**桌面CAT系统需求分析**

根据现实实际情况进行用户故事的假想：

当用户进入应用界面时，首先会看到一个简洁直观的界面，其中在左侧有一个用于输入源语言文本或者导入目标文档的文本框以及右侧一个显示翻译结果的区域，在右侧还会有一个显示翻译记忆库的区域，在遇到重复文本时可以快速得到翻译结果。

用户输入源语言文本后，点击界面上的翻译键，系统会显示翻译后的文本，帮助用户进行理解。用户也可以点击选择或修改翻译，并选择保存或导出最终的翻译结果。同时，用户可以进入翻译记忆库和术语库的管理界面，进行创建、编辑或者搜索翻译记忆库和术语库。用户还可以选择导入文档的功能，选择需要翻译的文件并限定在指定导入的格式内，即可导入进行翻译。翻译完成后，用户可以选择导出翻译结果，并可以选择需要的文件格式进行导出。

用户还可以管理翻译项目，创建新项目、添加文件到项目、分配翻译任务，并监控项目的进度。比如，当翻译多面文档内容时，机器翻译会有一个过程，可以显示当前已经翻译到文档的哪一个部分。另外，用户可以进入文本分段规则设置界面，选择或输入自定义分段规则，系统会根据用户的选择或输入，将长文本自动分段成合适的翻译单元，提高翻译效率。

最后，用户可以打开集成外部翻译服务界面，选择并配置需要的翻译服务，将文本发送到选择的翻译服务进行自动翻译，并在界面上显示翻译结果。用户体验到界面设计简洁直观，元素布局合理，快速找到需要的功能并进行操作，同时系统适配不同大小和分辨率的屏幕，确保在各种设备上都能正常使用。

**根据假想以及框架，将系统所要实现的功能描述为以下系统需求：**

**1.辅助翻译界面和功能**

1.1 系统将提供直观易用的用户界面，其中包括的有输入源语言文本的输入框和显示翻译结果的区域，以及显示翻译记忆库或者术语库的区域，以便用户查看。

如下图，参考OmegaT的用户界面，左侧用于导入或者输入源语言文本，上侧的文件可以选择导入文档。右侧上方的区域即是显示翻译结果的区域，点击翻译即可翻译当前选定的语句。

输入框和显示区域：实现一个用户界面（UI）框架，使用WinForms来设计用户界面。界面应包括文本输入框、翻译结果显示区域，以及翻译记忆库和术语库的显示区域。使用TextBox控件来输入源语言文本。

使用ListBox或DataGrid控件来显示翻译记忆库和术语库。

使用RichTextBox控件来显示翻译结果。

1.2 系统将实现实时翻译建议的显示，并允许用户通过在翻译区域进行单词的点击或类似方式选择或修改翻译。

翻译建议生成：在用户输入文本时，系统调用翻译引擎，生成翻译建议并显示在界面上。可以使用事件处理机制，如TextChanged事件，来捕捉用户的输入，并实时调用翻译API。

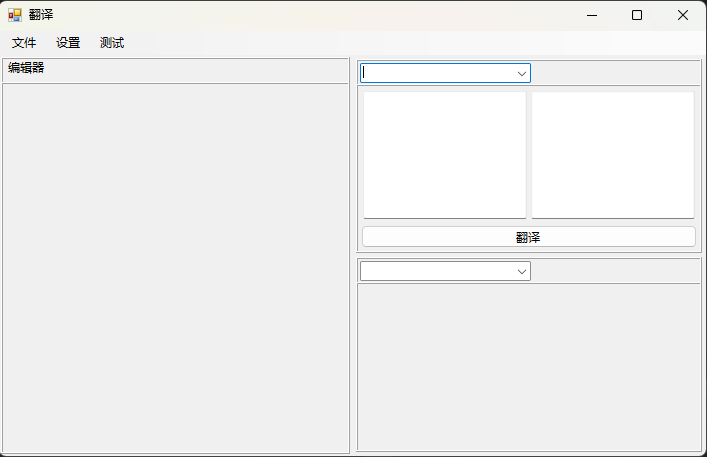
使用比如“HttpClient”调用外部翻译服务，如百度翻译API。

在获取翻译结果后，更新显示区域。

1.3 系统应提供保存和导出翻译结果的功能，支持常见的文件格式，如Word文档（.docx）、文本文件（.txt）等。

文件导入导出：实现文件选择对话框，允许用户导入源语言文档，并将翻译结果保存为不同格式的文件。可以使用OpenFileDialog和SaveFileDialog来选择文件和保存文件。导入文件时，读取文件内容并显示在输入框中。

导出文件时，将翻译结果保存为指定格式的文件（如.docx、.txt）。



**2.翻译记忆库管理和辅助翻译的使用**

2.1 在翻译区域，系统的设计需支持创建、导入和编辑翻译记忆库，包括自主添加、修改和删除翻译条目。

使用SQLite数据库存储翻译记忆库。

提供用户界面和文件对话框（OpenFileDialog）来导入翻译记忆库文件（如 TMX文件）。

创建数据库表结构，并在导入文件时解析文件内容，插入数据库。

2.2 系统将实现在翻译过程中自动检索先前所存储的翻译记忆库，并提供与之匹配的文本翻译。

在用户输入文本时，通过TextBox的TextChanged事件触发自动检索。

使用SQLite的查询功能，查找与输入文本匹配的翻译条目。

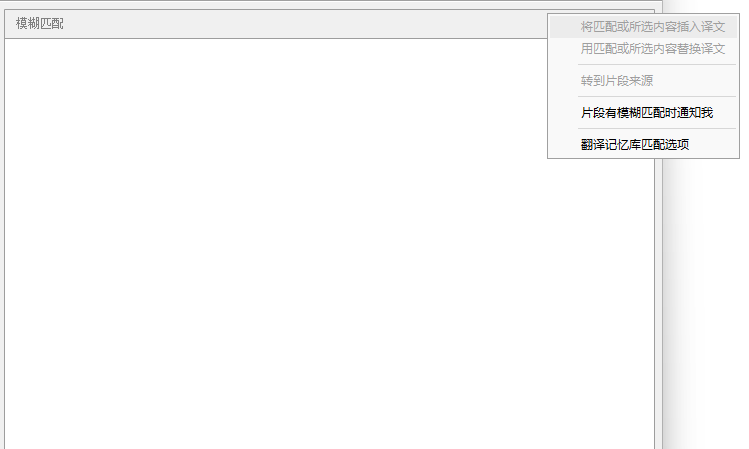
显示匹配的翻译结果在RichTextBox控件中。

2.3 系统应该提供搜索功能，允许用户通过关键词或一些条件来进行检索翻译记忆库。

提供搜索框和按钮，允许用户输入关键词进行搜索。

使用SQLite的查询功能，根据关键词在翻译记忆库中查找匹配的条目。

将搜索结果显示在DataGrid控件中，供用户查看和选择。





参考OmegaT翻译记忆库匹配中的功能进行取舍

1. **术语库管理和辅助翻译中使用**

术语库的制作应该参考翻译记忆库的功能进行设计

3.1 支持创建、导入和编辑术语库，其中包含的有添加、修改和删除术语条目。

3.2 系统提供搜索功能，允许用户通过关键词或一些条件来进行检索术语库。通过模糊匹配，即使搜索的关键词有部分出入或者英语大小写不同，也可以显示出结果，在无法检索到相关词汇会也会返回错误提示。

3.2.1 匹配查看器中显示了翻译记忆库中最相似的片段，来源既包括随着您对项目的翻译而实时创建的内部项目翻译记忆库，也包括来自于您的过往工作或从客户或翻译代理那里得到的辅助翻译记忆库。

3.2.2 模糊匹配的实现可以通过多种方法来实现，其中一种常见的方法是使用字符串匹配算法，如编辑距离算法。

编辑距离算法是一种用于衡量两个字符串之间相似程度的算法，它衡量的是从一个字符串转换到另一个字符串所需的最小操作次数，包括插入、删除和替换字符。基于编辑距离算法，可以计算出查询字符串与数据库中每个条目的相似程度，然后根据相似程度来排序和选择匹配的条目。

除了编辑距离算法，还有其他一些模糊匹配的方法，如基于相似度的搜索算法（如余弦相似度、Jaccard相似度等）、基于模式匹配的算法（如通配符匹配、正则表达式匹配等）等。这些算法可以根据具体的应用场景和需求来选择和实现。

提供界面供用户输入术语及其释义，然后由系统进行格式化，或者用户按照规定格式导入术语库，以保证术语库使用正常。

**4.多格式导入和导出**

4.1 系统应在翻译时同时支持多种常见文件格式的导入，包括Word文档、txt文件等。

使用DocX库处理Word文档，使用System.IO处理文本文件。

提供文件对话框（OpenFileDialog）让用户选择导入的文件，根据文件扩展名 决定处理方式。

示例代码展示了如何读取并显示文件内容。

4.2 系统应提供选择导入文件和导出格式的界面，并允许用户自由选择方便的界面设计。

提供导入和导出文件的用户界面，通过菜单项或按钮触发文件对话框。

使用OpenFileDialog和SaveFileDialog实现文件选择。

导入文件时，读取文件内容并显示在文本框中；导出文件时，将文本框内容保存为用户选择的格式。

4.3 系统应实现将翻译结果导出为用户选择的文件格式，同时保持格式的兼容性和准确性。

导出文件时，根据用户选择的格式创建并保存文件。

使用Aspose.Words库生成Word文档，使用System.IO写入文本文件。

确保导出的文件内容与用户输入的文本一致，保持文本的准确性和完整性。

**5.翻译项目管理**

5.1 系统应提供创建新项目、添加文件到项目和显示翻译内容的功能。

提供UI界面，让用户输入项目名称和描述，并创建新项目。在本地或数据库中生成唯一的项目ID。

使用文件对话框（OpenFileDialog）让用户选择文件，并将选中的文件与项目ID关联。

读取选中文件的内容并显示在翻译区域中。

5.2 系统应提供项目列表和详细信息的界面，以便用户查看和管理项目。

提供列表控件显示所有项目，包括项目名称、创建日期和状态。用户可以选 择项目查看详细信息。

详细信息界面显示项目名称、描述、创建日期和关联文件列表，并允许用户 编辑项目或添加文件。

5.3 系统应实现项目状态的实时更新，并提供提醒和通知功能，以便用户及时了解项目进展情况。

实现定时刷新或基于事件的机制，实时更新项目状态。使用任务调度或事件 驱动编程定时检查项目状态。

在UI中提供通知区域，当项目状态有变化时，及时提示用户。可以使用消 息框、状态栏或弹出通知。

**6.文本分段**

6.1 系统后续可以提供允许用户选择或自定义分段规则，如按段落、句子或标点符号分段。

6.2 系统后续应实现将长文本自动分段成合适的翻译单元，同时保持段落的逻辑完整性和语义一致性，不影响文章表达信息。

**7.翻译服务集成**

**7.1** 系统应支持集成外部翻译服务，如机器翻译API，并提供接口供用户选择和配置。

7.1.1 接口描述及接入方式：使用百度翻译API，通过 HTTP 接口对外提供多语 种互译服务。

7.1.2 接口功能：通过调用通用翻译API，传入待翻译的内容，并指定要翻译的 源语言和目标语言种类，就可以得到相应的翻译结果。

7.1.3 通信协议：使用通用翻译API HTTPS地址： https://fanyi-api.baidu.com/api/trans/vip/translate

7.1.4数据格式：与百度翻译API进行通信时使用的数据格式为json。

7.1.5 翻译结果生成：

Step1. 将请求参数中的 APPID(appid)， 翻译 query(q)，随机数(salt)，以及平台分配的密钥，按照 appid+q+salt+密钥的顺序拼接得到字符串 1。

Step2. 对字符串 1 做 MD5 ，得到 32 位小写的 sign。

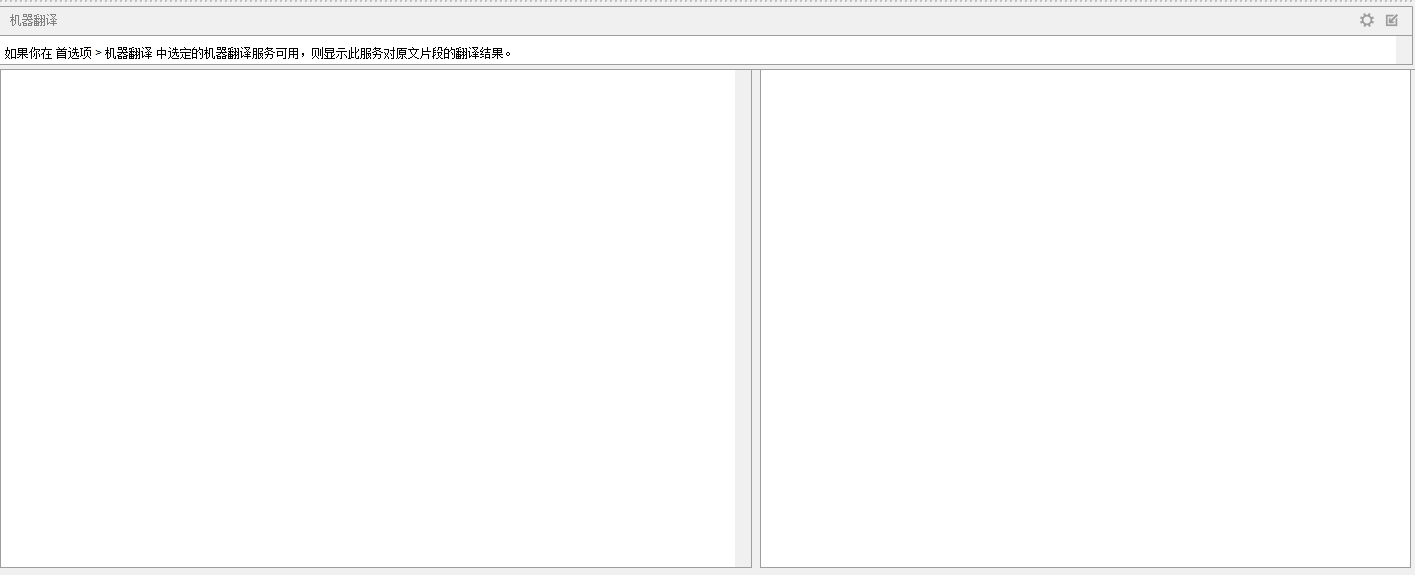
注：

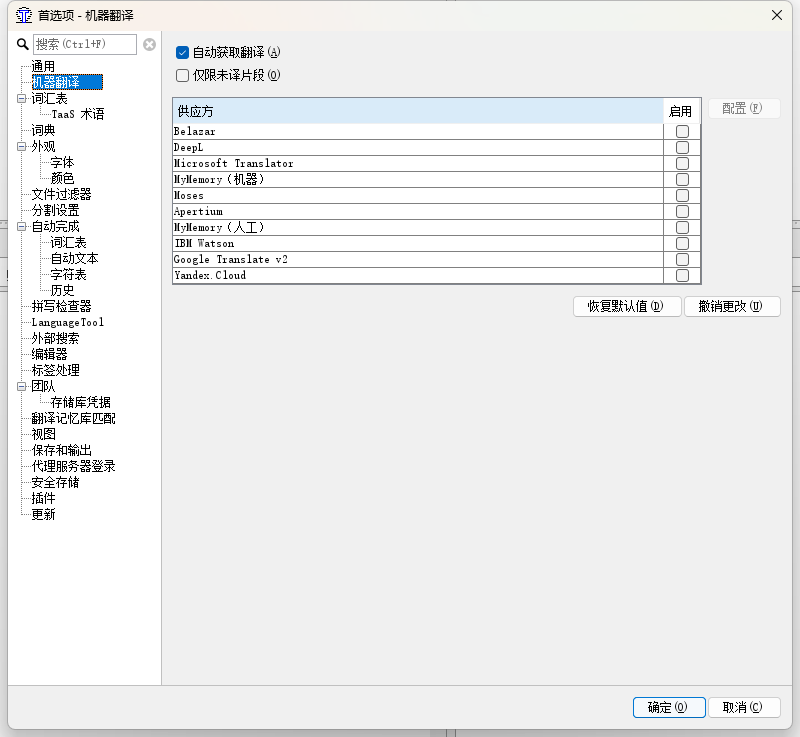
1. 待翻译文本（q）参数为 UTF-8 编码；

2. 在生成签名后，发送 HTTP 请求时，将 query 拼接在URL上，并对 query 做 URL encode。

7.1.6 安全性要求：为保证调用，使用 MD5 算法生成的一段字符串作为签 名，生成的签名长度为 32 位，签名中的英文字符均为小写格式。

7.2 系统应实现将文本发送到选择的翻译服务进行自动翻译，并在翻译区域自动显示出翻译结果。

7.3 系统应确保集成服务的稳定性和性能，当代码出现错误出现无法翻译之类的问题时应提供异常处理和错误提示功能。 



**8.性能需求**

8.1设计后的系统应能够在处理大规模文本时保持良好的响应速度，文本翻译应在用户输入确认后即时显示。

8.2系统的翻译需要具有高准确度和覆盖率，以提高翻译效率。

**9.用户界面需求**

9.1 界面应设计简洁直观，易于使用，以提高用户体验。目前的界面设计参考OmegaT进行设计，如下图。

9.2 界面元素应适配不同大小和分辨率的屏幕，以确保在各种设备上的可用性。

