

# 项目概述

---

## 项目名称

基于openai api的远程语音对话系统。

## 项目背景和目标

部分中老年人缺乏情感陪伴和健康关照，或需要聊天说话的对象，或需要解决一些不愿寻求他人帮助的健康问题。老人常因为不愿麻烦子女、不信任医生、有些症状难以启齿等某些原因对症状隐瞒或拖延，但内心十分希望可以解决疑惑，比有些人出现盲目相信网上错误回答的情况。

本产品希望解决老人的以下两个需求：

- 独居老人需要有人对话、情感陪伴。
- 解答对某些病症的疑惑，并适时提供建议和帮助。

# 需求概述

---

## 功能性需求

### 用户交互

用户只拿到一个硬件产品，上有开关和一个互动按钮。用户开机后，需要将设备接入家中的wifi。连接成功后按下按钮之后可以开始说话，说完后松开按钮，进入等待，系统响应后自动播放语音回答。全部对话完毕后将设备关机。

### 系统模块

语音识别和语音合成模块（硬件）、硬件设备的联网及收音播放模块、数据传输模块、并发处理模块、数据存储模块、智能对话模块。

若嵌入式语音识别和合成模块表现不符合预期，或将转向在服务器进行语音处理，到时将把语音识别和语音合成模块（硬件）拆分成语音识别和语音合成模块与硬件部分。

### 业务流程

每次开机时自动连接家中互联网，硬件设备监听按键状态，触发时麦克风开始录音，停止后结束录音，在下次扬声器播放完回答语句后按键不会再生效。录音后由硬件的语音识别模块本地处理，打包后通过socket向服务器发送请求。

服务器在接收到请求后创建新的进程，自动在数据库根据设备id查询相关用户记忆，结合查询结果生成给gpt的，结合了用户记忆的提问语句，调用OpenAI API并等待回答结果。得到回答结果后自动更新数据库中的记忆。若对话达到一定次数，再次调用OpenAI API请求帮助总结新的用户记忆并更新至数据库。服务器将回答语句处理后通过socket返回至硬件设备，进程结束。

设备接收到服务器返回信息后由硬件的语音合成模块本地处理，通过扬声器播放，并再次进入监听状态，直至关机。

若硬件在15秒内未能收到回应，应该停止接收，并且提示用户重新说。若无网络连接，应该提示用户网络连接状态异常。

## 非功能性需求

- 响应时间：从用户完成讲话到播放回复应在三秒内。但后续要参考OpenAI API的实际响应速度。
- 并发数：对于目前的原型阶段，至少应支持1000人的同时访问。
- 可用性：系统可用时间应达到99%以上。
- 可扩展性：尽量做到模块划分清晰、接口简洁易懂，后续若要扩展功能，比如改为实时通话易于修改，且不影响已有业务。
- 用户友好性：对老年用户友好，操作极为简洁，学习门槛极低。

## 用户场景

主要用户是中老年人，使用场景是在没有子女或其他年轻人在场陪伴时，在有wifi的家中使用。第一次联网成功后，后续自动进行网络连接，用户只需执行最简单的开关机、按键说话、等待响应即可。

# 技术架构（详见架构设计文档与技术设计文档）

## 技术架构说明

模块具体功能介绍：

- 客户端：
  - 语音识别和语音合成模块（硬件）：
  - 硬件设备的联网及收音播放模块
- 服务器端：
  - 数据传输模块
  - 并发处理模块
  - 数据存储模块
  - 智能对话模块

备注：

- 若集成式语音处理硬件性能不达标，语音识别和语音合成模块（硬件）会拆分成两个模块在服务器中实现
- 后续考虑扩展Redis缓存来缓解数据库访问负担

## 开发环境

- 开发系统：macOS、Windows
- 开发语言：python、C
- 依赖库：websocket、OpenAI API、pthread
- 数据库：mysql
- 若更改技术路线：Google Cloud Text-to-Speech API（未选定）、Google Cloud Speech-to-Text API（未选定）

## 接口与集成简述

- 语音识别和语音合成功能集成在硬件设备中进行本地处理。

- 客户端通过socket与服务器实时通信。
- 服务器通过调用OpenAI API生成智能对话答案

## 环境要求

---

硬件环境：暂无

软件环境：参考上一条开发环境。主要使用的IDE为vs code

## 接口设计

---

### 接口

- 客户端与服务器的通信接口：使用websocket进行通信，json格式传输消息。
- 智能对话模块接口：调用OpenAI API
  1. 请求为经过语音识别、结合历史数据预处理后的文本，响应为智能对话系统生成的文本回复
  2. 请求为用户历史数据，响应为更新后的用户记忆
- 数据库接口：
  1. 查询用户历史记忆
  2. 更新用户历史记忆

## 扩展

---

潜在的扩展需求包括：

- 转向在服务器处理语音文本互换的技术路线，可能调用google或其他相似产品API。
- 加入内存缓存Redis，减轻数据库压力，提升web应用响应速度。