisGlosbeLanguageData == null) {
            readGlosbeLanguagesFromJson();
        }
        for(Language currentLanguage : thisGlosbeLanguageData.getLanguages()) {
            if(thisLanguage.equals(currentLanguage.getName())) {
                return currentLanguage.getCode();
            }
        }
        return null;
    }
    public static ArrayList<String> getLanguages() {
        if(thisGlosbeLanguageData == null) {
            readGlosbeLanguagesFromJson();
        }
        if(thisGlosbeLanguagesList == null) {
            thisGlosbeLanguagesList = new ArrayList<String>();
            for(Language currentLanguage : thisGlosbeLanguageData.getLanguages()) {
                thisGlosbeLanguagesList.add(currentLanguage.getName());
            }
            return thisGlosbeLanguagesList;
        } else {
            return thisGlosbeLanguagesList;
        }
    }
    public static void readGlosbeLanguagesFromJson() {
        try {
            ReadJSON thisReader = new ReadJSON();
            String thisJson = thisReader.readFromPath("/json/translators/glosbe.json");
            ObjectMapper thisMapper = new ObjectMapper();
            thisGlosbeLanguageData = thisMapper.readValue(thisJson, LanguageData.class);
        } catch (JsonProcessingException thisException) {
            throw new RuntimeException(thisException);
        }
    }
    public static TranslationData parseGlosbeResponse(String thisResponseToParse) {
        ObjectMapper thisMapper = new ObjectMapper();
        try {
            return thisMapper.readValue(thisResponseToParse, TranslationData.class);
        }
    }
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						93
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    } catch(Exception thisException) {
        thisException.printStackTrace();
    }
    return null;
}
public static String callGlosbe(String fromThisLanguage, String toThisLanguage, String thisTextToTranslate) {
    try {
        String thisUrlAddress = "https://translator-api.glosbe.com/translateByLangWithScore?" +
            "sourceLang="+convertFullToShort(fromThisLanguage)+
            "&targetLang="+ convertFullToShort(toThisLanguage);
        URL thisURL = new URL(thisUrlAddress);
        HttpURLConnection thisConnection = (HttpURLConnection) thisURL.openConnection();
        thisConnection.setRequestProperty("content-type", "text/plain;charset=UTF-8");
        thisConnection.setDoOutput(true);
        byte[] thisByteDataPOST = thisTextToTranslate.getBytes(StandardCharsets.UTF_8);
        thisConnection.getOutputStream().write(thisByteDataPOST);
        BufferedReader thisReader = new BufferedReader(new InputStreamReader(thisConnection.getInputStream()));
        String thisLine;
        StringBuilder thisResponse = new StringBuilder();
        while((thisLine = thisReader.readLine()) != null) {
            thisResponse.append(thisLine);
        }
        thisReader.close();
        thisConnection.disconnect();
        return thisResponse.toString();
    } catch(IOException thisException) {
        thisException.printStackTrace();
    }
    return null;
}
}
// File: cat-tool/src/main/java/org/example/cattool/translate/mymemory/deserialization/json/languages/LanguageData.java
package org.example.cattool.translate.mymemory.deserialization.json.languages;
import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.*;
import java.util.List;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
public class LanguageData {
    private List<Language> languages;
    @Data
    @EqualsAndHashCode
    @Getter
    @RequiredArgsConstructor
    @Setter
    @ToString
    public static class Language {
        @JsonProperty("value")
        private String value;
        @JsonProperty("label")
        private String label;
    }
}
// File: cat-tool/src/main/java/org/example/cattool/translate/mymemory/deserialization/json/languages/Language.java
package org.example.cattool.translate.mymemory.deserialization.json.languages;
import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.*;
@Data

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						94
Змн.	Арк.	№ докум.	Підпис	Дата		

```

@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
public class Language {
    @JsonProperty("value")
    private String value;
    @JsonProperty("label")
    private String label;
}

// File: cat-tool/src/main/java/org/example/cattool/translate/mymemory/deserialization/json/response/Match.java
package org.example.cattool.translate.mymemory.deserialization.json.response;
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.*;

@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
@JsonIgnoreProperties(ignoreUnknown = true)
public class Match {
    @JsonProperty("id")
    private String id;
    @JsonProperty("segment")
    private String segment;
    @JsonProperty("translation")
    private String translation;
    @JsonProperty("source")
    private String source;
    @JsonProperty("target")
    private String target;
    @JsonProperty("quality")
    private int quality;
    @JsonProperty("reference")
    private String reference;
    @JsonProperty("usage-count")
    private int usageCount;
    @JsonProperty("subject")
    private String subject;
    @JsonProperty("created-by")
    private String createdBy;
    @JsonProperty("last-updated-by")
    private String lastUpdatedBy;
    @JsonProperty("create-date")
    private String createDate;
    @JsonProperty("last-update-date")
    private String lastUpdateDate;
    @JsonProperty("match")
    private int match;
}

// File: cat-tool/src/main/java/org/example/cattool/translate/mymemory/deserialization/json/response/ResponseData.java
package org.example.cattool.translate.mymemory.deserialization.json.response;
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.*;

@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						95
Змн.	Арк.	№ докум.	Підпис	Дата		

```

@Setter
@ToString
@JsonIgnoreProperties(ignoreUnknown = true)
public class ResponseData {
    @JsonProperty("translatedText")
    private String translatedText;
    @JsonProperty("match")
    private int match;
}
// File: cat-
tool/src/main/java/org/example/cattool/translate/mymemory/deserialization/json/response/TranslationResponse.java
package org.example.cattool.translate.mymemory.deserialization.json.response;
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.*;
import java.util.List;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
@JsonIgnoreProperties(ignoreUnknown = true)
public class TranslationResponse {
    @JsonProperty("responseData")
    private ResponseData responseData;
    @JsonProperty("quotaFinished")
    private boolean quotaFinished;
    @JsonProperty("matches")
    private List<Match> matches;
}
// File: cat-tool/src/main/java/org/example/cattool/translate/mymemory/MyMemoryRequest.java
package org.example.cattool.translate.mymemory;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import org.example.cattool.utils.json.ReadJSON;
import org.example.cattool.translate.mymemory.deserialization.json.languages.LanguageData.Language;
import org.example.cattool.translate.mymemory.deserialization.json.languages.LanguageData;
import org.example.cattool.translate.mymemory.deserialization.json.response.TranslationResponse;
import static org.apache.commons.text.StringEscapeUtils.*;
public class MyMemoryRequest {
    public static ArrayList<String> thisMyMemoryLanguages;
    public static LanguageData thisMyMemoryLanguageData;
    public static TranslationResponse parseMyMemoryResponse(String thisResponseToParse) {
        ObjectMapper thisMapper = new ObjectMapper();
        try {
            return thisMapper.readValue(thisResponseToParse, TranslationResponse.class);
        } catch (Exception thisException) {
            thisException.printStackTrace();
        }
        return null;
    }
    public static ArrayList<String> getLanguages() {
        if(thisMyMemoryLanguageData == null) {
            readMyMemoryLanguagesFromJson();
        }
    }
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						96
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    if(thisMyMemoryLanguages == null) {
        thisMyMemoryLanguages = new ArrayList<String>();
        for(Language currentLanguage : thisMyMemoryLanguageData.getLanguages()) {
            thisMyMemoryLanguages.add(currentLanguage.getLabel());
        }
        return thisMyMemoryLanguages;
    } else {
        return thisMyMemoryLanguages;
    }
}

public static void readMyMemoryLanguagesFromJson() {
    try {
        ReadJSON thisReader = new ReadJSON();
        String thisJson = thisReader.readFromPath("/json/translators/mymemory.json");
        ObjectMapper thisMapper = new ObjectMapper();
        thisMyMemoryLanguageData = thisMapper.readValue(thisJson, LanguageData.class);
    } catch(IOException thisException) {
        throw new RuntimeException(thisException);
    }
}

public static String encodeTextForCall(String thisTextToEncode) {
    return URLEncoder.encode(thisTextToEncode, StandardCharsets.UTF_8);
}

public static String callMyMemory(String fromThisLanguage, String toThisLanguage, String thisTextToTranslate) {
    try {
        URL thisURL = new URL(
            "https://api.mymemory.translated.net/get?q=" +
            encodeTextForCall(thisTextToTranslate) +
            "&langpair=" + fromThisLanguage +
            "|" + toThisLanguage);
        HttpURLConnection thisConnection = (HttpURLConnection) thisURL.openConnection();
        thisConnection.setRequestMethod("GET");
        BufferedReader thisReader = new BufferedReader(new InputStreamReader(thisConnection.getInputStream()));
        String thisLine;
        StringBuilder thisResponse = new StringBuilder();
        while((thisLine = thisReader.readLine()) != null) {
            thisResponse.append(thisLine);
        }
        thisReader.close();
        return unescapeJava(thisResponse.toString());
    } catch(IOException thisException) {
        thisException.printStackTrace();
    }
    return null;
}
}

// File: cat-tool/src/main/java/org/example/cattool/translate/pons/deserialization/json/languages/LanguageData.java
package org.example.cattool.translate.pons.deserialization.json.languages;
import lombok.*;
import java.util.List;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
public class LanguageData {
    private List<Language> languages;
}

// File: cat-tool/src/main/java/org/example/cattool/translate/pons/deserialization/json/languages/Language.java
package org.example.cattool.translate.pons.deserialization.json.languages;

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						97
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import lombok.*;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
public class Language {
    private String code;
    private String dir;
    private String display;
}
// File: cat-tool/src/main/java/org/example/cattool/translate/pons/deserialization/json/response/TranslationData.java
package org.example.cattool.translate.pons.deserialization.json.response;
import com.fasterxml.jackson.annotation.JsonIgnore;
import lombok.*;
import java.util.Map;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
public class TranslationData {
    private String requestId;
    private String sourceLanguage;
    private String targetLanguage;
    private String serviceMessage;
    private String text;
    @JsonIgnore
    private Map<String, String> links;
}
// File: cat-tool/src/main/java/org/example/cattool/translate/pons/PonsRequest.java
package org.example.cattool.translate.pons;
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import org.example.cattool.utils.json.ReadJSON;
import org.example.cattool.translate.pons.deserialization.json.languages.Language;
import org.example.cattool.translate.pons.deserialization.json.languages.LanguageData;
import org.example.cattool.translate.pons.deserialization.json.response.TranslationData;
public class PonsRequest {
    public static LanguageData thisPonsLanguageData;
    public static ArrayList<String> thisPonsLanguages;
    public static String convertFullToShort(String thisLanguage) {
        if(thisPonsLanguageData == null) {
            readPonsLanguagesFromJson();
        }
        for(Language currentLanguage : thisPonsLanguageData.getLanguages()) {
            if(thisLanguage.equals(currentLanguage.getDisplay())) {
                return currentLanguage.getCode();
            }
        }
        return null;
    }
    public static TranslationData parsePonsResponse(String thisResponseToParse) {

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						98
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    ObjectMapper thisMapper = new ObjectMapper();
    try {
        return thisMapper.readValue(thisResponseToParse, TranslationData.class);
    } catch (Exception thisException) {
        thisException.printStackTrace();
    }
    return null;
}

public static ArrayList<String> getLanguages() {
    if (thisPonsLanguageData == null) {
        readPonsLanguagesFromJson();
    }
    if (thisPonsLanguages == null) {
        thisPonsLanguages = new ArrayList<String>();
        for (Language currentLanguage : thisPonsLanguageData.getLanguages()) {
            thisPonsLanguages.add(currentLanguage.getDisplay());
        }
        return thisPonsLanguages;
    } else {
        return thisPonsLanguages;
    }
}

public static void readPonsLanguagesFromJson() {
    try {
        ReadJSON thisReader = new ReadJSON();
        String thisJson = thisReader.readFromPath("/json/translators/pons.json");
        ObjectMapper thisMapper = new ObjectMapper();
        thisPonsLanguageData = thisMapper.readValue(thisJson, LanguageData.class);
    } catch (JsonProcessingException thisException) {
        throw new RuntimeException(thisException);
    }
}

public static String callPons(String fromThisLanguage, String toThisLanguage, String thisTextToTranslate) {
    try {
        URL thisURL = new URL(
            "https://api.pons.com/text-translation-web/v4/translate?locale=" +
            convertFullToShort(fromThisLanguage));
        HttpURLConnection thisConnection = (HttpURLConnection) thisURL.openConnection();
        thisConnection.setRequestProperty("content-type", "application/json");
        thisConnection.setDoOutput(true);
        String thisJsonRequestString =
            "{ \"targetLanguage\": \"" +
            convertFullToShort(toThisLanguage) +
            "\", \"text\": \"" + thisTextToTranslate +
            "\" }";
        try (OutputStream thisStream = thisConnection.getOutputStream()) {
            byte[] thisByteInput = thisJsonRequestString.getBytes(StandardCharsets.UTF_8);
            thisStream.write(thisByteInput, 0, thisByteInput.length);
        }
        StringBuilder thisResponse = new StringBuilder();
        try (BufferedReader thisReader = new BufferedReader(
            new InputStreamReader(thisConnection.getInputStream(), StandardCharsets.UTF_8))) {
            String thisResponseLine;
            while ((thisResponseLine = thisReader.readLine()) != null) {
                thisResponse.append(thisResponseLine.trim());
            }
        }
        thisConnection.disconnect();
        return thisResponse.toString();
    } catch (Exception thisException) {
        thisException.printStackTrace();
    }
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						99
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return "";
    }
}
// File: cat-tool/src/main/java/org/example/cattool/translate/reverso/deserialization/json/languages/LanguageData.java
package org.example.cattool.translate.reverso.deserialization.json.languages;
import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.*;
import java.util.List;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
public class LanguageData {
    @JsonProperty("Languages")
    private List<Language> Languages;
    @JsonProperty("Directions")
    private List<String> Directions;
}
// File: cat-tool/src/main/java/org/example/cattool/translate/reverso/deserialization/json/languages/Language.java
package org.example.cattool.translate.reverso.deserialization.json.languages;
import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.*;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
public class Language {
    @JsonProperty("Id")
    private String Id;
    @JsonProperty("Name")
    private String Name;
    @JsonProperty("IsLtr")
    private boolean IsLtr;
}
// File: cat-tool/src/main/java/org/example/cattool/translate/reverso/deserialization/json/response/TranslationData.java
package org.example.cattool.translate.reverso.deserialization.json.response;
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.*;
@Data
@EqualsAndHashCode
@Getter
@RequiredArgsConstructor
@Setter
@ToString
@JsonIgnoreProperties(ignoreUnknown = true)
public class TranslationData {
    @JsonProperty("id")
    private String id;
    @JsonProperty("from")
    private String from;
    @JsonProperty("to")
    private String to;
    @JsonProperty("input")
    private String[] input;
    @JsonProperty("correctedText")
    private String correctedText;
    @JsonProperty("translation")

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						100
Змн.	Арк.	№ докум.	Підпис	Дата		

```

private String[] translation;
@JsonProperty("engines")
private String[] engines;
@JsonProperty("languageDetection")
private Object languageDetection;
@JsonProperty("contextResults")
private Object contextResults;
@JsonProperty("truncated")
private boolean truncated;
@JsonProperty("timeTaken")
private int timeTaken;
}
// File: cat-tool/src/main/java/org/example/cattool/translate/reverso/ReversoTranslateRequest.java
package org.example.cattool.translate.reverso;
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import org.example.cattool.utils.json.ReadJSON;
import org.example.cattool.translate.reverso.deserialization.json.languages.Language;
import org.example.cattool.translate.reverso.deserialization.json.languages.LanguageData;
import org.example.cattool.translate.reverso.deserialization.json.response.TranslationData;
public class ReversoTranslateRequest {
    public static ArrayList<String> thisReversoLanguagePairs;
    public static ArrayList<String> thisReversoLanguages;
    public static LanguageData thisReversoLanguageData;
    public static void readReversoLanguagesFromJson() {
        try {
            ReadJSON thisReader = new ReadJSON();
            String thisJson = thisReader.readFromPath("/json/translators/reverso.json");
            ObjectMapper thisMapper = new ObjectMapper();
            thisReversoLanguageData = thisMapper.readValue(thisJson, LanguageData.class);
        } catch (JsonProcessingException thisException) {
            throw new RuntimeException(thisException);
        }
    }
}
public static String convertFullToShort(String thisLanguage) {
    if(thisReversoLanguageData == null) {
        readReversoLanguagesFromJson();
    }
    for(Language currentLanguage : thisReversoLanguageData.getLanguages()) {
        if(thisLanguage.equals(currentLanguage.getName())) {
            return currentLanguage.getId();
        }
    }
    return null;
}
public static ArrayList<String> getLanguages() {
    if(thisReversoLanguageData == null) {
        readReversoLanguagesFromJson();
    }
    if(thisReversoLanguages == null) {
        thisReversoLanguages = new ArrayList<String>();
        for(Language currentLanguage : thisReversoLanguageData.getLanguages()) {
            thisReversoLanguages.add(currentLanguage.getName());
        }
    }
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						101
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return thisReversoLanguages;
    } else {
        return thisReversoLanguages;
    }
}
public static ArrayList<String> getPairs() {
    if(thisReversoLanguageData == null) {
        readReversoLanguagesFromJson();
    }
    if(thisReversoLanguagePairs == null) {
        thisReversoLanguagePairs = new ArrayList<String>();
        for(String currentString : thisReversoLanguageData.getDirections()) {
            thisReversoLanguagePairs.add(currentString);
        }
        return thisReversoLanguagePairs;
    } else {
        return thisReversoLanguagePairs;
    }
}
}
public static boolean checkDirections(String thisLanguageA, String thisLanguageB) {
    if(thisReversoLanguageData == null) {
        readReversoLanguagesFromJson();
    }
    String thisIdentifierA = "";
    String thisIdentifierB = "";
    for(Language currentLanguage : thisReversoLanguageData.getLanguages()) {
        if(thisLanguageA.equals(currentLanguage.getName())) {
            thisIdentifierA = currentLanguage.getId();
        }
        if(thisLanguageB.equals(currentLanguage.getName())) {
            thisIdentifierB = currentLanguage.getId();
        }
    }
    if(!thisIdentifierA.isEmpty() && !thisIdentifierB.isEmpty()) {
        for(String currentString : getPairs()) {
            if(currentString.equals(thisIdentifierA + "-" + thisIdentifierB)) {
                return true;
            }
        }
    }
    return false;
}
}
public static TranslationData parseReversoResponse(String thisResponseToParse) {
    ObjectMapper thisMapper = new ObjectMapper();
    try {
        return thisMapper.readValue(thisResponseToParse, TranslationData.class);
    } catch(Exception thisException) {
        thisException.printStackTrace();
    }
    return null;
}
}
public static String reversoTranslate(String fromThisLanguage, String toThisLanguage, String thisTextToTranslate)
throws IOException {
    URL thisURL = new URL("https://api.reverso.net/translate/v1/translation");
    HttpURLConnection thisConnection = (HttpURLConnection) thisURL.openConnection();
    thisConnection.setRequestProperty("content-type", "application/json");
    thisConnection.setRequestProperty("user-agent", "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36");
    thisConnection.setDoOutput(true);
    String thisRequestJSON =
        "{" +
        "\"format\":\"text\"," +

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						102
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        "\"from\": \"" + convertFullToShort(fromThisLanguage) + "\", " +
        "\"to\": \"" + convertFullToShort(toThisLanguage) + "\", " +
        "\"input\": \"" + thisTextToTranslate + "\", " +
        "\"options\": " +
        "{" +
        "\"sentenceSplitter\": false, " +
        "\"origin\": \"translation.web\", " +
        "\"contextResults\": false, " +
        "\"languageDetection\": false" +
        "}" +
        "};

try(OutputStream thisStream = thisConnection.getOutputStream()) {
    byte[] thisByteInput = thisRequestJSON.getBytes(StandardCharsets.UTF_8);
    thisStream.write(thisByteInput, 0, thisByteInput.length);
}
try(BufferedReader thisReader = new BufferedReader(new InputStreamReader(thisConnection.getInputStream(),
StandardCharsets.UTF_8))) {
    StringBuilder thisResponse = new StringBuilder();
    String thisResponseLine;
    while((thisResponseLine = thisReader.readLine()) != null) {
        thisResponse.append(thisResponseLine.trim());
    }
    thisConnection.disconnect();
    return thisResponse.toString();
}
}
}

// File: cat-tool/src/main/java/org/example/cattool/utils/json/ReadJSON.java
package org.example.cattool.utils.json;
import com.fasterxml.jackson.core.type.TypeReference;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import java.util.List;
import org.example.cattool.database.hibernate.HibernateGenericDAO;
import org.example.cattool.database.hibernate.entity.LanguageEntity;
public class ReadJSON {
    public String readFromPath(String thisFilePath) {
        try(InputStream thisInStream = ReadJSON.class.getResourceAsStream(thisFilePath)) {
            if(thisInStream != null) {
                StringBuilder thisBuilder = new StringBuilder();
                try(BufferedReader thisReader = new BufferedReader(new InputStreamReader(thisInStream,
StandardCharsets.UTF_8))) {
                    String thisLine;
                    while((thisLine = thisReader.readLine()) != null) {
                        thisBuilder.append(thisLine);
                    }
                }
                return thisBuilder.toString();
            } else {
                System.out.println("File not found: " + thisFilePath);
            }
        } catch(IOException thisException) {
            thisException.printStackTrace();
        }
        return null;
    }
    public static void initializeLanguages() {

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						103
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if(!checkIfLanguagesAlreadyExist()) {
    try {
        ReadJSON thisReader = new ReadJSON();
        String thisString = thisReader.readFromPath("/json/languages.json");
        ObjectMapper thisMapper = new ObjectMapper();
        ArrayList<LanguageEntity> thisLanguageEntities = (ArrayList<LanguageEntity>)
            thisMapper.readValue(thisString, new TypeReference<List<LanguageEntity>>() {});
        for(int currentIndex = 0; currentIndex < thisLanguageEntities.size(); currentIndex++) {
            LanguageEntity thisLanguageEntity = thisLanguageEntities.get(currentIndex);
            thisLanguageEntity.setIdx(currentIndex);
            thisLanguageEntities.set(0, thisLanguageEntity);
            thisLanguageEntity.merge();
        }
    } catch(IOException thisException) {
        thisException.printStackTrace();
    }
}
}

public static boolean checkIfLanguagesAlreadyExist() {
    ArrayList<LanguageEntity> thisLanguageEntityList = (ArrayList<LanguageEntity>)
HibernateGenericDAO.getAll(LanguageEntity.class);
    if(thisLanguageEntityList == null || thisLanguageEntityList.isEmpty()) {
        return false;
    } else {
        return true;
    }
}
}

// File: cat-tool/src/main/java/org/example/cattool/utills/split/SplitFileToSourceLines.java
package org.example.cattool.utills.split;
import jakarta.persistence.criteria.CriteriaBuilder;
import jakarta.persistence.criteria.CriteriaQuery;
import jakarta.persistence.criteria.Root;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import java.util.Arrays;
import org.example.cattool.database.hibernate.entity.FileEntity;
import org.example.cattool.database.hibernate.entity.LineSourceEntity;
import org.example.cattool.database.hibernate.entity.LineTranslationEntity;
import org.example.cattool.database.hibernate.entity.ProjectEntity;
import org.hibernate.Session;
import static org.example.cattool.database.hibernate.HibernateSession.getSession;
public class SplitFileToSourceLines {
    public static void splitToSourceLines(Integer thisFileId) {
        FileEntity thisFileEntity = new FileEntity();
        thisFileEntity = thisFileEntity.get(thisFileId);
        ProjectEntity thisProjectEntity = new ProjectEntity();
        thisProjectEntity = thisProjectEntity.get(thisFileEntity.getProjectId());
        byte[] thisFileByteContent = thisFileEntity.getContent();
        String thisText = new String(thisFileByteContent, StandardCharsets.UTF_8);
        ArrayList<String> thisLinesList = new ArrayList<>(Arrays.asList(thisText.split("\r?\n")));
        Session thisSession = getSession();
        CriteriaBuilder thisSourceBuilder = thisSession.getCriteriaBuilder();
        CriteriaQuery<LineSourceEntity> thisSourceQuery = thisSourceBuilder.createQuery(LineSourceEntity.class);
        Root<LineSourceEntity> thisSourceRoot = thisSourceQuery.from(LineSourceEntity.class);
        if(thisFileId != null) {
            thisSourceQuery.where(thisSourceBuilder.equal(thisSourceRoot.get("fileId"), thisFileId));
        }
        ArrayList<LineSourceEntity> lineSourceEntities =
            (ArrayList<LineSourceEntity>) thisSession.createQuery(thisSourceQuery).getResultList();
        CriteriaBuilder thisTranslationBuilder = thisSession.getCriteriaBuilder();

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		104

```

CriteriaQuery<LineTranslationEntity> thisTranslationQuery =
thisTranslationBuilder.createQuery(LineTranslationEntity.class);
Root<LineTranslationEntity> thisTranslationRoot = thisTranslationQuery.from(LineTranslationEntity.class);
if(thisFileId != null) {
    thisTranslationQuery.where(thisTranslationBuilder.equal(thisTranslationRoot.get("fileId"), thisFileId));
}
ArrayList<LineTranslationEntity> lineTranslationEntities =
    (ArrayList<LineTranslationEntity>) thisSession.createQuery(thisTranslationQuery).getResultList();
if(lineSourceEntities != null && lineTranslationEntities != null) {
    if(lineSourceEntities.isEmpty() && lineTranslationEntities.isEmpty()) {
        for(int currentIndex = 0; currentIndex < thisLinesList.size(); currentIndex++) {
            LineSourceEntity thisSourceEntity = new LineSourceEntity();
            thisSourceEntity.setContent(thisLinesList.get(currentIndex) == null ? "" : thisLinesList.get(currentIndex));
            thisSourceEntity.setFileId(thisFileId);
            thisSourceEntity.setIndex(currentIndex);
            thisSourceEntity.setSourceLanguageId(thisProjectEntity.getSourceLanguage());
            thisSourceEntity.mergeOrPersist();
            LineTranslationEntity thisTranslationEntity = new LineTranslationEntity();
            thisTranslationEntity.setContent(thisTranslationEntity.getContent() == null ? "" :
thisSourceEntity.getContent());
            thisTranslationEntity.setFileId(thisFileId);
            thisTranslationEntity.setIndex(currentIndex);
            thisTranslationEntity.setTargetLanguageId(thisProjectEntity.getTargetLanguage());
            thisTranslationEntity.mergeOrPersist();
        }
    }
}
}
}
}

```

// File: cat-tool/src/main/resources/hibernate.cfg.xml

<?xml version='1.0' encoding='utf-8'?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name="hibernate.connection.driver_class">org.sqlite.JDBC</property>

<property name="hibernate.connection.url">jdbc:sqlite:sqlite.db</property>

<property name="hibernate.current_session_context_class">thread</property>

<property name="hibernate.dialect">org.hibernate.community.dialect.SQLiteDialect</property>

<property name="hibernate.hbm2ddl.auto">update</property>

<property name="hibernate.show_sql">true</property>

<mapping class="org.example.cattool.database.hibernate.entity.FileEntity"/>

<mapping class="org.example.cattool.database.hibernate.entity.LanguageEntity"/>

<mapping class="org.example.cattool.database.hibernate.entity.LineSourceEntity"/>

<mapping class="org.example.cattool.database.hibernate.entity.LineTranslationEntity"/>

<mapping class="org.example.cattool.database.hibernate.entity.ProjectEntity"/>

</session-factory>

</hibernate-configuration>

// File: cat-tool/src/main/resources/org/example/cattool/gui/javafx/view/AppWelcomeWindowView.fxml

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.layout.VBox?>

<VBox alignment="CENTER"

spacing="15.0"

xmlns="http://javafx.com/javafx"

xmlns:fx="http://javafx.com/fxml"

fx:controller="org.example.cattool.gui.javafx.controller.AppWelcomeWindowController">

<padding>

<Insets bottom="15.0" left="15.0" right="15.0" top="15.0"/>

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		105

```

</padding>
<Label fx:id="thisAppWelcomeText" text="CAT Tool (ver. 1.0)" />
<Button onAction="#thisHelloButtonClickHandler" text="Hello CAT Tool" />
</VBox>
// File: cat-tool/src/main/resources/org/example/cattool/gui/javafx/view/CreateProjectWindowView.fxml
<?xml version="1.0" encoding="UTF-8"?>
<?import com.dlsc.gemsfx.ExpandingTextArea?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.layout.VBox?>
<?import org.controlsfx.control.SearchableComboBox?>
<AnchorPane fx:controller="org.example.cattool.gui.javafx.controller.CreateProjectWindowController"
    xmlns="http://javafx.com/javafx"
    xmlns:fx="http://javafx.com/fxml">
    <StackPane alignment="TOP_LEFT"
        AnchorPane.bottomAnchor="10.0"
        AnchorPane.leftAnchor="10.0"
        AnchorPane.rightAnchor="10.0"
        AnchorPane.topAnchor="10.0">
        <VBox alignment="CENTER_LEFT" spacing="10.0">
            <Label text="This Project" />
            <HBox alignment="CENTER_LEFT" spacing="10.0">
                <Label text="Name:" />
                <TextField fx:id="thisProjectNameField"
                    promptText="Enter project name"
                    HBox.hgrow="ALWAYS" />
            </HBox>
            <HBox alignment="CENTER_LEFT" spacing="10.0">
                <Label text="Description:" />
            </HBox>
            <ExpandingTextArea fx:id="thisProjectDescriptionField"
                promptText="Enter project description"
                VBox.vgrow="ALWAYS"/>
            <Label text="Languages" />
            <HBox alignment="CENTER_LEFT" spacing="10.0">
                <Label text="Source:" />
                <SearchableComboBox fx:id="thisSourceLanguageComboBox"
                    promptText="Select source language"
                    HBox.hgrow="ALWAYS" />
            </HBox>
            <HBox alignment="CENTER_LEFT" spacing="10.0">
                <Label text="Target:" />
                <SearchableComboBox fx:id="thisTargetLanguageComboBox"
                    promptText="Select target language"
                    HBox.hgrow="ALWAYS" />
            </HBox>
            <HBox alignment="CENTER_RIGHT" spacing="10.0">
                <Button text="Resource" onAction="#thisResourceButtonClickHandler"/>
                <Button text="Dictionary" onAction="#thisDictionaryButtonClickHandler"/>
                <Button text="Translator" onAction="#thisTranslatorButtonClickHandler"/>
                <Button text="Export" onAction="#thisExportButtonClickHandler"/>
                <Pane HBox.hgrow="ALWAYS"/>
                <Button onAction="#thisBackButtonClickHandler" text="Back" />
                <Button onAction="#thisSaveButtonClickHandler" text="Save" />
            </HBox>
        </VBox>
    </StackPane>

```

```

</AnchorPane>
// File: cat-tool/src/main/resources/org/example/cattool/gui/javafx/view/EditTextWindowView.fxml
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.layout.VBox?>
<VBox fx:controller="org.example.cattool.gui.javafx.controller.EditTextWindowController"
    xmlns="http://javafx.com/javafx"
    xmlns:fx="http://javafx.com/fxml"
    minWidth="720"
    minHeight="720">
    <TableView fx:id="thisTranslationTable" editable="true" VBox.vgrow="ALWAYS">
        <columns>
            <TableColumn fx:id="thisIndexColumn" text="Index" minWidth="30" prefWidth="45"/>
            <TableColumn fx:id="thisSourceColumn" text="Source"/>
            <TableColumn fx:id="thisTranslationColumn" text="Translation"/>
        </columns>
    </TableView>
</VBox>
// File: cat-tool/src/main/resources/org/example/cattool/gui/javafx/view/ListFilesWindowView.fxml
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.ListView?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<VBox fx:controller="org.example.cattool.gui.javafx.controller.ListFilesWindowController"
    xmlns="http://javafx.com/javafx"
    xmlns:fx="http://javafx.com/fxml"
    onDragOver="#thisHandleDragOverEvent"
    onDragDropped="#thisHandleDragDroppedEvent">
    <Label text="Resource"/>
    <ListView fx:id="thisFileListView" VBox.vgrow="ALWAYS" onMouseClicked="#thisListViewClickHandler"/>
    <HBox alignment="CENTER_RIGHT" spacing="10.0" style="-fx-padding: 10px;">
        <Button text="Add Files" onAction="#thisAddFilesButtonClickHandler"/>
        <Button text="Delete Selected" onAction="#thisDeleteFilesButtonClickHandler"/>
    </HBox>
</VBox>
// File: cat-tool/src/main/resources/org/example/cattool/gui/javafx/view/ListProjectsWindowView.fxml
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<BorderPane
    fx:controller="org.example.cattool.gui.javafx.controller.ListProjectsWindowController"
    xmlns="http://javafx.com/javafx"
    xmlns:fx="http://javafx.com/fxml">
    <top>
        <VBox spacing="10" style="-fx-padding: 10px;">
            <Label text="CAT Tool"/>
            <TextField fx:id="thisSearchField" promptText="Search projects" onKeyReleased="#thisSearchFieldKeyHandler"
        </VBox>
    </top>
    <center>
        <ListView fx:id="thisProjectFilterView" prefWidth="300" onMouseClicked="#thisProjectFilterClickHandler"/>
    </center>
    <bottom>
        <HBox spacing="10" alignment="CENTER_RIGHT" style="-fx-padding: 10px;">
            <Button text="Settings" onAction="#thisSettingsButtonClickHandler" />
            <Region HBox.hgrow="ALWAYS" />
            <Button text="Delete" onAction="#thisDeleteProjectButtonClickHandler"/>
            <Button text="New Project" onAction="#thisCreateNewProjectButtonClickHandler"/>
        </HBox>
    </bottom>
</BorderPane>

```

```

        <Button text="Open" onAction="#thisOpenProjectButtonClickHandler"/>
    </HBox>
</bottom>
</BorderPane>
// File: cat-tool/src/main/resources/org/example/cattool/gui/javafx/view/TranslateTextWindowView.fxml
<?xml version="1.0" encoding="UTF-8"?>
<?import com.dlsc.gemsfx.ExpandingTextArea?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ChoiceBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.layout.VBox?>
<AnchorPane fx:controller="org.example.cattool.gui.javafx.controller.TranslateTextWindowController"
    xmlns="http://javafx.com/javafx"
    xmlns:fx="http://javafx.com/fxml"
    minHeight="720"
    minWidth="480">
    <StackPane
        AnchorPane.bottomAnchor="10.0"
        AnchorPane.leftAnchor="10.0"
        AnchorPane.rightAnchor="10.0"
        AnchorPane.topAnchor="10.0">
        <VBox spacing="10.0">
            <Label text="Translate"/>
            <HBox spacing="10.0">
                <Label text="Select Translator:"/>
                <ChoiceBox fx:id="thisTranslatorChoiceBox"/>
            </HBox>
            <HBox spacing="10.0">
                <Label text="Language: "/>
                <Label text="Source "/>
                <ChoiceBox fx:id="thisSourceLanguageChoiceBox"/>
                <Label text="Target"/>
                <ChoiceBox fx:id="thisTargetLanguageChoiceBox"/>
            </HBox>
            <Label text="Source:"/>
            <ExpandingTextArea fx:id="thisSourceText"/>
            <Label text="Translation:"/>
            <ExpandingTextArea fx:id="thisTranslatedText"/>
            <Button text="Query" onAction="#thisQueryTranslationButtonClickHandler"/>
            <Button fx:id="thisUseText" text="Use" onAction="#thisUseTextButtonClickHandler"/>
        </VBox>
    </StackPane>
</AnchorPane>
// File: cat-tool/src/main/resources/org/example/cattool/gui/javafx/view/ViewDictionaryWindowView.fxml
<?xml version="1.0" encoding="UTF-8"?>
<?import com.dlsc.gemsfx.ExpandingTextArea?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.CheckBox?>
<?import javafx.scene.control.ComboBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.VBox?>
<AnchorPane fx:controller="org.example.cattool.gui.javafx.controller.ViewDictionaryWindowController"
    xmlns="http://javafx.com/javafx"
    xmlns:fx="http://javafx.com/fxml"
    minHeight="150.0" minWidth="750.0">
    <VBox alignment="CENTER_LEFT" spacing="10.0"

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						108
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    AnchorPane.bottomAnchor="10.0"
    AnchorPane.leftAnchor="10.0"
    AnchorPane.rightAnchor="10.0"
    AnchorPane.topAnchor="10.0">
    <CheckBox fx:id="thisOpenInBrowserCheckbox" text="Open in Browser"/>
    <HBox alignment="CENTER_LEFT" spacing="10.0">
        <Label text="Copied Element:"/>
        <ExpandingTextArea fx:id="thisCopiedElementField" promptText="Enter copied element"
HBox.hgrow="ALWAYS"/>
    </HBox>
    <HBox alignment="CENTER_LEFT" spacing="10.0">
        <Label text="Use Dictionary:"/>
        <ComboBox fx:id="thisUseDictionaryComboBox" promptText="Select dictionary"/>
    </HBox>
    <HBox alignment="CENTER_RIGHT" spacing="10.0">
        <Button onAction="#thisSearchButtonClickHandler" text="Search"/>
        <Button fx:id="thisClipboardButton" onAction="#thisClipboardButtonClickHandler" text="Clipboard"/>
        <Pane HBox.hgrow="ALWAYS"/>
    </HBox>
</VBox>
</AnchorPane>

```