



温州肯恩大学
WENZHOU-KEAN UNIVERSITY

CPS 1032 Hackathon
AIGC Product For Seniors

AI Assistant for Seniors with Alzheimer's
Disease

Student : Shang Songlin shangs@kean.edu 1308184

Professor: Tiffany Tang

Date: 11/24/2024-11/25/2024

Content

1. Project instruction	3
1.1 Target population introduction	3
1.2 Project Aim	3
2. Project Design Idea	3
2.1. Preventing the Elderly from Getting Lost	3
2. 2 Sending Location Updates to Family Members	4
2.3 Voice Interaction Functionality	5
3. Code	5
3.1 The associated code for the location record	5
3.1.1 Explanation And Progress:	5
3.2 Emergency call function code	12
3.3 Voice interaction features Code	12

In this group project, each team is expected to use AIGC to create any "future" product for the elderly. It can be a piece of music/art work, a special cup, a social space to enable trans-generation communication and entertainment, a senior-version of an app which can better serve them, mini games, etc.

1. Project instruction

1.1 Target population introduction

Elderly individuals with Alzheimer's disease experience significant memory decline due to the degeneration of neurons in the brain, leading to numerous inconveniences and even dangers in daily life. In the early stages of Alzheimer's disease, patients may momentarily forget what they intend to do or where they have placed certain items. During the middle stages, patients may lose the ability to recall the way home or become unaware of their current activities. In the later stages, patients often lose the ability to think and reason entirely.

1.2 Project Aim

The AI assistant I aim to develop for individuals with Alzheimer's disease is designed to significantly reduce safety risks for this demographic. While its effectiveness may be limited for patients in the advanced stages of the disease, it can provide a certain degree of safety and security for those in the early to middle stages. This solution seeks to enhance the quality of life and mitigate potential hazards for Alzheimer's patients during the earlier phases of the condition.

Especially for the elderly with Alzheimer's disease who are lost outside alone, it is the case that we mainly help.

2. Project Design Idea

2.1. Preventing the Elderly from Getting Lost

The AI assistant is designed to continuously track the real-time location of the elderly and record their movements using a linked list data structure for efficient storage. It will periodically check in with the user through prompts to determine if assistance is needed. If the system detects that the individual is lost, it will initiate navigation based on pre-configured home address data to guide the elderly back to their residence.

code design think:

```
If ChatModel_Receive_Help==True
```

```
    return Navigation functions;
```

```
Func(Navigation functions):
```

```
    location[0] ---> location[1]--->.....--->location[n];
```

```
return location[0]
```

```
try:
    directions = gmaps.directions(
        origin=current_location,
        destination=HOME_ADDRESS,
        mode="walking"
    )
    steps = directions[0]['legs'][0]['steps']
    route = [step['html_instructions'] for step in steps]
    return route
except Exception as e:
    print(f"导航失败: {e}")
    return []
```

2. 2 Sending Location Updates to Family Members

When the AI detects that the elderly user is lost, it will simultaneously, it will assist the elderly in navigating home using the built-in navigation system. And Notify pre-registered family contacts by sending the elderly user's current location and a message indicating that they may be lost. This ensures that family members are informed in real-time and can intervene if necessary.

code design think:

Hash Table to store phone number:

```
phone_book = {
    "123-456-7890": "Alice",
    "987-654-3210": "Bob",
    "456-789-1234": "Charlie"
}
for phone, name in phone_book.items():
    print(f"Phone: {phone}, Name: {name}")
```

If ChatModel_Receive_Help:

```
return emergency_calls
Func(emergency_calls):
```

```
// Replace with a phone number
```

```
String phoneNumber = "1234567890";
```

```
// Create dialing Intent
```

```
Intent intent = new Intent(Intent.ACTION_DIAL);
intent.setData(Uri.parse("tel:" + phoneNumber));
```

```
// Start the dialing screen
```

```
startActivity(intent);
```

2.3 Voice Interaction Functionality

The project aims to integrate advanced voice interaction capabilities by leveraging APIs such as OpenAI's ChatGPT voice models or Doubao APIs. These voice interfaces will allow users to interact with the AI assistant effortlessly, enabling the execution of functionalities such as navigation assistance or safety checks through simple voice commands. This ensures an intuitive and user-friendly experience for elderly individuals who may have difficulty operating traditional digital interfaces.

3. Code

3.1 The associated code for the location record

3.1.1 Explanation And Progress:

For address lookup, my initial plan was to use the Google Maps API for location tracking. However, Google Maps lacks permission to operate in mainland China. As an alternative, I decided to use the Gaode (Amap) API. As shown in the **Figures 3.1**, the Gaode API can successfully locate my position in Wenzhou, Zhejiang Province. However, when attempting to obtain more detailed information, significant discrepancies arise. For instance, the location is identified in Lucheng District but with an approximate offset of 5 kilometers. This suggests that positioning via public internet services can result in substantial errors.

Subsequently, I attempted to retrieve GPS latitude and longitude coordinates directly from the device's network connection. To achieve this, I used ChatGPT to generate an HTML code capable of acquiring the current GPS data. While the code appears functional, the webpage fails to retrieve the data. This might be due to device permission restrictions, which are challenging to resolve without access to the appropriate hardware.

Given this limitation, I am considering purchasing GPS-related positioning chips in the future to enable precise positioning functionality. This hardware-based approach may overcome the inaccuracies of public internet APIs and the device permission issues encountered when acquiring GPS data directly.

```

# 位置记录模块
location_history = []

def record_location():
    """
    每隔一段时间记录位置信息。
    """
    try:
        # 获取设备的当前位置（示例为模拟位置）
        location = gmaps.geolocate() # 实际需要设备支持
        lat, lng = location["location"]["lat"], location["location"]["lng"]
        timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        location_history.append({"latitude": lat, "longitude": lng, "timestamp": timestamp})
        print(f"[位置记录]: 经度 {lng}, 纬度 {lat} 时间: {timestamp}")
    except Exception as e:
        print(f"位置记录失败: {e}")

```

✓ 0.0s

Python

```

# 导航模块
def get_directions_to_home(current_location):
    """
    根据当前位置规划回家的路线。
    """
    try:
        directions = gmaps.directions(
            origin=current_location,
            destination=HOME_ADDRESS,
            mode="walking"
        )
        steps = directions[0]['legs'][0]['steps']
        route = [step['html_instructions'] for step in steps]
        return route
    except Exception as e:
        print(f"导航失败: {e}")
        return []

def navigate_home():
    """
    获取当前位置并规划导航回家的路径。
    """
    current_location = "your_current_lat_lng" # 这里需要通过 GPS 获取当前位置
    print("[导航中]...")
    route = get_directions_to_home(current_location)
    for step in route:
        # 播报每一步导航信息
        text_to_speech(step)
        time.sleep(5) # 模拟步行时间

```

✓ 0.0s

Python

```
import requests

def get_location_from_ip(AMAP_API_KEY):
    """
    调用高德地图 IP 定位 API 获取当前地址
    """
    url = f"https://restapi.amap.com/v3/ip"
    params = {
        'key': AMAP_API_KEY # 高德地图 API 密钥
    }
    response = requests.get(url, params=params)
    if response.status_code == 200:
        data = response.json()
        if data['status'] == '1': # 确认请求成功
            # 提取返回的地址信息
            province = data.get('province', '未知省份')
            city = data.get('city', '未知城市')
            adcode = data.get('adcode', '未知行政区划代码')
            return {
                'province': province,
                'city': city,
                'adcode': adcode
            }
    return None

# 示例
AMAP_API_KEY = "1eac4b127c4bedfe50b5ae79fdc9c12e" # 替换为你的高德地图 API 密钥
location_info = get_location_from_ip(AMAP_API_KEY)

if location_info:
    print(f"当前地址信息: {location_info['province']} {location_info['city']} (行政区划代码: {location_info['adcode']})")
else:
    print("无法获取当前地址信息")
```

✓ 0.1s Python

当前地址信息: 浙江省 温州市 (行政区划代码: 330300)

当前地址信息: 浙江省 温州市 (行政区划代码: 330300)

Figure3.1: Code Result For Location1

AIForAlzheimer 2024/11/23 创建 编辑 添加Key 删除

Key 名称	Key <small>商用说明</small>	安全密钥 (点击查看安全密钥使用说明)	绑定服务	操作 <small>⌵</small>
Ip2	ffcdcade84032fd4ebbc9893efc32ef4	—	Web服务	设置 查看配额 删除
IP	1eac4b127c4bedfe50b5ae79fdc9c12e	—	Web服务	设置 查看配额 删除

Figure3.2:Figure For Gaode API Key

* 服务平台: ☐ Android平台 ☐ iOS平台 ☐ Web端(JS API)

☒ Web服务 ☐ 智能硬件 ☐ 微信小程序

☐ HarmonyOS NEXT 平台

可使用服务: 静态地图API 地理编码API 逆地理编码API 关键字搜索API

周边搜索API 多边形搜索API ID查询API 输入提示API

路径规划API 坐标转换API 行政区划查询API IP定位API

天气查询API 矩形区域交通态势API 圆形区域交通态势API 指定线路交通态势API

地理围栏API 猎鹰服务API GeoHUB服务API

Figure3.3:Figure For Gaode API Key types service

```
import requests

def get_location_from_ip(amap_api_key):
    """
    调用高德地图 IP 定位 API 获取大致经纬度和区域信息
    """
    url = "https://restapi.amap.com/v3/ip"
    params = {
        'key': amap_api_key # 高德地图 API 密钥
    }
    response = requests.get(url, params=params)
    if response.status_code == 200:
        data = response.json()
        if data['status'] == '1': # 确认请求成功
            # 提取省、市、矩形范围
            province = data.get('province', '未知省份')
            city = data.get('city', '未知城市')
            rectangle = data.get('rectangle', None) # 返回矩形区域范围
            return {
                'province': province,
                'city': city,
                'rectangle': rectangle
            }
    return None
```


First, we successfully get the location in WhenZhou, but we want to get more detailed location.

This HTML page is generated by ChatGPT

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>实时获取 GPS 坐标</title>
5 </head>
6 <body>
7   <h1>实时 GPS 数据</h1>
8   <p id="location">正在获取位置...</p>
9   <script>
10     const ws = new WebSocket("ws://localhost:8080");
11
12     if ("geolocation" in navigator) {
13       navigator.geolocation.watchPosition(
14         (position) => {
15           const gpsData = {
16             latitude: position.coords.latitude,
17             longitude: position.coords.longitude,
18             accuracy: position.coords.accuracy
19           };
20
21           // 更新页面显示
22           document.getElementById("location").innerText =
23             `纬度: ${gpsData.latitude}, 经度: ${gpsData.longitude}, 精度: ${gpsData.accuracy} 米`;
24
25           // 将 GPS 数据通过 WebSocket 发送到服务器
26           ws.send(JSON.stringify(gpsData));
27         },
28         (error) => {
29           console.error("无法获取位置:", error.message);
30         },
31         {
32           enableHighAccuracy: true
33         }
34       );
35     } else {
36       document.getElementById("location").innerText = "设备不支持 GPS 定位。";
37     }
38
39     ws.onopen = () => console.log("WebSocket 已连接");
40     ws.onclose = () => console.log("WebSocket 已断开");
41   </script>
42 </body>
43 </html>
```

Figure3.4:Html to open webpage to get GPS location



Figure3.5:GPS Result

```

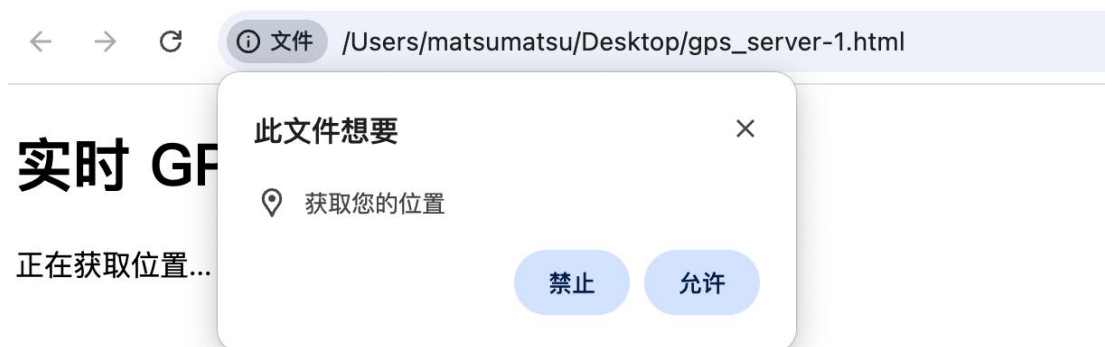
import requests
import time

def get_gps_coordinates():
    url = "file:///Users/matsumatsu/Desktop/gps_server.html" # HTML 提供的 API 地址
    while True:
        try:
            response = requests.get(url)
            if response.status_code == 200:
                gps_data = response.json()
                print(f"实时 GPS 数据: 纬度={gps_data['latitude']}, 经度={gps_data['longitude']}, 精度={gps_data['accuracy']} 米")
            else:
                print("无法获取 GPS 数据")
        except Exception as e:
            print(f"请求失败: {e}")
            time.sleep(5)

get_gps_coordinates()

```

We can get the current gps coordinates manually because of location permissions on the web page, but the program can't get them directly from the web page.



```

9.0s
请求失败: No connection adapters were found for 'file:///Users/matsumatsu/Desktop/gps_server.html'
请求失败: No connection adapters were found for 'file:///Users/matsumatsu/Desktop/gps_server.html'

```

```

def get_detailed_address_from_coordinates(amap_api_key, lng, lat):
    """
    调用高德地图逆地理编码 API 解析经纬度为详细地址
    """
    url = "https://restapi.amap.com/v3/geocode/regeo"
    params = {
        'key': amap_api_key,
        'location': f"{lng},{lat}", # 经纬度
        'extensions': 'all' # 返回详细的街道、门牌号、兴趣点等信息
    }
    response = requests.get(url, params=params)
    if response.status_code == 200:
        data = response.json()
        if data['status'] == '1': # 请求成功
            regeocode = data.get('regeocode', {})
            formatted_address = regeocode.get('formatted_address', '未知地址')
            address_component = regeocode.get('addressComponent', {})

            # 提取详细的区、街道、门牌号、兴趣点
            district = address_component.get('district', '未知区')
            township = address_component.get('township', '未知街道')
            street = address_component.get('streetNumber', {}).get('street', '未知街道')
            number = address_component.get('streetNumber', {}).get('number', '未知门牌号')
            poi = regeocode.get('pois', [{}])[0].get('name', '无兴趣点')

            return {
                'formatted_address': formatted_address,
                'province': address_component.get('province', '未知省份'),
                'city': address_component.get('city', '未知城市'),
                'district': district,
                'township': township,
                'street': street,
                'number': number,
                'poi': poi
            }
    return None

```

```

def get_full_detailed_address(amap_api_key):
    """
    综合 IP 定位和逆地理编码，返回最详细的地址
    """
    # Step 1: 通过 IP 定位获取经纬度范围
    location_info = get_location_from_ip(amap_api_key)
    if not location_info or not location_info.get('rectangle'):
        return "无法获取位置信息"

    # Step 2: 计算矩形范围的中心点作为经纬度
    rectangle = location_info['rectangle']
    lng1, lat1, lng2, lat2 = map(float, rectangle.replace('; ', ',').split(','))
    lng = (lng1 + lng2) / 2 # 计算经度中点
    lat = (lat1 + lat2) / 2 # 计算纬度中点

    # Step 3: 调用逆地理编码 API 获取详细地址
    detailed_address = get_detailed_address_from_coordinates(amap_api_key, lng, lat)
    return detailed_address

# 示例
amap_api_key = "1eac4b127c4bedfe50b5ae79fdc9c12e" # 替换为你的高德地图 API 密钥
address_info = get_full_detailed_address(amap_api_key)

if address_info:
    print(f"详细地址: {address_info['formatted_address']}")
    print(f"省份: {address_info['province']}")
    print(f"城市: {address_info['city']}")
    print(f"区: {address_info['district']}")
    print(f"街道: {address_info['township']}")
    print(f"门牌号: {address_info['street']} {address_info['number']}")
    print(f"兴趣点: {address_info['poi']}")
else:
    print("无法获取详细地址信息")

```

✓ 0.2s

Python

```
详细地址：浙江省温州市鹿城区南汇街道温州市人民政府(北门)
省份：浙江省
城市：温州市
区：鹿城区
街道：南汇街道
门牌号：绣山路 321号
兴趣点：温州市人民政府
```

Figure3.6: Code Result For Detailed Location

3.2 Emergency call function code

```
# 紧急呼叫模块
def send_emergency_message():
    """
    向紧急联系人发送求助信息。
    """
    try:
        # 使用 Twilio 或短信 API 发送信息
        print(f"[发送求助信息] 联系人: {EMERGENCY_CONTACT}, 位置: {location_history[-1]}")
        text_to_speech("求助信息已发送, 请稍候。")
    except Exception as e:
        print(f"求助信息发送失败: {e}")
```

Python

3.3 Voice interaction features Code

For the voice assistant module, my initial plan was to utilize ChatGPT's API to enable voice interaction. However, due to restrictions on accessing the GPT API in mainland China, I decided to use the Doubao API as an alternative to implement the voice interaction functionality. This alternative aims to ensure that the voice assistant module can function effectively and provide a user-friendly interface for interaction.

Unfortunately, the Doubao developer platform does not provide the relevant API functionality needed for this purpose. This limitation poses a challenge to implementing the voice interaction module and necessitates exploring alternative solutions or platforms to achieve the desired functionality.

```

import sounddevice as sd
import numpy as np
import wave
import openai
from gtts import gTTS
import os

# 设置 OpenAI API 密钥
openai.api_key = "your_openai_api_key" # 替换为您的 OpenAI API 密钥

def record_audio(filename="input.wav", duration=5, samplerate=44100):
    """录音并保存为 WAV 文件"""
    print("开始录音...")
    audio = sd.rec(int(duration * samplerate), samplerate=samplerate, channels=2, dtype=np.int16)
    sd.wait() # 等待录音完成
    print("录音完成, 正在保存音频文件...")

    # 保存为 WAV 文件
    with wave.open(filename, 'wb') as wf:
        wf.setnchannels(2) # 双声道
        wf.setsampwidth(2) # 每样本 2 字节
        wf.setframerate(samplerate)
        wf.writeframes(audio.tobytes())
    print(f"音频已保存为 {filename}")

```

Figure3.7: Code Part For Chat GPT API

```

# 豆包 API 的基础 URL 和 API Key
BASE_URL = "https://api.doubao.com/chat" # 示例 URL, 请替换为实际的
API_KEY = "your_doubao_api_key" # 替换为您的 API Key

def chat_with_doubao(user_input):
    """调用豆包 API 进行聊天"""
    headers = {
        "Authorization": f"Bearer {API_KEY}", # 使用 API Key 进行身份验证
        "Content-Type": "application/json"
    }
    payload = {
        "message": user_input # 用户输入
    }
    try:
        # 发起 POST 请求
        response = requests.post(BASE_URL, json=payload, headers=headers)
        response.raise_for_status() # 如果 HTTP 状态码不是 200, 会抛出异常

        # 解析返回的数据
        data = response.json()
        if "reply" in data:
            return data["reply"] # 返回聊天机器人回复
        else:
            return "抱歉, 我不太明白您的意思。"
    except requests.exceptions.RequestException as e:
        print(f"调用豆包 API 出错: {e}")
        return "抱歉, 无法连接到豆包服务。"

# 示例交互
if __name__ == "__main__":
    print("欢迎使用豆包聊天助手! (输入 '退出' 以结束)")
    while True:
        user_input = input("我: ")
        if user_input.lower() in ["退出", "再见"]:
            print("豆包: 再见!")
            break
        reply = chat_with_doubao(user_input)
        print(f"豆包: {reply}")

```

Python

欢迎使用豆包聊天助手! (输入 '退出' 以结束)

Figure3.8: Code Part For DouBao API

```

def transcribe_audio_to_text(filename="input.wav"):
    """将音频文件转换为文本"""
    import speech_recognition as sr
    recognizer = sr.Recognizer()
    with sr.AudioFile(filename) as source:
        print("正在转换音频为文本...")
        audio = recognizer.record(source)
        try:
            text = recognizer.recognize_google(audio, language="zh-CN") # 中文语音识别
            print(f"您说的是: {text}")
            return text
        except sr.UnknownValueError:
            print("无法识别语音, 请再试一次。")
            return None
        except sr.RequestError as e:
            print(f"语音识别服务出错: {e}")
            return None

def get_chatgpt_response(prompt):
    """调用 ChatGPT API 获取回复"""
    try:
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo", # 使用 GPT-3.5 或更高版本模型
            messages=[{"role": "user", "content": prompt}],
            temperature=0.7
        )
        message = response["choices"][0]["message"]["content"]
        return message.strip()
    except Exception as e:
        print(f"调用 ChatGPT API 出错: {e}")
        return "抱歉, 我无法连接到 ChatGPT 服务。"

```

```

def speak_text(text):
    """将文本转换为语音并播放"""
    try:
        tts = gTTS(text=text, lang="zh") # 中文语音输出
        tts.save("response.mp3")
        os.system("start response.mp3" if os.name == "nt" else "afplay response.mp3") # Windows 用 start, Mac 用 afplay
    except Exception as e:
        print(f"语音播放出错: {e}")

def chat_with_gpt():
    """主循环: 语音助手和 ChatGPT 对话"""
    while True:
        print("\n等待您的语音输入...")
        record_audio() # 录音并保存为 input.wav
        user_input = transcribe_audio_to_text() # 转录为文本
        if user_input is None:
            continue # 如果语音无法识别, 重新开始
        if "退出" in user_input or "再见" in user_input: # 用户说"退出"或"再见"时结束程序
            print("再见!")
            break

        # 获取 ChatGPT 回复
        print("正在调用 ChatGPT API...")
        response = get_chatgpt_response(user_input)
        print(f"ChatGPT 回复: {response}")

        # 语音输出
        speak_text(response)

if __name__ == "__main__":
    chat_with_gpt()

```

