

Пермский национальный исследовательский политехнический университет
Кафедра информационных технологий и автоматизированные системы

Творческая работа
“Автоматизация научного перевода”

Выполнил:

студент группы: ИВТ-24-16

Е.А.Попов

Проверил:

доцент кафедры ИТАС

О.А. Полякова

Пермь 2025

Цель проекта

Создать приложение с графическим интерфейсом, которое позволяет пользователю:

- Авторизоваться или зарегистрироваться.
- Вводить текст на английском языке.
- Получать перевод текста на русский язык и объяснения терминов с использованием внешних Python-скриптов.

Описание проекта

Проект представляет собой приложение, разработанное с использованием фреймворка Qt (C++). Приложение имеет интуитивно понятный интерфейс, включающий приветственное окно с изображением котика, окно входа/регистрации и основное окно для перевода текста. Пользователь может зарегистрироваться, войти в систему, ввести текст на английском, а затем получить перевод на русский и объяснения терминов, выполненные с помощью внешних скриптов `translator.py` и `explain_terms.py`.

Структура проекта

1. **WelcomeWindow:**

- Приветственное окно с изображением котика (cat.jpg) в качестве фона.
- Кнопка "Welcome" для перехода к окну входа.

2. **LoginWindow:**

- Окно входа с полями для ввода логина и пароля.
- Кнопки "Вход" и "Регистрация".
- Данные пользователей хранятся в файле users.json.

3. **RegisterWindow:**

- Окно регистрации с полями для ввода логина и пароля.
- Кнопка "Зарегистрировать" для сохранения нового пользователя.

4. **MainWindow:**

- Основное окно с текстовыми полями для ввода текста и отображения результата.
- Кнопки "Перевести и объяснить" (выполняет перевод и анализ) и "Очистить" (очищает поля).

Использованные технологии

- **Язык программирования:** C++ (для интерфейса и логики приложения).
- **Фреймворк:** Qt (для создания графического интерфейса).
- **Python-скрипты:**
 - translator.py: перевод текста с английского на русский.
 - explain_terms.py: анализ и объяснение терминов.
- **Формат данных:** JSON (для хранения пользователей в users.json).
- **Дополнительные библиотеки Qt:**
 - QProcess: для выполнения внешних Python-скриптов.
 - QCryptographicHash: для хеширования паролей.
 - QFile, QJsonDocument, QJsonObject: для работы с файлами и JSON

Библиотеки, API и другие инструменты, которые были использованы автором

- **Библиотеки:**

- **Qt Core и Qt Widgets:** основа для создания графического интерфейса и работы с процессами (QProcess), файлами (QFile) и JSON (QJsonDocument, QJsonObject).
- **QCryptographicHash:** встроенная библиотека Qt для хеширования паролей с использованием алгоритма SHA-256.

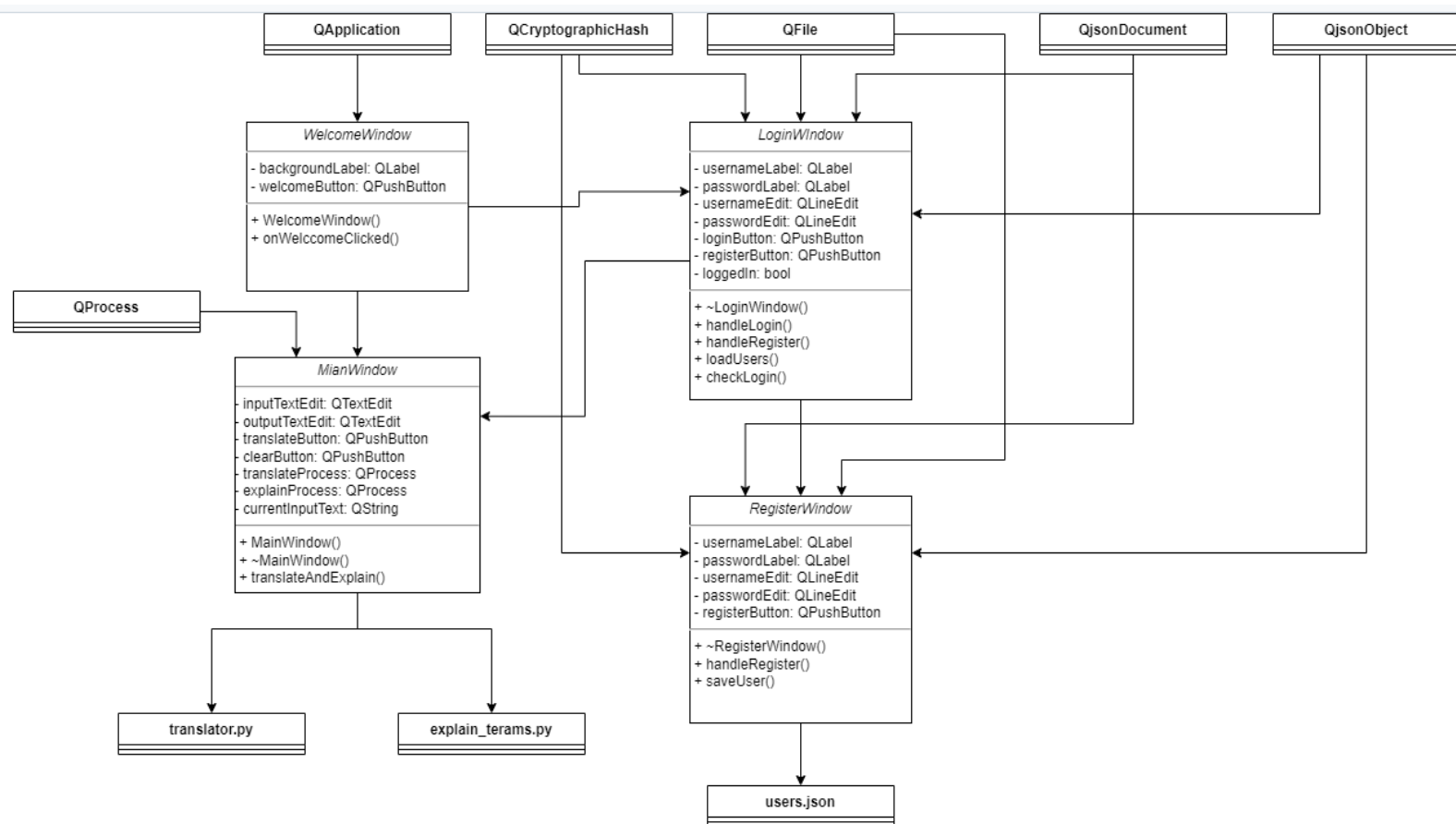
- **API и внешние инструменты:**

- Нет использования сторонних API (например, REST API), так как перевод и анализ реализуются через локальные Python-скрипты (translator.py и explain_terms.py).

- **Инструменты разработки:**

- **Qt Creator:** среда разработки для написания и компиляции кода на C++.
- **Python:** язык для написания скриптов translator.py и explain_terms.py, выполненных с использованием библиотек (например, translate или кастомных модулей, если они есть).

UML-диаграмма



Основные классы:

1. WelcomeWindow

- От WelcomeWindow → LoginWindow: создаёт и вызывает окно входа.
- От WelcomeWindow → MainWindow: открывает основное окно при успешном входе.

2. LoginWindow

- От LoginWindow → RegisterWindow: создаёт и вызывает окно регистрации.
- От LoginWindow → MainWindow: проверяет доступ и передаёт управление основному окну.

3. RegisterWindow

- От RegisterWindow → users.json: сохраняет данные пользователей (стрелка пунктирная, так как это запись в файл).

4. MainWindow

- От MainWindow → QProcess: использует для выполнения внешних процессов.
- От MainWindow → translator.py: вызывает скрипт для перевода.
- От MainWindow → explain_terms.py: вызывает скрипт для объяснения терминов.

Дополнительные классы:

1. QApplication

- От QApplication → WelcomeWindow: управляет запуском приложения и первым окном.

2. QFile

- От QFile → LoginWindow: используется для чтения данных из файла.
- От QFile → RegisterWindow: используется для записи данных в файл.

3. QJsonDocument

- От QJsonDocument → LoginWindow: используется для парсинга JSON данных.
- От QJsonDocument → RegisterWindow: используется для записи JSON данных.

4. QJsonObject

- От QJsonObject → LoginWindow: используется для работы с объектами JSON.
- От QJsonObject → RegisterWindow: используется для работы с объектами JSON.

5. QCryptographicHash

- От QCryptographicHash → LoginWindow: используется для хеширования паролей.
- От QCryptographicHash → RegisterWindow: используется для хеширования паролей.

6. QProcess

- От QProcess → MainWindow: предоставляет функциональность для выполнения процессов.

Вывод

Проект "Автоматизация научного перевода" успешно реализован. Программа позволяет пользователям регистрироваться, входить в систему и переводить текст с английского на русский с объяснением терминов. Интерфейс интуитивно понятен, дизайн современный, а архитектура приложения задокументирована через UML-диаграмму. В будущем можно добавить дополнительные функции, такие как поддержка других языков или улучшение алгоритмов перевода.